ISAST Transactions on No. 1, Vol. 1, 2008 (ISSN 1797-1802)

Intelligent Systems

Regular Papers

Greetings from ISAST

Dear Reader,

You have the first ISAST Transactions on Intelligent Systems on your hands. It consists of eleven original contributed scientific articles at the various fields of intelligent systems. Every article has gone through peer-review process.

ISAST - International Society for Advanced Science and Technology – was founded in 2006 for the purpose of promote science and technology, mainly electronics, signal processing, communications, networking, intelligent systems, computer science, scientific computing, and software engineering, as well as the areas near to those, not forgetting emerging technologies and applications.

To show, how large the diversity of computers and software engineering field is today, we shortly summarize the contents of this Transactions Journal:

In the paper of D. Bhanu and P. Balasubramanie introduce a rule mining approach in predicting market behaviour. Grant Cochenour, Sanjoy Das, Anil Pahwa and Jerad Simon present a radial basis function neural network approach to predict the failures in overhead transmission lines of power delivery systems. M. Bastanfard and S. D. Katebi have three research papers from the area of reinforcement learning. In the first paper they discussed using reinforcement learning algorithm to solve the combinational optimization problem, according to mutual information restriction. In their second paper they presented a novel idea for enhancing planning and scheduling efficiency by reinforcement learning. A method for feature selection by reinforcement learning is considered in their third paper. Hassan Ezzaidi studies the use of neural networks in recognising instruments and families from their spectral information. On the other hand Jonathan M. Blackledge and Dmitry A. Dubovitskiy have presented a novel idea for detecting skin cancer from digital images using object detection and classification. Nhien-An Le-Khac, Lamine M. Aouad and M-Tahar Kechadi introduce a knowledge map layer that assists creating efficient knowledge management systems in distributed data mining. Vasudha Bhatnagar, Sharanjit Kaur and Ajit Chaturvedi propose a framework for grading data regions as anomalies or patterns in streams. Peter Kazik, Igor Sivy and Frantisek Jakab have a research paper about intelligent e-learning system that has intelligent features to satisfy the needs of different level learners. Finally, Karl Altenburg, Michael Hennebry, Jonathan Pikalek and Kendall E. Nygard introduce a multi-agent simulation environment for unmanned aerial vehicle task allocation.

We are happy to see how much we have obtained manuscripts with ambitious and impressive ideas. We hope that you will inform of the existence of our Society to your colleagues to all over the academic, engineering, and industrial world.

Best Regards,

Professor Timo Hämäläinen, University of Jyväskylä, FINLAND, Editor-in-Chief
Professor Jyrki Joutsensalo, University of Jyväskylä, FINLAND, Vice Editor-in-Chief

A Predictive and Forecasting model for increased sales – A Rule Mining approach

D.Bhanu, P.Balasubramanie

Abstract— **It is often difficult to find situations where sophisticated methods yield more accurate forecasts of market trends than that can be obtained from naïve forecasts. This paper investigates market basket analysis as an important component of attracting customers in retail stores. Association Rule Mining, which is an integral part of Data Mining, is often used to generate the set of rules. The discovery of rules from large databases has proven beneficial for companies since such rules can be very effective in revealing actionable knowledge that leads to strategic decisions. Typically, rules generated are derived from patterns in a particular dataset. A major issue that needs more attention is the soundness of the rules outside of the dataset from which they are generated. When a new phenomenon happens, the change in the set of rules generated from the new dataset becomes more significant. In this paper, we provide a model for understanding how the differences between different situations affect the set of rules generated based on the concept of groups what we call factions. Also we provide a technique called Coalescent Dataset, to generate a set of rules for a new situation. Various experimental results are reported by comparing with real life and synthetic datasets.**

Index Terms— **Association Rules, Data Mining, Coalescent Dataset, Factions**

I. Introduction

One of the challenges for companies that have invested heavily in customer data collection is how to extract important information from their vast databases and product feature databases, in order to gain competitive advantage. Market Basket analysis (popularly known as Association Rule Mining) is one of the Data Mining methods [5] focusing on discovering purchasing patterns by extracting associations or co-occurrences from stores transactional data. Several aspects of Market Basket analysis have been studied in academic literature, such as using customer interest profile and interests on particular features of the product for the product development and one-to-one marketing [16], purchasing patterns in a multi-store environment [Chen et. al., 2004] or point at certain weaknesses of Market Basket analysis techniques [15]. But there is a little attention to address the soundness of the rules outside of the dataset from which these rules are derived. In general rules are derived from patterns of a particular dataset. For a new situation, knowing which products sell together can be very useful to any business. Once it is known that customers who buy one product are likely to buy another, it is possible for the company to market the products together or to make the purchases of one product the target prospects for another. So, to make strategic decisions for a new situation, the only available rule from the dataset collected from an earlier situation becomes the major source. The problem that has to be addressed in detail is how far the rules derived from the existing stores in different locations are effective when they are applied to a store at an entirely new location. Hence, we need a way to obtain an applicable set of rules for the new store, without the availability of sales transaction at the new store. We have utilized the formulation of association rule problem for interval data in a way that respects the quantitative properties and the semantics of the data.

In this paper, we address the issue of extending the applicability of association rules. A model is proposed for understanding the differences between the two sets of rules from datasets at different situations. It helps us to understand the differences in the set of rules generated, which are derived from the given dataset collected over earlier domain. The model is based on the notion of using a common set of partitioned groups for every situation, which we call factions. We provide a simple technique called the Coalescent Dataset, which yields the new set of rules. Whenever, factors affecting the rules can be modeled as distinct set of groups, the technique can be applied to generate the rules for a new situation based on the availability of the data from an available situation.

The main objective of the model is to formulate and sample the Coalescent Dataset as how to discover association rules; we can use existing algorithms [3],[4],[13], [10], [9], [12], proposed. The rest of the paper is organized as follows. Section 2, discusses the Association Rule Mining in detail. Section 3 describes the problem and methodology. Section 4 explains the experimental set-up and the results and will conclude with section 5.

Manuscript received August 16, 2007.

D Bhanu is with Department of Computer Applications, Sri Krishna College of Engineering and Technology, Coimbatore, Tamilnadu, - 641008, INDIA (Phone: 91-0422-2529952;e-mail:bhanusaran@yahoo.com.

P.Balasubramanie is with the Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, Tamilnadu – 638032, INDIA.

II. Association Rule Mining

Much work in data mining revolves around the discovery of rules within large quantities of data. Rules over relations are of the form $C_1 \Rightarrow C_2$ where C_1 and C_2 are conditions on tuples of the relation [11]. Such a rule may be exact, meaning that all tuples that satisfy C_1 also satisfy C_2, it may be strong, meaning that tuples satisfying A almost always satisfy B or it may be approximate meaning the some of the tuples satisfying A also satisfy B [11]. Most often, the conditions are restricted to be simple equality predicates or conjunctions of such predicates on different attributes.

The term association rule has been used to describe a specific form of such rules [1, 2, 8, 9, 10, 14]. The rule frequency is a measure of how often a rule occurs in a dataset. When defined as the fraction of all tuples in a relation that satisfies a rule (specifically, $| C1 \wedge C2 | / |r|$ where r is the set of all tuples considered), frequency is referred to as support [1]. The strength of the rule implication is a measure of how often a rule is likely to be true within the dataset. When defined as the fraction of all tuples satisfying C1 that also satisfies C2 (that is $| C1 \wedge C2 | / |C1|$), rule strength is referred to as Confidence. The problem of discovering association rules is then traditionally defined as follows: given fixed thresholds for the permissible minimum support and confidence, find all association rules that hold good with more than the given support and confidence.

Algorithms for discovering classical association rules make different assumptions about the type of data sets to be mined. In general, the data set is a relational table. The domains of the attributes may be restricted to Boolean domains [1, 2]. Under this formulation, an association rule $(X_1 = 1) \wedge (X_2 = 1) \wedge \ldots (X_n = 1) \Rightarrow (Y_1 = 1) \wedge \ldots (Y_m = 1)$ is often abbreviated as $X \Rightarrow Y$ where $X = \{X_1 \ldots X_n\}$ and $Y = \{Y_1 \ldots Y_m\}$ are sets of attributes. This definition has been generalized to include relational tables over arbitrary domains including qualitative and quantitative domains [13].

III. Problem Description

A. Importance of Locales

The importance of rules identified by the association rule mining is based on the validity and the credibility of the discovered rules. Consider a supermarket (company) managing a chain of stores that discovers association rules in the sale transactions at one of its stores. For opening a new store, the company would make use of the knowledge obtained from the discovered association rules. However, if the buying behaviour and people culture is different, the rules that are derived are not pertinent to the new store. From the strategic point of view, it is important from the point of decision maker to know which rules discovered from the transactions at the first store are applicable to the new store at a different location. Therefore, when applying association rules derived from one dataset to a new situation, it is necessary to ensure that the factors deriving the rules are consistent between the source of the rules and where the rules are applied. Hence, a locale, which determines the changes in the set of rules, helps us understand the discrepancy in the rules observed at the two stores.

B. Problem Formulation

The problem focuses at estimating the association rules generated for a situation based on the data collected for another situation with a varied distribution of locales. The set of fields/ attributes from which association rules are determined are called items. An attribute whose function is not primary, but whose value affects the relation between items is called a factor. Factors that possess demographic characteristics are used to identify a particular group of people having similar characteristics. Gender, age etc., can be considered as factors which affect / changes the buying behaviour in different situations. A set of factors with associated values that generates certain set of rules is called factions. In section C.2, we utilize a method to generate factions. The model for generating association rules is as follows: Every faction has a tight binding in generating set of association rules. The concept behind being that people from the same faction will behave in the same way. Each situation has a mix of different factions in some proportions. The difference in the mix of proportions of the factions for different situations gives rise to the difference between the set of association rules for each situation.

The major task is to determine the association rules for a varying situation. The problem of estimating association rules can be done if we have the dataset for the first situation (the source dataset), the set of factions that make up the two situations, and the proportions of the factions for the new situation. Also, we define a method to construct Coalescent Dataset, which is a dataset sampled from the source dataset by the proportions of the factions for the new situation, to get a good estimate of the large item sets for the new situation. Obtaining the dataset is not much difficult from which we can get source dataset with its corresponding background values. For the Construction of factions, a clustering method is utilized. The proportions of factions vary and the survey approach is adopted to get the proportions of the new situation.

C. Coalescent Data Set

This is mainly based on the availability of sample representatives from each faction. The mining process uses the same notion of support as the measure of significance. Other measures of significance can replace an appropriate support in the data combination process in order to estimate rules for a new situation.

1. Sampling Technique

The behaviour of a faction is considered to be consistent in any situation. This behaviour is inherent from the transactions of the factions and is represented in the form of mined patterns / rules. Since the transactions of a particular faction contains all its patterns for constructing a Coalescent Data Set for a new situation, the faction with larger proportion in the overall population must have more transactions in the data set to reflect its importance.

In this sampling method, a Coalescent Data set for the new situation is constructed from a source data set. The relative proportion of faction in the new situation determines the number of transactions for that faction in the Coalescent Data Set. The total number of transactions for a particular faction in the Coalescent data Set is equal to the product of the size of the Coalescent Data Set and all the relative proportions of the particular faction. Once the Coalescent data Set has been constructed, the association rule mining can be performed on to it to obtain the rules for the new situation.

2. Construction of Factions

The factions play a major role in constructing the Coalescent Data Set. In this section, we will discuss the methods used in constructing the factions. The idea is to use a standard clustering algorithm to identify intervals of interest in case of quantitative attributes. The method use a single partitioning of attribute into disjoint sets X_i over which there is a meaningful distance metric. Most often each X_i is an individual attribute / a small set of closely related attributes over similar domains. The basic idea behind is that clusters can be incrementally identified and refined in a single pass over data. A Clustering Feature, that is a succinct summary of the properties of the cluster, represents each cluster. From the CFS of two clusters, the CF of their union and the number of distance metrics can be derived [7]. Hence, clusters can be combined and new points are added to clusters using only the CFs. All clusters found will satisfy the density threshold. Moreover, all clusters also satisfy the frequency threshold i.e., the clusters with sufficient support are used. These clusters form the set of factions that are used as input to the generation of Coalescent Dataset.

D. Analysis of Sampling

As the next case, we analyze the relation of sample size to the accuracy of results. We first consider how accurate the frequencies computed from a random sample are. Samples of reasonable size provide good approximations for frequent sets.

1. Accuracy and Sample Size

We consider the absolute error of the estimated frequency. Given an attribute set X which is a subset of R and a random sample s from a relation over binary attributes R, the *error* e (X,s) is the difference of the frequencies:

$$e(X,s) = | fr(X) - fr(X,s) |, \quad (1)$$

where fr (x) is the frequency of X in the relation from which s was drawn. To analyze the error, we consider sampling with replacement. The reason is that we want to avoid making other assumptions of the database size except that it is large. For sampling with replacement the size of the database has no effect on the analysis, so the results apply in principle, on infinitely large databases. For very large databases there is practically no difference between sampling with and without replacement.

We analyze the number of rows in the sample s that contain X, denoted m (X, s). The random variable m (X, s) has binomial distribution, i.e., the probability of m (X, s) = c, denoted

$$\Pr [m (X, s) = c], \text{ is}$$
$$P (|s|, c) fr (X)^{c} (1\text{-}fr (X))^{|s| - c} \quad (2)$$

First, we consider the necessary size of the sample, given requirements on the size of the error. The following theorem gives a lower bound for the size of the sample, given an error bound ξ and a maximum probability δ for an error that exceeds the bound.

Theorem 1: Given an attribute set X and a random sample s of size

$$|s| \geq 1/ 2 \xi^{2} \ln 2/ \delta \quad (3)$$

The probability that x (X, s) > ξ is at most δ.

Proof: We have

$$\Pr [e (X,s) > \xi] = \Pr\{\{| fr(X,s) \}- fr(X) ||s| > \xi |s|] \quad (4)$$

The Chernoff bounds give an upper bound

$$2e^{-2(\varepsilon |s|) 2 / |s|} \approx \delta \quad (5)$$

for the probability. Table 1 gives values for the sufficient sample size |s|, for ε = 0.01, 0.001 and δ = 0.01, 0.001, 0.001. With tolerable error ε around 0.01, samples of a reasonable size suffice. For many applications, these parameter values are perfectly reasonable – errors in the input data may be more likely than 0.0001. In such cases, approximate rules can be produced based on a sample. With tighter error requirements the sample sizes can be quite large.

Table 1: Sufficient sample sizes, given ε and δ

ξ	δ	|s|
0.01	0.01	27,000
0.01	0.001	38,000
0.01	0.0001	50,000
0.001	0.01	2,700,000

0.001	0.001	3,800,000
0.001	0.0001	5,000,000

IV. EXPERIMENTAL RESULTS

We now explain the experiments we conducted in order to assess the practical feasibility of using samples of Coalescent Dataset for finding frequent sets and the set of factions generated for the datasets. We use synthetic datasets and real life datasets in our tests. These databases model supermarket basket data. The central properties of the datasets are the following. There are $|R| = 1000$ attributes, and the average number T of attributes per row is 5, 10 and 20. The number $|r|$ of rows is approximately 100,000. The average size I of maximal frequent sets is 2, 4 and 6. Table 2 summarizes the parameters for the datasets [2]. We assume that the real datasets from which association rules are discovered can be much larger than the test datasets. To make the experiments fair, we use sampling with replacement. We considered sample sizes from 20,000 to 80,000 since samples of these sizes are large enough to give good approximation.

Table 2: Dataset characteristics (T = row size on average, I = size of maximal frequent sets on average).

min_fr (%)	**Sample Size**			
	20,000	40,000	60,000	80,000
0.25	0.13	0.17	0.18	0.19
0.50	0.34	0.38	0.40	0.41
0.75	0.55	0.61	0.63	0.65
1.00	0.77	0.83	0.86	0.88
1.50	1.22	1.30	1.33	1.35
2.00	1.67	1.77	1.81	1.84

Table 3: Lowered Frequency Thresholds (%) for $\delta = 0.001$

min_fr (%)	**Sample Size**			
	20,000	40,000	60,000	80,000
0.25	0.13	0.17	0.18	0.19
0.50	0.34	0.38	0.40	0.41
0.75	0.55	0.61	0.63	0.65
1.00	0.77	0.83	0.86	0.88
1.50	1.22	1.30	1.33	1.35
2.00	1.67	1.77	1.81	1.84

The lowered threshold depends on the frequency threshold and the sample size. The lowered threshold values are given in Table 3. We used in the computations the exact probabilities from binomial distribution. Since the accuracy of the rules is directly dependent on the accuracy of the large item sets, we demonstrate the efficiency of the technique from the quality of large item sets generated. We tested our method with two different types of data: Synthetic data and real life data. In both cases, we use the data generation procedure described below to generate the datasets, one for the source and the another for the target, respectively. We then generate the Coalescent dataset using the technique stated, and mine the Coalescent Dataset to obtain its large item sets. The extent of matching between the source and the target dataset determines the accuracy of using only the source dataset, and the extent of matching between the coalescent dataset and the target determines the accuracy of the approach.

A. Data Generation

Synthetic Dataset: The popular method of generating transactions containing associations, presented by Agarwal and Srikant [2] is used. The same probability distribution is used and the same notations are taken [2]**.** We generate datasets for two situations: one is used as the source and the other as target. Each of the two datasets is partitioned into a common set of G factions. For each dataset, the generation program assigns random proportions to every faction. The relative proportion of a faction for the source has no relation to the relative proportion of any faction for the target. Given the total number of transactions in a dataset, the relative proportion of the faction determines the number transactions in that faction. The set of items involved in different factions is a major source that causes the difference of the large item sets / rules among the factions. Therefore the items assigned to factions are different from faction to faction. These two together give rise to the overall difference between the large item sets of the two data sets.

B. Generating Item sets

An item set for a faction is generated by selecting initially the size of the item set from a Poisson distribution with mean which is equal to |I|, the average size of an item set. A fraction of items in an item set is chosen from the previous item set. An exponentially distributed random variable with mean that is equal to the correlation level decides this fraction for each item set. This generates the fact that large item sets usually have common items. The remaining items in the item sets are picked according to the weights assigned with the items. The total number of item sets for a faction is an input, to the program. There is a weight distribution for the item sets for each faction, which is derived from an exponential with unit mean. The weights correspond to the probability of choosing the corresponding item sets when creating transactions.

C. Generating Transactions

Transactions are created by the addition of item sets. All transactions for a particular faction is generated using the item sets of that particular item set. First, the size of the transaction is chosen from a Poisson distribution with mean given by an input parameter |T|. Item sets are added to the transaction until the total number of item is equal to or exceeds the number of items for that transaction. When the length of the transaction is exceeded, the item set is added to the transaction with a probability of 0.5. The Table 4 gives the various input parameters and the values used for them.

Table 4: Parameters for Data Generation

G	Number of Factions	25
..	Number of items in a faction	100
..	Number of common items between factions	0,50,100
|I|	Average size of item set	4
|L|	Number of item sets for a faction	500
|T|	Average size of transaction	10
N	Number of transactions in a data set	100000
C	Corruption level	Mean = 0.5 Variance = 0.1
--	Average value for Correlation level	0.5

D. Real Life Data Sets

The technique has been tested on the Census data set, which is downloaded from the UCI Repository of Machine Learning databases. For our purpose, we modify the dataset to form our database. We have used 14 attributes in our dataset. Even if the number of items is very large, each transaction has exactly 14 items. To create datasets the following method is adopted. Select some attributes as background attributes and set the remaining as foreground attributes. Generate the factions. Assign the factions for two data sets D1 and D2. Proportions of the factions for D1 are the same as that of the original database, but the size of the database is smaller than that of the original. Proportions of factions for D2 are different from the original database, and also with its size smaller than the original database. Generate datasets D1 and D2, according to their size and the corresponding proportions of factions

E. Performance on Synthetic Data

For synthetic data, to detect any variation in the efficiency of the technique, the important aspect has been studied. All the factions involve the same set of items, half of the items for a faction is common to all and half is unique to that faction and all the items for a faction is distinct from the items in any other faction. For all the cases, the total number of items for each faction is set to 100. A support value of 5% is used as the threshold. Figure 1 shows the distribution of transactions with the specified threshold for the algorithm applied along time. Figure 2 (a), 2(b), 2(c) shows the number of large item sets predicted and Figure 2(d), 2(e) shows the distribution of error in support values for predicted item sets.

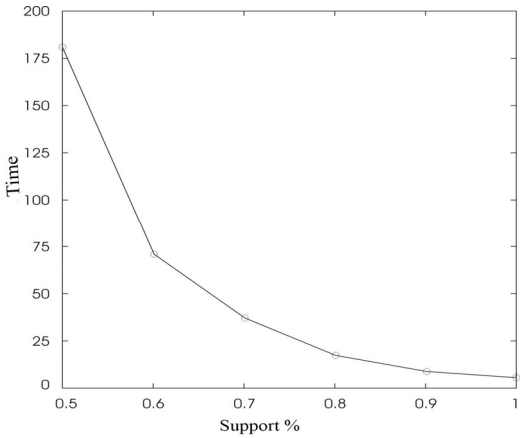


Figure: 1 Effect of increasing the size of transactions for support – 0.5%

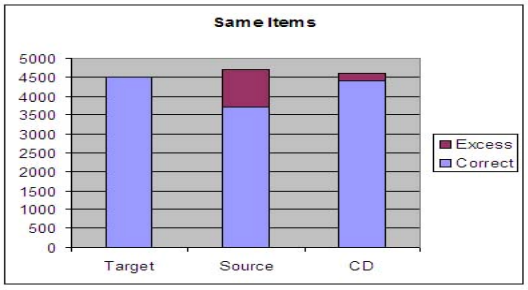


Figure: 2(a) Distribution of predicated large item sets for same items

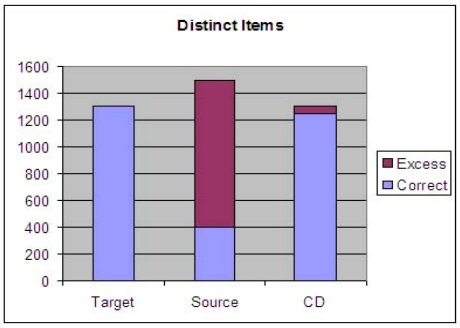


Figure: 2(b) Distribution of predicated large item sets for half same and half different items

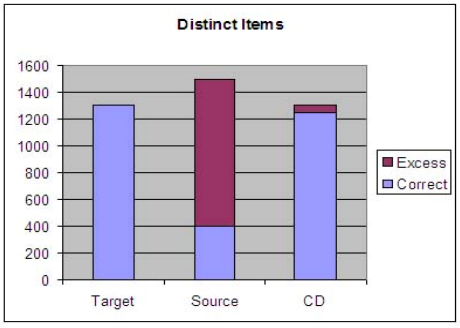


Figure: 2(c) Distribution of predicated large for distributed items

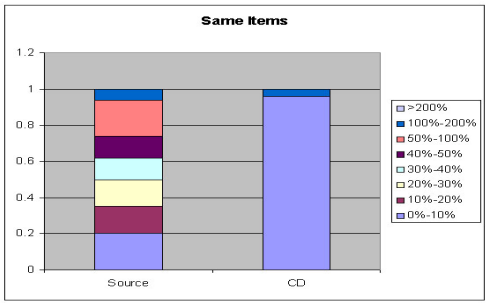


Figure: 2(d) Distribution of error in support value for same items

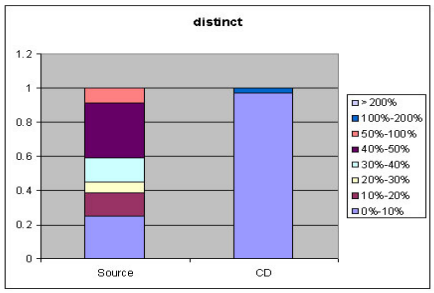


Figure: 2 (e) Distribution of error in support values for half same and half distinct items

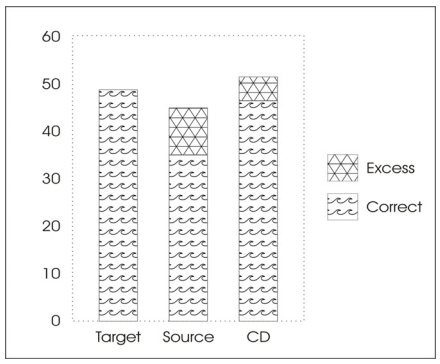


Figure 3: Distribution of error in the source and CD approach

In Figure 3, the bars marked Source and CD is associated with the direct approach and the Coalescent dataset approach. The target represents the distribution of predicated large item sets. From the experiments conducted, we know that our technique is quite accurate in capturing the correct set of item sets and also the values for their actual support.

V. CONCLUSION

The mining of association rules derived from data has been investigated in data mining. Most of the research is focused on optimizing the process, developing methods to handle different kinds of data, developing appropriate significance metrics, and improving the users control over the process. In this paper, we address the issue of extending the applicability of rules out of the existing datasets from where they were generated. We provide a model that distinguishes the difference between the situations using the concept of factions and Coalescent Dataset. Different situations have their own factions in different proportions. Using this model, we derive rules for a new situation, when given the datasets for the old situation.

REFERENCES

[1] - R. Agarwal, T. Imielinski, and A. Swami. Mining Association Rules between sets of items in large databases. In Proc. 1993 ACM – SIGMOD Int. Conf. management of Data, Pages 207 – 216, Washington, D.C., May 1993.

[2] - R.Agarwal and R.Srikant. Fast algorithms for mining association rules. In Proc. 1994 Int. Conf. Very large Data Bases, pages 487 – 489, Santiago, Chile, September 1994.

[3] – R.Agarwal and J.Shaffer, "Parallel Mining of association rules: Design, Implementation and Experience," IEEE Trans. Knowledge and Data Engg, 1996.

[4] – S.Brin, R.Motwani, J.D.Ullman, and S.Tsur, " Dynamic Item set Counting and Implication rules for Market Basket Data," Proc. 1997 ACM SIGMOD Int'l Conf. Management of Data, pp. 255-264, 1997.

[5] – Berry, M.J.A., Linoff, G.S.: Data Mining Techniques: for Marketing, Sales and Customer Relationship Management (second edition), Hungry Minds Inc., 2004.

[6] - Chen, Y. -L., Tang, K., Shen, R. -J., Hu, Y.-H. : "Market basket analysis in a multiple store Environment", Decision Support Systems, 2004, accessed through www.ScienceDirect.com.

[7] – J.Han, Y.Cai, and N.Cercone. Data-driven Discovery of Quantitative Rules in Relational Databases. IEEE Transactions on Knowledge and Data Engineering, 5(1):29-40, 1993.

[8] – J.Han and Y.Fu, " Discovery of Multiple-Level Association Rules from Large Databases," Proc. 21st Int'l Conf. Very Large Databases, pp. 429-431, Sept. 1995.

[9] - H.Mannila, H.Toivonen, and A.Verakamo, "Efficient Algorithms for discovering association rules," Proc. AAAI '94 Workshop Knowledge Discovery in Databases, pp. 181 – 192, July 1994.

[10] - J.S.Park, M.S.Chen, and P.S.Yu. An effective hash based algorithm for mining association rules. In Proc. 1995 ACM – SIGMOD Int. Conf. Management of Data, pages 175 – 186, San Jose, CA, May 1995.

[11] – G.Piatetsky – Shapiro. Discovery, Analysis, and Presentation of Strong rules. In [PSF91], pages 229 – 248, 1991.

[12] – K.Rajamani, B.Iyer, and A.Chadha, "Using DB2's Object Relational Extensions for Mining Association Rules" , Technical Report TR 03, 690, Santa Teresa Laboratory, IBM Corp., Sept.1997.

[13] – R.Srikant and R.Agarwal, "Mining Generalized Association Rules", Proc. 21st Int'l Conf. Very Large Databases, pp.407 – 419, Sept. 1995.

[14] – A.Savasere, E.Omiecinski, and S.Navathe. An Efficient Algorithm for Mining Association Rules in Large Databases (VLDB), Zurich, Switzerland, September 1995.

[15] - Vindovogel, B., VandenPoel, D., Wets.G.: " Why promotion strategies based on market based analysis do not work" (article in press), Expert Systems with Applications, 2005,accessed through www.ScienceDirect.com.

[16] – Weng, S. -S., Liu, J.-L.: "Feature –based recommendation for one-to-one marketing", Expert Systems with Applications, Vol. 26, 2004, pp. 493-508.

ISAST Transactions on Intelligent Systems, No. 1, Vol. 1, 2008
G. Cochenour et al.: A Multi-Objective Evolutionary Strategy Based Radial Basis Function Network Approach for Predicting Failure Rates in Distribution Systems

A Multi-Objective Evolutionary Strategy Based Radial Basis Function Network Approach for Predicting Failure Rates in Distribution Systems

Grant Cochenour, Sanjoy Das, Anil Pahwa, Jerad Simon

***Abstract*--This paper outlines a radial basis function neural network approach to predict the failures in overhead transmission lines of power delivery systems. The networks are trained using historical data from various sources. The network sizes and errors are simultaneously minimized using the Pareto Archive Evolutionary Strategy algorithm. Mutation of the network is carried out by invoking an orthogonal least square procedure. The performance of the proposed method is compared to a fuzzy inference approach and with multilayered perceptrons. The results suggest that this approach outperforms the other techniques for the prediction of failure rates.**

***Index Terms*--Distribution system, evolutionary strategy, multi-objective optimization, neural network, radial basis function network, reliability.**

I. Introduction

RELIABLE power distribution systems are essential to deliver electric power from generating stations to customers. A failure in any component of a distribution system will cause a power interruption. Since a very large part of the distribution systems in the world consist of overhead feeders with radial configuration, failures on distribution systems are responsible for a large part of the interruptions experienced by customers. It is therefore essential to incorporate an effective maintenance schedule for these overhead feeders, which includes the ability to predict failure rates based on historical data. Wind, lightning, ice, tornadoes and other weather are some of the factors that could cause line failures. High tree density near a line or lack of regular tree trimming can also cause frequent interruptions.

In a previous work, an adaptive fuzzy inference system approach is used for failure rate prediction of overhead lines [1, 2, 3]. The present work proposes the use of radial basis function (RBF) neural networks for failure rate prediction. RBF networks are selected for the model, as the data set that we have considered is of low dimensionality. RBF neural networks are particularly well suited for function interpolation in these cases, and have consistently outperformed other approaches in a variety of tasks [4, 5]. Further motivation is drawn from the fact that RBF networks are functionally equivalent to fuzzy logic systems [6] which are already shown to predict failure rates with a small error rate [3]. As these networks make use of units having highly localized responses, called kernels, many methods exist to train such networks that use clustering [4]. Evolutionary algorithms [7] have also been a popular choice amongst researchers [8, 9, 10, 11, 12, 13]. Additionally, an inbuilt smoothing mechanism is provided by Gaussian shaped kernels within the RBF networks (see later).

For failure rate prediction, it is especially important to accurately predict failure rates of transmission lines that are more prone to failures. Unlike local algorithms, evolutionary approaches can easily place kernels outside the convex hull of the input data set, thereby reducing edge effects making room for more accurate predictions in these boundary cases. Typically, RBF networks use a large number of kernels in their hidden layer. However, to avoid over-fitting of the training data, a parsimonious network is always desirable. In this paper, a multi-objective optimization approach has been adopted with the number of kernels being a second objective (in addition to the error). Multi-objective optimization [14, 15, 16, 17, 18] is particularly well suited in our case as historical data of failure rates is extremely difficult to obtain. Moreover, preprocessing the data is a cumbersome task [2]. A multi-objective approach yields a number of RBF networks of varying sizes, and after the training has been accomplished using training data, the best network can be selected using a separate test data set.

A. Previous Work

Previous work on the prediction of the failure rates of overhead lines has used an adaptive fuzzy inference system approach [3]. The data to train the model has been gathered from various sources, including a large utility company based in the western United States, and the National Climatic Data Center, for a seven year period, 1990-1996. A total of 37 feeder sections are involved. The input data into this system consisted of four fields pertaining to weather related local data. For further details, see Section 4. The output data, used for training, is the failure rate, i.e. the average number of

Manuscript received March 5, 2007. This work was supported in part by the U.S. National Science Foundation under Grant NSF ECS 0501288. This work is based on an earlier work: A Pareto Archive Evolutionary Strategy Based Radial Basis Function Network Training Algorithm for Failure Rate Prediction in Overhead Feeders, in Proceedings of GECCO 2005 © ACM.

Grant Cochenour is with Oklahoma Gas & Electric (e-mail: cochengr@oge.com).

Sanjoy Das and Anil Pahwa are with the Department of Electrical and Computer Engineering, Kansas State University, Manhattan, KS 66506 USA (e-mails: sdas@ksu.edu, pahwa@ksu.edu).

Jerad Simon is with Sandia National Lab (e-mail: jlsimon@sandia.gov).

failures per mile of feeder, per year. 75% of the data is used to train the fuzzy model, while the rest is use to evaluate its performance. For further details, one is referred to [2]. *A priori* expert domain knowledge (such as the fact that more lightning will result in an increased number of failures) is effectively used to design a fuzzy rule base. The rule base is further fine-tuned using a gradient descent algorithm to minimize the mean square error between the model's predicted failure rate, and the actual values obtained from the data.

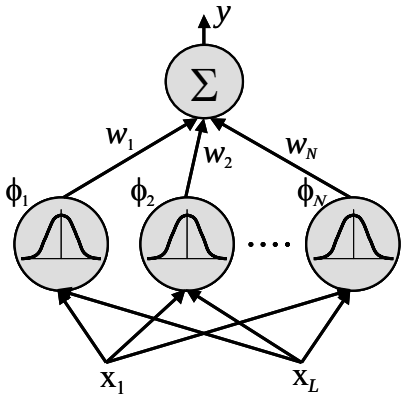


Fig. 1. A Radial Basis Function Network.

II. BACKGROUND

A mathematical description of the RBF network is outlined in this section, followed by an introduction to multi-objective optimization, and the Pareto Archive Evolutionary Strategy algorithm.

A. Radial Basis Function Networks

RBF networks considered in this paper are capable of performing mappings from an *n*-dimensional input $\Re^{\mathbf{n}}$ to a one-dimensional output. The network consists of a set of *N* nonlinear elements with local responses, called kernels. For an input $\mathbf{x}(q)$, q=1, 2, … Q, drawn from a set of Q samples, the response of the i^{th} kernel is given by,

$$\phi_i(\mathbf{x}(q)) = e^{-(\mathbf{x}(q)-\mu_i)^2/\sigma_i^2}. \quad (2)$$

In the above equation μ_i and σ_i are parameters associated with each kernel, usually referred to as its location and width. The network output is a linear combination of the kernel responses,

$$y(q) = \sum_{i=1}^{N} w_i \phi_i(\mathbf{x}(q)), \quad (3)$$

where each w_i is a weight linking the kernel to an output summing unit. Arranging the kernel responses as a $Q \times N$ matrix, $\mathbf{\Phi}$, whose $(q, i)^{\text{th}}$ element is $\phi_i(\mathbf{x}(q))$, and the weights and outputs as $N \times 1$ and $Q \times 1$ vectors, $\mathbf{w}$ and $\mathbf{y}$ we get,

$$\mathbf{y} = \mathbf{\Phi w}. \quad (4)$$

Suppose $\mathbf{t}$ is a vector of desired outputs, called targets, the mean square error between the actual outputs and targets is,

$$E = \tfrac{1}{2}(\mathbf{t}-\mathbf{y})(\mathbf{t}-\mathbf{y})^T \quad (5)$$

The optimal weight vector to minimize the mean square error, that can be directly obtained by equating ∇E to zero, is given by [4],

$$\mathbf{w} = \left(\mathbf{\Phi}^T\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^T\mathbf{t} = \mathbf{\Phi}^+\mathbf{t}, \quad (6)$$

where $\mathbf{\Phi}^+$ is the pseudoinverse of $\mathbf{\Phi}$. A number of RBF training methods apply orthogonal least square (OLS) techniques to the response matrix $\mathbf{\Phi}$. We postpone a fuller discussion on this approach until later.

B. Evolutionary Optimization

Evolutionary algorithms have emerged as one of the most popular approaches for the complex optimization problems. They draw upon Darwinian paradigms of evolution to search through the solution space (the set of all possible solutions). These algorithms maintain a large population of interacting individuals (also called chromosomes), where each individual represents a potential solution to the problem. This population is constantly pruned to maintain constant size by applying the Darwinian criterion of survival of the fittest, where the fitness of each individual reflects the degree of optimality of the solution. This operation is called selection. Mutation is another operation that is applied to the individuals. Mutation introduces genetic diversity to a finite population, thereby allowing the search process to explore new regions of the solution space. Evolutionary algorithms can be regarded as stochastic search through the solution space.

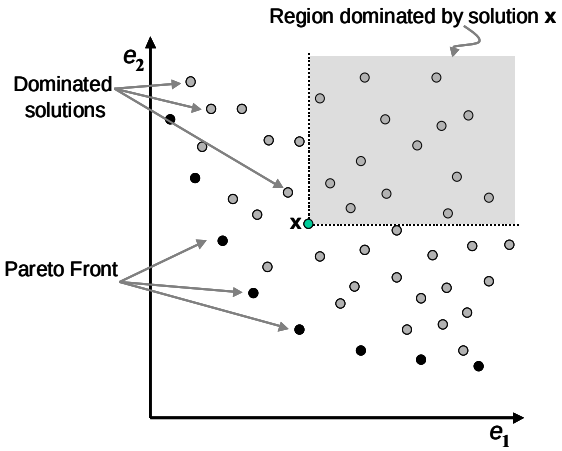


Fig. 2. A set of solutions with its Pareto front.

C. Multi-objective Optimization

When dealing with optimization problems with multiple objectives, the conventional concept of optimality does not hold good [11, 12, 13, 14]. Hence, the concepts of dominance and Pareto-optimality are applied. Without a loss of generality, if we assume that the optimization problem involves minimizing each objective $e_j(\cdot), j = 1, 2, ..., M$, a solution u is said to dominate over another solution v if and only if $\forall j \in \{j = 1, 2, ...M\}$, $e_j(u) \le e_j(v)$ with at least one of the inequalities being strict, i.e. for each objective, u is better than or equal to v and better in at least one objective. This relationship is represented as $u \prec v$. In a population of solution vectors, the set of all non-dominating solutions is

called the Pareto front. In other words, if S is the population, the Pareto Front, Γ, is given by,

$$\Gamma = \{u \in S \mid \forall v \in S, \neg(v \prec u)\} \quad (7)$$

Figure 2. shows a set of solutions in the objective function space of two objectives e_1 and e_2 and the region dominated by an example solution.

The simplistic approach of aggregating multiple objectives into a single one often fails to produce good results. It produces only a single solution. Further complications arise when the objective function space is concave. Multi-objective optimization on the other hand involves extracting the entire Pareto front from the solution space. In recent years, many evolutionary algorithms for multi-objective optimization have been proposed [14, 15, 16, 17].

Unlike single-objective evolutionary algorithms, where the only goal is to find a feasible optimum with a reasonably low computation, in order to be effective, multi-objective approaches such as PAES have to address two other critical issues when computing the Pareto front, which are as follows. (*i*) *Diversity maintenance*: An uneven, distribution of sample solutions with far too many concentrated in some "crowded" region and with uncovered regions elsewhere is an undesirable property in multi-objective approaches. Hence, the algorithm must preserve diversity, *i.e.* ensure that the samples delivered from the Pareto front of the solution space occur at approximately evenly spaced intervals. Figure 2. shows a Pareto front with a very desirable distribution of solutions. (*ii*) *Archiving*: Multi-objective problems must preserve all non-dominated solutions found until the algorithm stops, unless either another dominating solution is found, or the solution is located at a densely populated region of the front to be of much interest. This is accomplished within the algorithm by maintaining an archive of solutions. New non-dominated solutions, when found, are inserted into this archive. Conversely, when a dominating solution is found to an already archived solution, the latter is removed from the archive. In some cases it is necessary to prune the archive in order to prevent it from getting too large. However, this feature is unnecessary in the present application.

D. Optimization of Radial Basis Function Networks

Evolutionary algorithms have been very popular in determining the parameters of RBF networks. A genetic algorithm is used in [9] to evolve the network for function approximation. The genetic algorithm population consisted of individual kernels instead of entire networks. The kernel locations and widths are evolved, while the weights are determined separately. The network size is a constant. A similar strategy has been used in [8] which makes use of an OLS procedure and singular value decomposition to evaluate the contribution of individual kernels. Other methods [e.g. 10] evolve an entire population of RBF neural networks. A two-level algorithm is proposed in [12] where a micro genetic algorithm at the upper level is used to evolve the kernel widths, while a lower level OLS process determines the other parameters. In [11] the number of kernels is also determined dynamically using a genetic algorithm. In the first application of multi-objective optimization to RBF training [13], the network size is the second objective to be minimized. The proposed method makes use of an OLS algorithm and singular value decomposition on the kernel response matrix to perform mutation.

```
Algorithm RBF_PAES
populate A with initial RBFs;
for each w in A that is dominated,
  A = A - {w};
endfor
t=0;
while t < maxIterations
   select u randomly from A;
   v = mutate(u);
   if u≺v
     do nothing;
   elseif v≺u
     A = A∪{v};
     for each w in A dominated by v
       A = A-{w};
     endfor
   else //neither v≺u nor u≺v
     if v≺w  for some w in A
       A = A∪{v};
       for each w in A such that v≺w
         A = A-{w};
       endfor
     else
       z = argmin(crowd(u), crowd(v));
       A = A-{u};
       A = A∪{z};
     endif
  t=t+1;
end
```

Fig. 3. The modified PAES algorithm used to train the RBF network.

PAES is selected as the training algorithm in this paper as it is arguably the simplest of all multi-objective optimization algorithms. In terms of algorithm complexity, it compares favorably with other popular approaches such as NSGA-II and SPEA [17, 19]. Furthermore, preliminary simulations by us suggested that applying crossover, one of the standard operators in evolutionary algorithms, failed to improve the convergence towards the Pareto front, while adding to computer time. Unlike genetic algorithms, evolutionary strategies rely almost solely on mutation, the other primary evolutionary operator. Keeping in view the general philosophy of simplicity in PAES, we have devised a simple but effective mutation operator to evolve accurately trained RBF networks.

III. APPROACH

A. Overview of the Algorithm

The proposed algorithm begins with by generating several RBF networks and picking out the non-dominated set as the initial archive. The network size, and the kernel location and width parameters of the initial RBF networks are generated

randomly, but the weights are computed using Equation (6). The archive records the size, as well as the kernel locations, widths and weights up to a limit. Within each iteration of the algorithm, a solution *u* is picked at random from the archive *A*. The solution is mutated to produce a mutant *v*. The mutant is then compared with the original solution. If the mutant dominates the original solution, *i.e.* $v \prec u$, *v* is inserted into *A*. Any archived solution *w*, including *u* that is dominated by *v* is also removed. On the other hand, if the mutant *v* is dominated by *u*, *v* is discarded. When neither dominate, *i.e.* neither $u \prec v$ nor $v \prec u$, the algorithm checks to see if the new mutant *v* dominates any other solution in the archive *A*. If so, then *v* is inserted into *A* while *u* as well as other solutions in *A* that are dominated by *v* are discarded. If none of these cases apply, then the algorithm uses the approach adopted by other recently proposed multi-objective evolutionary algorithms, which is to prefer solutions that lie in sparser regions of the non-dominated stored in the archive *A* [19]. The crowdedness may be estimated by first sorting the solutions according to their errors, and then considering the perimeter of a rectangle that encloses the solution, and whose corners are the solution's immediate neighbors in the sorted list [19]. For any RBF solution *v*, is given by,

$$\text{crowd}(v) = \frac{E_u - E_w}{\max(E) - \min(E)} + \frac{N_u - N_w}{\max(N) - \min(N)} \qquad (8)$$

In the above equation, *u* and *w* are *v*'s left and right neighbors in the sorted list, and $E_u, E_v,$ and $N_u, N_v,$ are the associated errors as given in Equation (5) and the number of kernels in the neighboring RBF networks. The min(.) and max(.) operators are carried out over the entire archive *A*. Extreme solutions which do not have either a left or a right neighbor are assigned ∞. In other words, they are the most preferred solutions by the algorithm.

An outline of the entire algorithm is provided in Figure 3.

B. Mutation

Mutation is invoked by applying the Gram-Schmidt orthogonalization algorithm to the kernel response matrix, $\mathbf{\Phi}$. Using this method $\mathbf{\Phi}$ can be factored into two matrices, $\mathbf{B}$ and $\mathbf{A}$ of size $Q \times N$ and $N \times N$, as,

$$\mathbf{\Phi} = \mathbf{BA} = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_i & \dots & \mathbf{b}_N \end{bmatrix} \mathbf{A}, \qquad (9)$$

where the columns of $\mathbf{B}$ are mutually orthogonal (i.e. $\mathbf{b}_i^T \mathbf{b}_j = 0$). Under these circumstances it has been shown in [5] that when the weights are optimally chosen based on Equation (6),

$$\mathbf{t}^T \mathbf{t} = \sum_{i=1}^{N} \frac{(\mathbf{b}_i^T \mathbf{t})^2}{\mathbf{b}_i^T \mathbf{b}_i} + \mathbf{e}^T \mathbf{e}, \qquad (10)$$

where $\mathbf{e}$ is a vector of errors. Each term in the summation is the contribution of the i^{th} kernel to the target and can be isolated. It is important to note that obtaining each $\mathbf{b}_i$ is dependent on the values of the previously computed $\mathbf{b}_j$ s, where $1 \le j < i$ in the OLS algorithm. Further details of this method can be found in [5].

The mutation operator acting upon a kernel of size *N* can either increase its size by one or two additional kernels, reduce it by removing up to two, or not alter the size of the RBF network at all. The mutation operator inserts three new candidate kernels into the RBF network and applies the OLS algorithm to determine the contribution of the *N* existing kernels, as well as the three new ones if each are to be individually added to the network. Out of the $N+3$, candidate kernels, the operator retains the $N-2$, $N-1$, N, $N+1$, or $N+2$, kernels with the highest contribution as given by Equation (9), discarding the remainder, each of the five outcomes being equally probable.

The kernel widths are computed using the standard nearest neighbor heuristic,

$$\sigma_i = \min_j \left(\left\| \mu_i - \mu_j \right\| \right) \Big/ \sqrt{N}, \qquad (11)$$

where min is the minimum operation. The kernel locations are subjected to small perturbations from their original locations by adding uniformly distributed random vector,

$$\mu_i = \mu_i + \text{rnd}(-\varepsilon, \varepsilon). \qquad (12)$$

In the above equation, rnd(.) generates random vectors whose elements lie in the range, $(-\varepsilon, \varepsilon)$ provided as its argument. The kernel weights are determined optimally using Equation 6.

IV. Results & Discussion

A. Experimental Setup

The proposed algorithm is implemented in MATLAB Release 14. The maximum archive size is set to 20. The archive is initialized with 10 non-dominated networks of with random number of kernels, *K*, and random kernel centers with the widths determined by Equation 10. The number of kernels, *K*, for each candidate solution is limited to the range of 4 to 50 for the duration of the experiment. The nearest neighbor heuristic in Equation 10. has a uniformly distributed random scaling factor on the interval [1, 10]. The limits of the perturbation described in Equation (12) for each kernel in a solution, μ_i is always less than the kernel's width, σ_i as in [13]. The selection of a solution from the archive for mutation is biased towards larger networks because larger networks take more time to converge. The algorithm is run for 5000 iterations. A total of three separate sets of simulations is carried out using the following data (*i*) California Data, (*ii*) Kansas A Data, and (*iii*) Kansas B Data. In each case, the performance of the proposed algorithm is compared with multilayered perceptrons (MLPs) with a single hidden layer which is trained using the standard back-propagation algorithm [23]. The MLPs are trained with the backpropagation algorithm with a momentum term using the standard MATLAB neural network toolbox. The best learning rate and momentum for the MLP are applied separately in each case, which are significantly better than the default values the toolbox provides. The MLP training is halted after the error for the test data shows no further improvement. Additionally, the California data is compared with the results in [3]. The error reported here is the standard RMS error, which is $\sqrt{E}/Q$, where *E* is computed using Equation (5) and *Q* is number of training samples.

B. Performance with California Data

This data is obtained from [2] as described in Section 1. It is broken up in the same manner as Case 4 from [2] into test and training data sets. The data sets have 4 input variables and 1 output. The four input fields are, (*i*) the lightning rate, i.e. the number of days experiencing lightning each year, (*ii*) the tree density, i.e. the number of trees per mile length of feeder-section, (*iii*) the years since tree trimming is carried out in that area, and (*iv*) the wind index, which incorporates information on the wind velocity as well their duration. The training data set has 1775 samples, and the test data set has 591 samples. This data is then normalized. An earlier set of results for this data has been presented elsewhere [24]. Figure 4. shows the histogram of the output, the failure rate, which is measured as the total number of failures per mile of transmission line length per year.

The training data is used in the algorithm, and the test data is used to verify the generality of the resulting solutions. In Figure 5. is shown the non-dominated solutions in the archive evolving with increasing number of iterations for the training samples. It is seen that the error of the untrained networks are quite low. This is because the weights of the network are optimal, having already been determined by Equation 6 and the kernel widths have been computed using the heuristic in Equation 11. PAES determines the locations of the kernels, the σ_i s. Figure 6. contains the Pareto front obtained by the proposed method for the test and training data. A curious feature in the figure, which has been noted earlier in [1, 2] is the mean square error for the test data being lower than that of the training data, which is used explicitly to train the RBF networks.

TABLE 1
COMPARATIVE PERFOMANCES FOR THE CALIFORNIA DATA SET

	RMSE Training	RMSE Test
Proposed Algorithm (N=5)	0.110	0.093
Proposed Algorithm (N=10)	0.103	0.085
Proposed Algorithm (N=16)	0.100	0.081
FIS	0.142	0.121
MLP (4x2x1)	0.129	0.123
MLP (4x3x1)	0.122	0.119
MLP (4x4x1)	0.118	0.114

Table 1. provides a comparison between the proposed method and other techniques used. Specifically, the PAES trained RBF network approach is compared against the fuzzy inference system approach in [2] and also against standard multi-layered perceptrons (MLPs) with various network sizes. The learning rate and momentum constants are 0.1 and 0.5 respectively. With these parameters, the MLPS did better than the default values, and converged in 10,000 iterations. The results reported in Table 1 are the averaged results over 10 trials.

The proposed algorithm outperforms both the MLP and the fuzzy inference approaches. The runtime for the proposed algorithm is 2127 seconds, and for the three MLP models collectively, the runtime is 2640 seconds. Larger MLPs showed no significant improvement in the performance for the training data, while the error for the test data only increased.

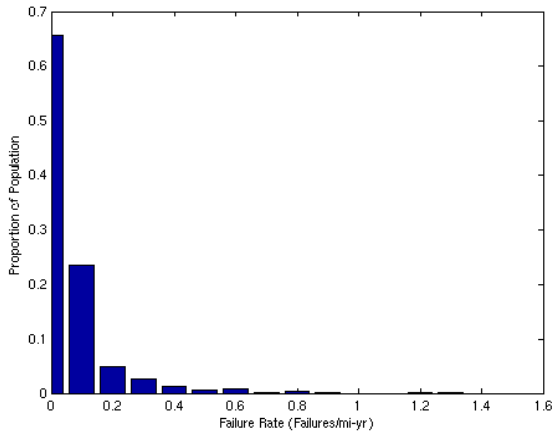


Fig. 4. Histogram of the failure rates for California

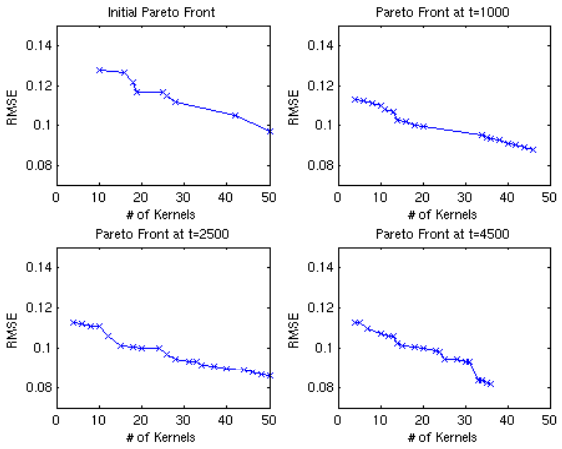


Fig. 5. Evolution of the non-dominated front for California.

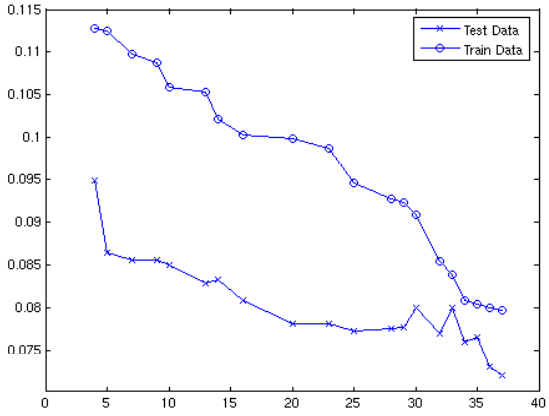


Fig. 6. RMS errors of the trained RBF nets for the test and training data for California.

C. Performance with Kansas Data A

This data is obtained from a private electrical utility company located in the mid-western region of the US. The data set

contains two predictors, (*i*) the lightning rate, which in this case is the sum of the magnitudes of the lightning current for each day whose range is between 0 and 65341 kA, and (*ii*) maximum daily gust speed, where gusts lasting for more than five seconds are recorded. The range of this second field lies in the range 0 to 68 mph. A third field is introduced, which is the product of the lightning rate and maximum gust speed. The output is the total daily number of outages experienced by the entire system, the maximum of which is 45 per day. The entire data has 2157 samples. Its output histogram is shown in Figure 7.

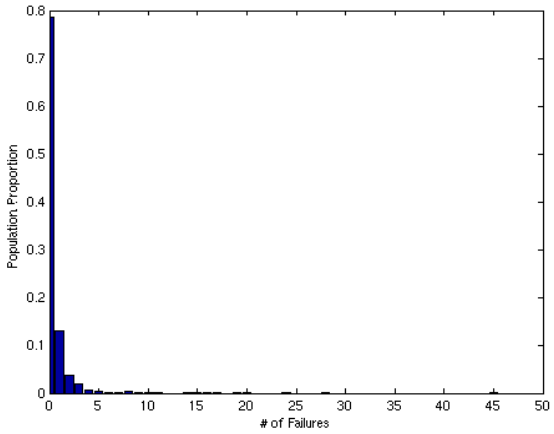


Fig. 7. Histogram of the failure rates for Kansas A.

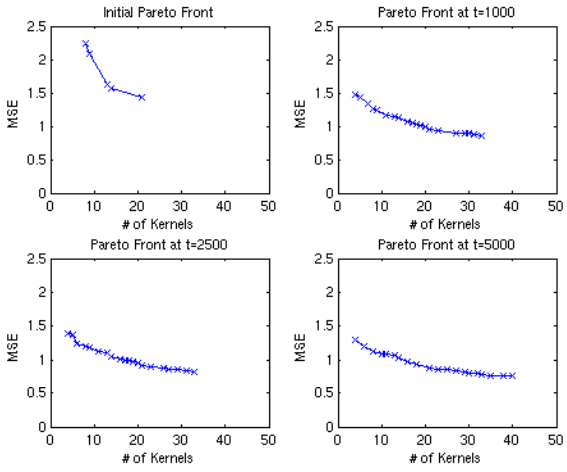


Fig. 8. Evolution of the non-dominated front for Kansas A.

TABLE 2
COMPARATIVE PERFOMANCES FOR THE KANSAS A DATA SET

	RMSE Training
Proposed Algorithm (N=4)	1.29
Proposed Algorithm (N=10)	1.10
Proposed Algorithm (N=15)	0.98
MLP (3x3x1)	3.56
MLP (3x5x1)	1.75
MLP (3x7x1)	1.39
MLP (3x9x1)	1.30

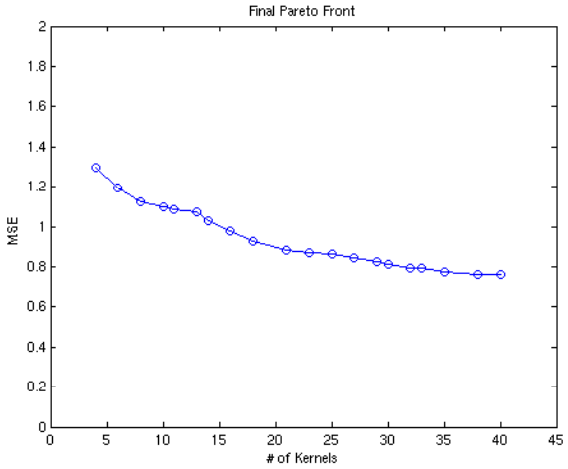


Fig. 9. RMS errors of the trained RBF nets for data in Kansas A.

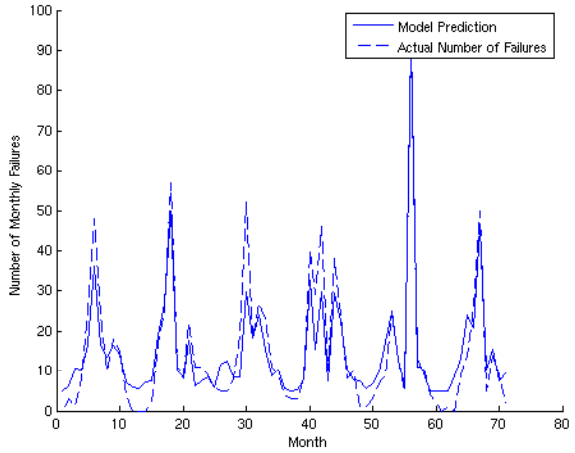


Fig. 10. Comparison of the daily model and actual failure rates for Kansas A.

In Figure 8. is shown the evolution of the non-dominated front at four different intermediate stages of the PAES training process. In the next figure (Figure 9.) we provide the Pareto front obtained at the end of the training process. Since Kansas A data samples are provided on a daily basis, it is possible to compare the daily predicted failure rate with the real data. This is shown in the form of two time series in Figure 10. Finally, in Table 2 is provided a comparison with sample MLPs, again averaged over 10 trial runs. For larger sized MLPS, there is no further improvement in the error. The optimal learning rate and momentum constants are 0.1 and 0.5 respectively, and a total of 10,000 iterations of MLP training is allowed. It is clear that the proposed algorithm outperforms the MLPs.

D. Performance with Kansas Data B

This data is also obtained from a private electrical utility company from the mid-western USA. The data pertained to the annual failures of individual line sections. A line section is defined as the conductor covered by a specific protective

device (fuse, recloser, or breaker). The data contained four input classes, which are (*i*) the tree density, i.e. the number of trees per mile in the vicinity of the transmission line, which ranged between 0 and 63.5 trees/mile, (*ii*) the years since last trimming of the trees, the maximum being 7 years, (*iii*) the wind data, and (*iv*) the lightning data. The wind index and lightning index data contained several sub-fields each. The data related to lightning contained (*i*) average lightning per day of each year (kA/day), (*ii*) the maximum daily lightning for each year (kA), and (*iii*) and the number of days each year the lightning is above 1 kA. Likewise, the wind data consisted of the following sub-fields, (*i*) average maximum daily gust speed, where gusts lasting for more than five seconds are recorded and averaged for the entire year, (*ii*) the number of days each year the wind speed is more than 30 mph, (*iii*) the number of days each year the wind speed is more than 40 mph, and (*iv*) the number of days each year the wind speed is more than 50 mph. Both the lightning and wind data are reduced to a single field each by means of principle component analysis. The ranges obtained are 0-4.5 for the wind index and 0-3.75 for the lightning index. Hence in all, there are four input predictors. The output histogram is shown in Figure 11. The data is divided into separate training and test sets.

Figure 12. illustrates the evolution of the non-dominated front at different stages of the proposed training algorithm, and Figure 12. has the final Pareto front at the end of the training. Lastly, Table 3 contains a comparison with sample MLPs (10 run averages). The optimal learning rate and momentum constants in this case are 0.05 and 0.5 respectively, and a total of 10,000 iterations of MLP training are allowed. The mean square errors of the RBF networks are clearly better than those of the MLPs, although for the test data, the error for $N = 15$ is too close to the perceptron with 7 or 9 hidden neurons.

TABLE 3
COMPARATIVE PERFOMANCES FOR THE KANSAS B DATA SET

	RMSE Training	RMSE Test
Proposed Algorithm (N=4)	0.171	0.144
Proposed Algorithm (N=7)	0.165	0.144
Proposed Algorithm (N=15)	0.158	0.155
MLP (4x5x1)	0.210	0.165
MLP (4x7x1)	0.179	0.151
MLP (4x9x1)	0.171	0.152

V. CONCLUSION

In this paper a modified version of the PAES algorithm has been presented for the optimization of RBF networks used in the prediction of overhead distribution failure rates. The proposed algorithm is simple and outperforms other methods in the prediction of failure rates on overhead distribution feeders within comparable computer time. The choice of RBF networks as the tool for modeling failure rates is justified as RBF networks have a solid mathematical foundation in Cover's theorem and are universal approximators [25].

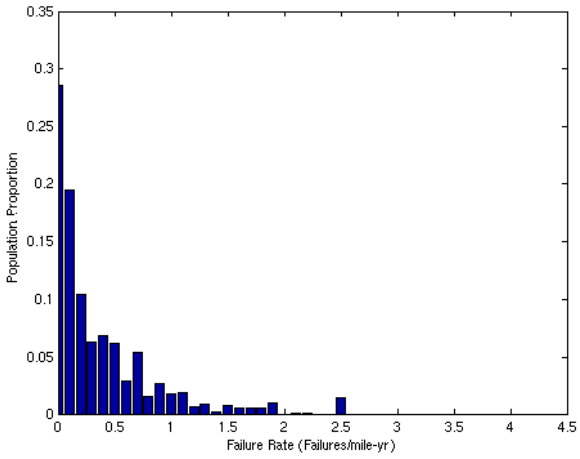


Fig. 11. Histogram of the failure rates for Kansas B.

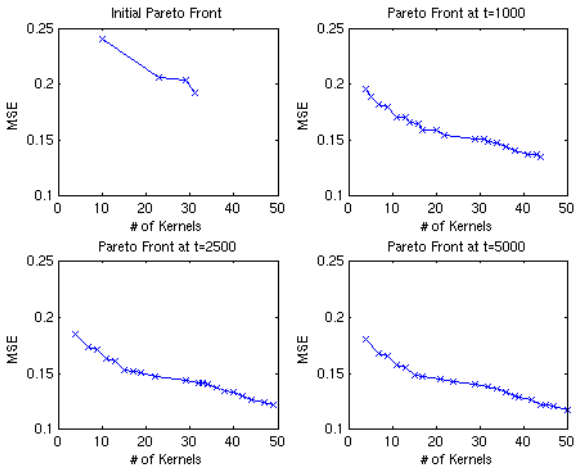


Fig. 12. Evolution of the non-dominated front for Kansas B.

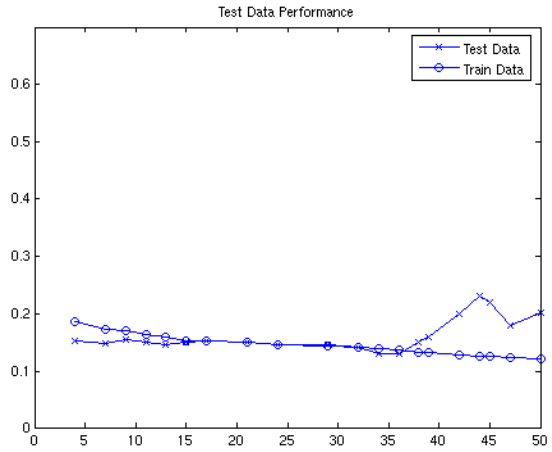


Fig. 13. RMS errors of the trained RBF nets for the test and training data for Kansas B.

A distinguishing feature of the method is that it accomplishes the simultaneous training of RBF networks of multiple sizes. Because the mutation allows network sizes to be changed within the PAES algorithm, new networks that are created are allowed to benefit from the training received by networks of other sizes. This allows multiple networks to be evolved simultaneously, curtailing the additional computer time that would have been necessary to train the RBF networks independently of each other.

Multiple training of RBF networks is a feature that can prove to be highly useful in the present application. First, although an arbitrarily large number of kernels reduces the error with respect to the training data, beyond a certain point it would be overfitting the data. Simultaneously trained networks of various sizes would provide a clear picture of the exact number of kernels necessary for the prediction task. Secondly, using multiple networks opens up the possibility of using ensemble methods, methods that combine multiple models for regression or classification to improve the overall reliability of the prediction [26, 27]. Finally a large number of predictions would allow a utility service company to make human decisions relating to installation, maintenance and repair of distribution systems.

VI. References

[1] S. Gupta, A. Pahwa, R.E. Brown, S. Das, "A Fuzzy Model for Overhead Distribution Feeders Failure Rate", *Proceedings*, *34th Annual North American Power Symposium*, Tempe, Arizona, 2002, pp. 248–253.

[2] A. Pahwa, S. Gupta, Y. Zhou, R.E. Brown, S. Das, "Data Selection to Train a Fuzzy Model for Overhead Distribution Feeders Failure Rates", *Proceedings*, *IEEE 12th Intelligent Systems Applications to Power Systems Conference*, Athens, Greece, 2003.

[3] S. Gupta, A. Pahwa, R. E. Brown, Y. Zhou, S. Das, "An Adaptive Fuzzy Model for Failure Rates of Overhead Distribution Feeders", *Electric Power Components and Systems*, Vol. 33, 2005, pp. 1175– 190.

[4] D.S. Broomhead, D. Lowe, "Multivariate functional interpolation and adaptive networks," *Complex Systems*, Vol. 2, 1988, pp. 321-355.

[5] S. Chen, C.F.N. Cowan, P.M. Grant, "Orthogonal least squares learning for radial basis function networks," *IEEE Transactions on Neural Networks*, Vol. 2, no. 2, 1991, pp. 302-309.

[6] J.-S. R Jang., C.-T. Sun, **"**Functional equivalence between radial basis function networks and fuzzy inference systems", IEEE Transactions on Neural Networks, Vol. 4, no. 1, 1993, pp. 156-159.

[7] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, Reading, MA, 1989.

[8] B.A. Whitehead, "Genetic Evolution of Radial Basis Function Coverage Using Orthogonal Niches," *IEEE Transactions on Neural Networks*, Vol. 7, no. 6, Nov 1996, pp. 1525-1528.

[9] B.A. Whitehead, T.D. Choate, "Cooperative-Competitive Genetic Evolution of Radial Basis Function Centers and Widths for Time Series Prediction," *IEEE Transactions on Neural Networks*, Vol. 7, no. 4, July 1996, pp. 869-880.

[10] S.A. Billings, G.L. Zheng, "Radial basis function network configuration using genetic algorithms," Vol. 8, no. 6, 1995, pp. 302-309.

[11] A. Alexandridis, H. Sarimveis, G. Bafas, "A new algorithm for online structure and parameter adaptation of RBF networks," *Neural Networks*, Vol. 16, no. 5-6, June 2003, pp. 529-535.

[12] S. Chen, Y. Wu, B.L. Luk, "Combined Genetic Algorithm Optimization and Regularized Orthogonal Least Square Learning for Radial Basis Function Networks," *IEEE Transactions on Neural Networks*, Vol. 10, no. 5, Sept 1999, pp. 1239-1243.

[13] J. González, I. Rojas, J. Ortega, H. Pomares, A. Díaz, "Multiobjective Evolutionary Optimization of the Size, Shape and Position Parameters of Radial Basin Function Networks for Function Approximation," *IEEE Transactions on Neural Networks*, Vol. 14, no. 6, Nov 2003, pp. 1478-1498.

[14] C.M. Fonseca, P.J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary Computation*, Vol. 3, no. 1, Spring 1995, pp. 1-16.

[15] C.A. Coello Coello, "A comprehensive survey of evolutionary-based multiobjective optimization techniques," *Knowledge and Information Systems*, Vol. 1, no. 3, Aug 1999, pp. 269-308.

[16] D.A. Van Veldhuizen, G.B. Lamont, "Multiobjective evolutionary algorithms: Analyzing the state-of-the-art," *Evolutionary Computation*, Vol. 8, no. 2, 2000, pp. 125-147.

[17] E. Zitzler, L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, Vol 3., no. 4, 2002, pp. 257-271.

[18] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V.G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Transactions on Evolutionary Computatio*n, Vol. 7, no. 2, Apr. 2003, pp. 117-132.

[19] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computatio*n, Vol. 6, No. 2, pp. 182-197, 2002

[20] L. Knowles, D. Corne, "Approximating the nondominated front using the Pareto Archived Evolution Strategy**,"** *Evolutionary Computation*, Vol. 8, no. 2, pp. 149-172, 2000.

[21] J. Knowles, D. Corne, "Properties of an adaptive archiving algorithm for storing nondominated vectors**,"** *IEEE Transactions on Evolutionary Computation*, Vol. 7, no. 2, Apr. 2003, pp. 100-116.

[22] J. Knowles, D. Corne, "The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Pareto Multiobjective Optimisation**,"** Proceedings *of the 1999 Congress on Evolutionary Computation*, 1999, pp. 98-105.

[23] D.E. Rumelhart, J. L. McClelland, *Parallel Distributed Processing, Vols. 1 and 2*, Cambridge, MA, The M.I.T. Press, 1986.

[24] G. Cochenour, J. Simon, S. Das, A. Pahwa, S. Nag, "A Pareto Archive Evolutionary Strategy Based Radial Basis Function Network Training Algorithm for Failure Rate Prediction in Overhead Feeders", *Proceedings of the Genetic and Evolutionary Computing Conference*, Washington, D. C., 2005, pp. 2127-2132.

[25] J. Park, I.W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Computation*, Vol 5., pp. 305-316, 1993.

[26] Breiman, L., "Bagging Predictors", *Machine Learning*, Vol. 24, No. 2, pp. 123-140, 1996.

[27] R.E. Schapire, "A brief introduction to boosting", *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 1999.

ISAST Transactions on Intelligent Systems, No. 1, Vol. 1, 2008
M. Bastanfard and S. D. Katebi: Optimization and mutual information through machine learning technique

Optimization and mutual information through machine learning technique

M. Bastanfard and S. D. katebi

Abstract— Sophisticated agents operating in open environments must make complex decisions on scheduling and coordination of domain activities. These decisions are made in the context of multiple conditions and mutual information.

The question of how to deal with and consider all possible activities without consuming too many time, is the main problem for an intelligent agent. The focus of this research is to provide effective social intelligence and improved performance of individual agents in a cooperative multi-agent system. This is done by decisions made by the agents using reinforcement learning methods. The agents have a simple reactive and local perception and learn in a decentralized way to overcome the difficulties of reinforcement learning. A game frame work that supports detailed reasoning about combinational optimization and coordination costs; for testing and comparing the performance of agents is performed.

Keywords— Combinational optimization, multi agent, mutual information, reinforcement learning.

I. Introduction

Reinforcement learning(RL) has been of a great interest in research and application areas for many years [21].

Different RL algorithms mostly deal with only one agent [20]. Recently there has been growing interest in extending RL to the multi agent domain where autonomous agents interact with each other [7].

Learning the Multi Agent Systems (MAS) can indeed take many forms [14]. In many cases it is very promising to use RL as it don't require a teacher who knows the best decisions, it just needs feedback from environment as a scalar value for example, that forms a reward or punishment to actions that has been done by the agents.

We consider Multi-Agent Systems composed of simple reactive situated agents with local perceptions that try to solve the problem in a decentralized way [13], which is easier to build and can be the case in real problems as most often global knowledge is not available for each individual agent. Whereas RL has an optimizing inheritance, it can be used in optimization problems. Combinational optimization is one of complex open problems in which one or more condition should be satisfied simultaneously. Besides, these kinds of problems might be involved by mutual information [17] restriction. Many efficient architectures and algorithms that support these restrictions have been developed. However, none of these architectures explicitly reason about the consumption of time to attain an optimal solution, which may degrade algorithm's performance. In classical methods for solving the combinational optimization problems, such as integer programming [9][6], the little changes in the conditions, can cause a large amount of necessary changes in the model and the solution. In addition, when the dimension increases the solution becomes unreliable even sometimes becomes impossible. Admiringly in reinforcement learning, just setting the reward function is sufficient and no change in other parts of algorithm is needed

As RL suffers from combinatorial explosion when there are a large number of states in the system, using decentralization is beneficial, where we have simple agents with local perception, each learns its own behavior by itself, state space of each agent is not too large and we can avoid this computational bottleneck. However, by using this approach we encounter some problems; Partial observability and Non-stationary.

To deal with these problems we used decentralized learning algorithm that is based on RL techniques for finding stochastic policies. Actually, agents with stochastic behaviors are more adapted to the problem of partial perception.

This paper is organized as follows. Section 2 introduces a necessary vocabulary and gives an overview of reinforcement learning. In Section 3, problem and proposed algorithm are explained in detail. Section 4 gives results and comparisons.

II. Frame Work

In this part reinforcement learning problems in a multi agent system is shown. Then the profit-sharing [8][10] in reinforcement learning is explained. At last the kind of agents is described which are considered in a game frame work for ignoring the difficulties of the real word.

A. Reinforcement learning

Reinforcement learning methods are very useful ways for learning optimal memory-less behaviors for agents as they only require a scalar feedback from the system to the agents for them to learn. Besides, these techniques can be used when there is uncertainty in the world's evolution. But the convergence of RL algorithms (such as Q-Learning [2] and TD [16]) has only been proven for Markov Decision Processes (MDP). An MDP can be viewed as a stochastic

Manuscript received April 1, 2007. optimization and mutual information through machine learning technique.

M. Bastanfard, and S.D. Katebi are with Computer Science and Engineering department of Shiraz University, Shiraz, Iran (fax: +98-711-6271747; e-mail: bastanfard@cse.shirazu.ac.ir, katebi@shirazu.ac.ir).

automaton in which an agent's actions influence the transitions between states, and rewards are obtained depending on the states visited by an agent. In a fully observable MDP, it is assumed that the agent can determine its current state with certainty. Formally, a fully observable MDP —that is, the agent knows the true state at each time (once that stage is reached) — can be defined as a tuple <S,*A,T,R*>, where *S* is a finite set of states or possible worlds, *A* is a finite set of actions, *T* is a state transition function, and *R* is a reward function. The agent can control the state of the system to some extent by choosing actions from the set *A* that influence state transitions: movement from the current state to some new state. Actions are stochastic in that the actual transition cannot generally be predicted with certainty. The transition function *T* describes the effects of each action at each state. *T(s,a,s')* is the probability of ending up in state *s′* when action *a* is performed at state *s*. After each transition, the environment generates a reward given by *R(s,a)*. The problem is to find the optimal mapping *π(s,a)* between states and actions so as to maximize the reward received over time. Such a mapping is called a policy.

When the system is fully observable its choice of action can be based solely on the current state. Thus uncertainty lies only in the prediction of an action's effects, not in determining its actual effect after its execution. As pointed out by [3] the evolution of the kind of MAS we are interested in is a MDP. As such, it could be solved using classical reinforcement learning algorithms where the state of the system is the composition of the states of all agents and an action is a joint action composed of all individual actions of the agents. Thus, the number of states and actions in a centralized view of the problem should quickly prove to be too big for RL to be applied. Solving our problem in a non centralized way would mean to solve it in a decentralized solution as each agent should learn by itself. As is noted before we face two major difficulties:

• Non-stationary world: Reactive agents with a local view of the environment can not use joint actions to solve the problem. In fact, other agents are unpredictable elements of the environment while they are learning too. Moreover transitions from one state of the system to another, as seen by an agent, are non-stationary. For example the probability for an agent to corporate in a team depends greatly on the decisions of other adjacent agents.

• Partial observability*:* Agents can only rely on their own imperfect, local and partial perception of their world. The global state of the system is hidden, which prevents classical RL algorithms from finding a globally optimal policy [18]. As such, the problem at hand belongs to the class of partially observed Markov decision models [12].

Classical stationary Partially Observed Markov Decision Processes are nearly impossible to solve when there are more than a hundred of states [4]. The combination of the two problems (i.e non-stationary and partial observation) makes the problem non-solvable without using approximations. In the context of partially observable MDP, stochastic behaviors perform better than deterministic policies [18].

B. Profit sharing approach

The most important difference between profit-sharing approach and the DP-based reinforcement learning algorithms, such as Q-learning and Temporal Difference, is that profit-sharing does not use one step backup and not need eligibility trace to treat a delayed reward. Therefore, it is robust against the problems due to the existence of multiple agents, such as concurrent learning and perceptual aliasing. In addition, it can save the required memory space because it does not need to keep eligibilities and whole state-spaces which the agent experienced.

First, when an agent observes current state O_t , it checks its look up table to search the matched state as O_t and gets its action set A_t , which consists of available actions at time t. the action selection make agent behave under the stochastic policy based on exploration. After agent outputs the selected action a_t it checks if a reward is given or not. If there is no reward after an action a_t , the agent stores the state-action pair (O_t , a_t) into its episodic memory as a rule, and continues the same cycles until getting the reward R. the period from the start to the getting R, is called an episode.

Second, when the agent got the reward R, it reinforces rules which are stored in the episodic memory according to a credit assignment function f(R, t) which satisfies the "rationality theorem [10]. For example, the geometrically decreasing function (1) in which T: time at goal, t=0: time at initial state, is satisfied this theorem.

$$f = R \times (1/(L-1))^{T-t} \tag{1}$$

The gist of this theorem is that the reward should not be given to an ineffective action which makes agent move in a loop path more than to the effective action which makes the agent move straight.

C. Agents

We used very simple agents for complexity reasons. It allows us to concentrate on the learning aspect of the problem. Our agents share the same goal and they have to coordinate to reach the common goal. Our agents don't have any explicit communication with each other and they have their own imperfect and vague local perception of the environment.

III. PROBLEM DESCRIPTION

The task chosen involves agents in a 30×30 grid world whose goal is to save the victims. The victims are distributed randomly in the plane. Three agents as a team are needed for saving one of the victims. The goal is to save the victims as soon as possible.

Our agents have to select two other agents between their closest neighbors all around, to join together to form a team. Two agents A and B are close neighbors if there is a closed disc that contains A and B on its boundary and does not contain any other agent [11] (Fig. 1).

Selecting and corporation in possible teams which can be assumed as triangles, is the action set of each agent. The teams with the best performance are desirable and no agent can

corporate in more than one team. Performance test for a team is that has, the highest density of victims in its effective area according to minimize the maximum median of triangles as overhead time. Overhead time is the time which is consumed till the members of a team come together to attain the ability of saving the victims. The area which is bound up to half distance between the vertices of the team and their closest neighbors is the effective area of this team.

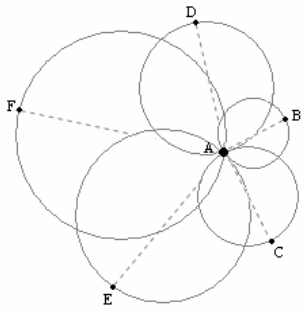


Fig. 1 B, C, D, E and F are the closest neighbors of A.

It makes the consequence of agent's actions stochastic depending on global state of the world and the sequence in which agents have been considered to do their actions. As stated before, each agent has its own perception of the world, agent perceptions are:

• Direction of victims which are located in area bound up to the closest neighbors of itself (Front, Left, Right, Back).
• Direction of other agents (Front, Left, Right, Back).
• Distance from closest agents all around.

Also the agents can talk to each other and share just the distance and direction of their closest neighbors.

A. Learning algorithm

Each agent should be joined in a team which is the final state for each one. So each agent with individual perceptions first, selects one of possible actions from the set which contains all its closest neighbors to be the second member of the team (see Fig. 2). The action is selected by the roulette selection, soft greedy method [20], in which the selection rate of the action is in proportion to its current value. This selection method makes agent behave under the stochastic policy and explore its strategy. Each action's value is initialized by zero.

$$\Pi_{\theta}(s,a) = e^{\theta(s,a)} / \sum_{b \in A} e^{\theta(s,b)} \qquad (2)$$

Next, the action set contains closest neighbors of the first and second team's member (see Fig. 2).

The rates of all teams' performance are accumulated in a variable such as R. In the global consideration, the arrangement is rewarded by R and penalized by -1 for each common agent in one more team up to two corporations (It is clear, if there is three common members in two teams they join single team admittedly).

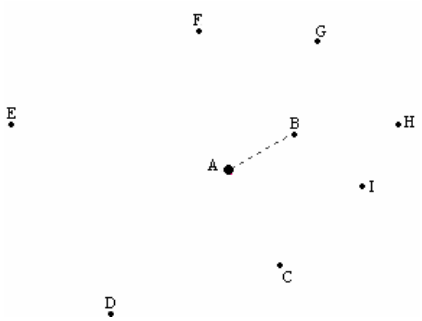


Fig. 2 Action set in 1st state is {B, C, D, E, F}, action set in 2nd state is {A, G, H, I, C}.

The total reward for each agent is computed by (3). When the total reward is got, it reinforces selected actions which are stored in the episodic memory of each agent just whenever the got total reward be the greatest value since now, according to a credit assignment function f(R, t) (4) as value, in which is satisfied rationality theorem. In this way the first best arrangement would be attained.

$$totalreward = (R - P) \times \delta^{n} \qquad (3)$$

Where R is accumulated performance rating, P is total penalties, n is the number of created teams and $0.5<\delta<1$.

$$f(R,t) = totalreward \times \gamma^{T-t} \qquad (4)$$

Where T: time at goal, t=0: time at initial state and $0<\gamma<1$ which can be decided by 1/(L-1) (L: the average number of available action in each state).

IV. Experimental results

We use function $f(R, t) = [(R - P) \times \delta^{n}](0.4)^{T-t}$ to assign a reward as value to each state-action pair of the episodic memory. The δ value is 0.7 (note that the lower δ causes more exploration). In each experimental condition, agents learn 2500 episodes as a trial; and iterate 10 trials to evaluate the average, where the lookup table for each agent is reset after each trial.

Fig. 3 shows the learning curve of the required episodes to optimize the arrangement of 15 agents in 5 teams, according to the conditions. The X-axis indicates the number of episodes and the Y-axis indicates the average performance rating of arrangements in 10 trials.

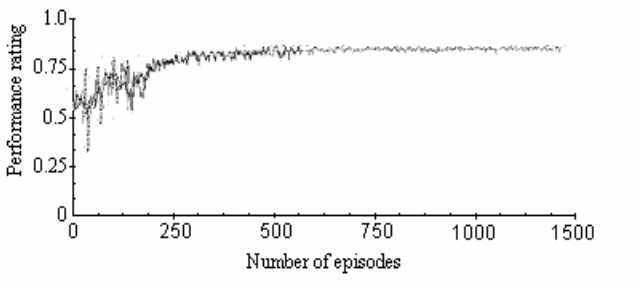


Fig. 3 Number of episodes_Performance rating.

The most popular investigated triangulation is so-called Greedy (Shewchuk) [19] method. It is based on the sweep-line paradigm combined with the recursive local optimization procedure. The sweep-line is one of the most popular acceleration techniques used to solve 2D geometric problems [15]. The idea of the sweep-line technique is very simple: it is imagined that the sweep-line glides over the plane and stops at the end of plane. A portion of the problem is solved and the data structures are updated. The part of the problem being swept is already completely solved, while the remaining part is unsolved. Unfortunately, this simple and clear principle cannot be applied directly for constructing team participation (in this case). Namely, at the end of the survey points (agents) which could participate in no team aspect to their distance. In this case, the arrangement is reconstructed from the end to dawn such that causes minimum changes in team participation. Only fortune found an optimal solution [5]. This algorithm maintains information about those parts of the diagram already swept and never has an overall consideration. Proposed method in this paper, and greedy approach were applied for testing. Greedy algorithm is often used as the reference for evaluation of triangulation algorithms [1]. The results of this evaluation are shown in Fig. 4.

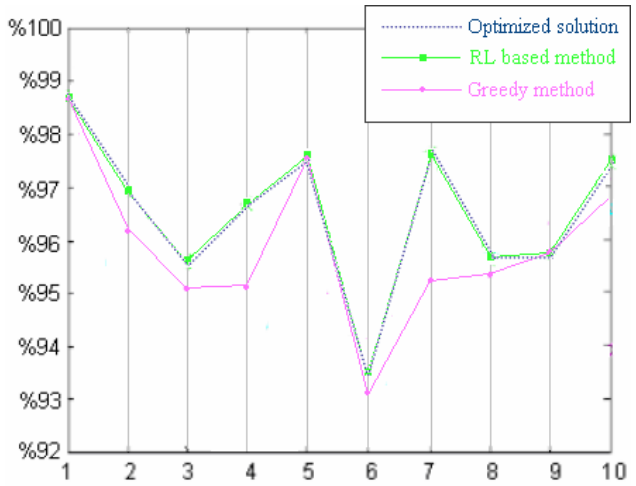


Fig. 4 RL based method in comparison with Greedy method.

V. Conclusion

In this paper we addressed a multi agent system including agents using a Reinforcement Learning algorithm to adapt their behavior to the desired global task. To avoid problems such as combinatorial explosion we used a decentralized approach having agents with local perception. But this learning problem is generally unsolvable because of its decentralized aspect and classical limitations like partial observability of state and non-stationery world. To deal with Partial observability and non-stationery problems, we have used profit sharing learning scheme which is more efficient than other reinforcement learning methods such as Q-learning or Temporal Difference since it is significantly easier to implement and more applicable to real world problems.

The C++ programs are implemented where agents have to coordinate to optimize the performance of the teams. The experiments showed the large efficiency of reinforcement learning to solve the combinational optimization problem, according to mutual information restriction.

References

[1] Boissonnat JD, Devillers O, Teillaud M, Yvinec M. Triangulations in CGAL. In: Computational Geometry 2000. Hong Kong: ACM Press; 11–8, 2000.
[2] C. Watkins; and P. Dayan, Technical note: Q-learning. Machine learning, 8, 1992.
[3] C.Boutilier, Planning, Learning and Coordination in Multiagent Decision Processes, Proc. the 6th Conference on Theoretical Aspects of Rationality and Knowledge (TARK '96), De Zeeuwse Stromen, The Netherlands, 1996.
[4] D. Dutech, Solving POMDP Using Selected Past-Events, Proc the 14th European Conference on Artificial Intelligence, ECAI2000, 2000.
[5] Fortune S. A sweep-line algorithm for Voronoi diagrams. Algorithmica; 2:153–74, 1987.
[6] G. B. Dantzig, Linear programming and extension, Princeton University press, Princeton, NJ, 1963.
[7] J. Ferber, Multi-agent systems: an introduction to distributed artificial intelligence, John Wiley & Sons Inc., New York, 1999.
[8] J. Grefenstette, Credit assignment in rule discovery systems based on genetic algorithms. Machine learning 3:225-245, 1988.
[9] K. L. Clarkson, A Lasvegas algorithm for linear programming when the dimension is small. 29th Annu.IEEE sympos. Found. Comput. Sci., 1988.
[10] K. Miyazaki; M. Yamamura; and S. Kobayashi, On the rationality of profit sharing in reinforcement learning. In proceedings of the 3rd international conference on fuzzy logic, Neural nets and soft computing, 1994.
[11] M. De Berg, M. Van Kreveld, M. Overmars, Computational geometry Algorithms and applications, 9, 2000.
[12] M.Littman, A.Cassandra , L.Kaelbling, Learning Policies for Partially Observable Environments: Scaling Up, Proc. the 12th Int. Conference on Machine Learning, 1995.
[13] O.Buffet, A.Dutech , F.Charpillet, Multiagent Systems by Incremental Reinforcement Learning, LORIA, BP, 2002.
[14] P. Stone, M.Veloso, Multiagent systems: a survey from a machine learning perspective. Autonomous Robotics, 8(3), 2000.
[15] Preparata FP, Shamos MI. Computational geometry: an introduction. Berlin: Springer; 1985.
[16] R. Sutton, Learning to predict by the methods of temporal differences, Machine learning, 3, 1988.
[17] Rosenfeld R., A Maximum Entropy Approach to Adaptive statistical Language Modelling. In Computer, Speech and Language, 10:187-288, 1996.
[18] S.Singh, T.Jaakkola , M.Jordan , Learning Without State Estimation in Partially Observable Markovian Decision Processes, Proc. The Eleventh International Conference on Machine Learning, 1994.
[19] Shewchuk JR. A two-dimensional quality mesh generator, Delaunay triangulator. http://www.cs.berkeley.edu/jrs/index.html, 2004.
[20] R.S.Sutton,A.G. Barto, Reinforcement learning Bradford Book, MIT Press, Cambridge, MA, 1998.
[21] Y.Shoham, R.Powers , T.Grenager, Multi-agent Reinforcement Learning: a critical survey. Computer Science Department, Stanford, CA, 2003.

ISAST Transactions on Intelligent Systems, No. 1, Vol. 1, 2008
M. Bastanfard and S. D. Katebi: Planning and scheduling by reinforcement learning

Planning and scheduling by reinforcement learning

M. Bastanfard and S. D. katebi

***Abstract*—** The objective is to provide operational and engineering recommendations system for planning and scheduling, in order to achieve performance improvements, during steady-state and fault free conditions, given a set of exploitation constraints. Because of inherent complexity of these kinds of problems machine learning, reinforcement learning notably, could be so helpful to attain a solution for this aim, according to the conditions. Thus, the expected result is a system, reducing the time necessary to obtain the desired result.

***Keywords*—** Planning, scheduling, machine learning, reinforcement learning.

I. Introduction

Reinforcement learning methods are very useful ways for learning optimal memory-less behaviors [1]. Recently there has been growing interest in extending RL to the multi agent domain where autonomous agents interact with each other and the environment [2][3].

Learning in Multi Agent Systems (MAS) can indeed take many forms [4]. In many cases it is very promising to use RL as RL algorithms don't require a teacher who knows the best decisions, it just needed feedback of environment as a scalar value for example, that forms a reward or punishment to actions has been done by the agents.

Whereas RL has a optimizing inheritance, it can be used in planning and scheduling problems. Planning is one of complex open problems in which one more condition should be satisfied simultaneously. Besides, these kinds of problems might be involved by a set of exploitation constraints and restrictions. In classical methods for solving the planning problems, such as integer programming [5][6], the little changes in the conditions, could cause a large amount of necessary and inescapable changes in the model and the solution. In addition, when the dimension increases the solution becomes unreliable even sometimes becomes impossible. Admiringly in reinforcement learning, just set the reward function is sufficient and no change in other parts of learning algorithm is need.

Manuscript received April 1, 2007. Planning and scheduling using reinforcement learning.

M. Bastanfard, and S.D. Katebi are with Computer Science and Engineering department of Shiraz University, Shiraz, Iran (fax: +98-711-6271747; e-mail: bastanfard@cse.shirazu.ac.ir, katebi@shirazu.ac.ir).

As RL suffers from combinatorial explosion when there are a large number of states in the system, using decentralization is beneficial in this way, where we have simple agents with local perception, each learns its own behavior by itself, state space of each agent is not too large and we can avoid this computational bottleneck. However, by using this approach we encounter some problems:

- Partial observability*:* Agents can only rely on their own imperfect, local and partial perception of their world.
- Non-stationary world: Reactive agents with a local view of the environment can not use joint actions to solve the problem.

To deal with these problems, in this paper, reinforcement learning, a machine learning method for the demonstration of research results in the field of planning and scheduling is described.

We considered Multi-Agent Systems composed of simple reactive situated agents with local perceptions those try to solve the problem, which can be the case in real problems as most often global knowledge is not available for each individual agent. RL techniques are used for finding stochastic policies. Actually, agents with stochastic behaviors are more adapted to the problem of partial perception.

II. Reinforcement Learning

Reinforcement learning(RL) methods are very useful ways for learning optimal memory-less behaviors for agents as they only require a scalar feedback from the system to the agents for them to learn. Besides, these techniques can be used when there is uncertainty in the world's evolution. But the convergence of RL algorithms (such as Q-Learning and TD) has only been proven for Markov Decision Processes (MDP). An MDP can be viewed as a stochastic automaton in which an agent's actions influence the transitions between states, and rewards are obtained depending on the states visited by an agent. In a fully observable MDP, it is assumed that the agent can determine its current state with certainty. Formally, a fully observable MDP —that is, the agent knows the true state at each time (once that stage is reached) — can be defined as a tuple <S,*A,T,R*>, where *S* is a finite set of states or possible worlds, *A* is a finite set of actions, *T* is a state transition function, and *R* is a reward function. The agent can control the state of the system to some extent by choosing actions from the set *A* that influence state transitions: movement from the current state to some new state. Actions are stochastic in that the actual transition cannot generally be predicted with certainty. The transition function *T* describes the effects of each action at each state. *T(s,a,s')* is the probability of ending

up in state s' when action a is performed at state s. After each transition, the environment generates a reward given by $R(s,a)$. The problem is to find the optimal mapping $\pi(s,a)$ between states and actions so as to maximize the reward received over time. Such a mapping is called a policy.
When the system is fully observable its choice of action can be based solely on the current state. Thus uncertainty lies only in the prediction of an action's effects, not in determining its actual effect after its execution. As pointed out by [7] the evolution of the kind of MAS we are interested in is a MDP. As is noted before we face two major difficulties:
• Non-stationary world: Reactive agents with a local view of the environment can not use joint actions to solve the problem. In fact, other agents are unpredictable elements of the environment while they are learning too. Moreover transitions from one state of the system to another, as seen by an agent, are non-stationary. For example the probability for an agent to corporate in a team depends greatly on the decisions of other adjacent agents.
• Partial observability: Agents can only rely on their own imperfect, local and partial perception of their world. The global state of the system is hidden, which prevents classical RL algorithms from finding a globally optimal policy [8]. As such, the problem at hand belongs to the class of partially observed Markov decision models [9].

Classical stationary Partially Observed Markov Decision Processes are nearly impossible to solve when there are more than a hundred of states [10]. The combination of the two problems (i.e non-stationarity and partial observation) makes the problem non-solvable without using approximations. In the context of partially observable MDP, stochastic behaviors perform better than deterministic policies [8].

III. PROBLEM DESCRIPTION

The interactions of agents and their behaviors are considered in a game frame work for ignoring the difficulties of the real word. The task chosen involves agents in a 30×30 grid world whose goal is to save the victims, where there exist multiple victims and multiple rescuers. Each rescuer is assumed to be a learning agent, whereas the victims does not learn and distributes randomly in the environment. Three agents as a team are needed for saving one of the victims.
Our agents are assumed to participate in teams, by selecting two other agents between their closest neighbors all around and now each team has to visit and save the victims. A rescuer can know the location of a victim only when the victim is in the rescuer's sight. The sight of rescuer is decomposed into 4 different directions {left, front, right and back} (Figure 1). It makes the consequence of agent's actions stochastic depending on global state of the world and the sequence in which agents have been considered to do their actions.

The final goal of the agents is to visit and save all the victims in the environment. Under these conditions, the rescuers need not only to find the path to the victim but also to decide each target victim which should be common to the rescuers. As far as finding a path to the victim, the rescuers must come close to the target victim. On the other hand, deciding which victim to target for saves requires additional cooperation to form consensus on the sequence of saving the victims. Therefore, we need to take the agents' concurrent learning [15] [16] into consideration.

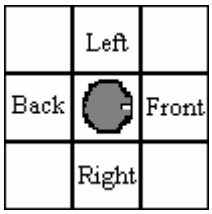


Figure 1: Action set of an agent.

A sequence of visited victims by a team's member with the best performance is desirable. As well as Attending and responding the closest request is optimal policy in a client server system [5], performance test for a sequence is that be sorted in a ascending form of overall distance to all the team's members.

A. Agents

We used very simple agents for complexity reasons. It allows us to concentrate on the learning aspect of the problem. Our agents share the same goal and they have to coordinate to reach the common goal. Agents don't have any explicit communication with each other and they have their own imperfect and vague local perception of the environment.
Each agent has its own perception of the world, agent perceptions are:
• Direction of victims which are located in area bound up to the closest neighbors of itself (Front, Left, Right, Back).
• Direction of other agents (Front, Left, Right, Back).
• Distance from closest agents all around.
• Is there any victim?

B. Learning Algorithm

Agents have to visit and save all the victims which is the final state for the system. An agent can know a victim only when the victim is in the agent's sight.

Each agent with individual perceptions first, selects one of possible actions from the set which consists {Left, Front, Right and Back}. The action is selected by the roulette selection, soft greedy method [2], in which the selection rate of the action is in proportion to its current value. This selection method makes agent behave under the stochastic policy and explore its strategy. Each action's value is initialized by zero.

$$\pi_\theta(s,a) = \frac{e^{\theta(s,a)}}{\sum_{b \in A} e^{\theta(s,b)}} \quad (1)$$

Next, the action selection is repeated till the agent recognizes a victim.
The agent is rewarded when visit a victim and one step before visiting a victim (Figure 2). Rewards are accumulated in a variable such as R. In the global consideration, the visiting

sequence is rewarded by R and penalized by -1 for each no common victim up to two. Additional reward is given to the agents when all the victims are visited.

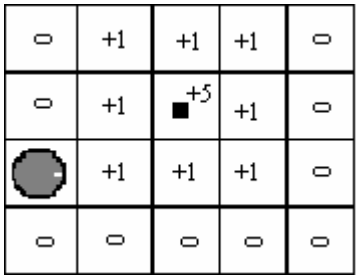


Figure 2: Rewards / states.

The total reward for each agent is computed by (2). When the total reward is got, it reinforces selected actions which are stored in the episodic memory of each agent according to a credit assignment function f(R, t) (3) as value, in which is satisfied rationality theorem.

$$\text{total reward} = (\text{R - P}) / (\delta_a^{na}\, \delta_b^{nb}\, \delta_c^{nc}) \qquad (2)$$

Where R is accumulated rewards, P is total penalties, n_i is the number of steps of i^{th} agent for visiting the victim and $0.5<\delta_i<1$ where can be differed for each agent.

This reward function implies the standpoint of action_orientation concept learning [17], in which the path to the target is rewarded as well as the actions which are selected.

$$f(R,t) = \text{total reward} \times \gamma^{T-t} \qquad (3)$$

Where T: time at goal, t=0: time at initial state and $0<\gamma<1$ which can be decided by 1/(L-1) (L: the average number of available action in each state).

IV. Experimental results

We use function $f(R,\ t) = (R\ -\ P)\ /(\delta_a^{n_a}\delta_b^{n_b}\delta_c^{n_c}) \times (0.4)^{T-t}$ to assign a reward as value to each state-action pair of the episodic memory. The δ value is performed here is 0.6 for all the agents (note that the lower δ causes more exploration). In each experimental condition, agents learn 5000 episodes as a trial; the lookup table of each agent is reset after each trial, and iterates 10 trials to evaluate the average.

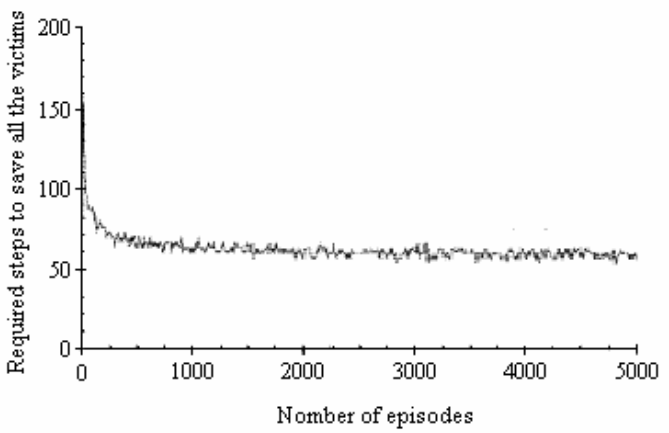


Figure 3: Number of episodes_ Required steps to visit and save all the victims.

Figure 3 shows the learning curves of the required episodes to plan and order of victims visiting according to the conditions. The X-axis indicates the number of episodes and the Y-axis indicates the average of required steps in 10 trials.

To evaluate performance of the rescuers without global knowledge, we compared with the baseline condition in which a single global agent schedules the ordering of victim visiting. In this case, the global agent is given the information about the location of all the victims and rescuers, then selects a target victim by $\arg\min_{j\in victims} \sum_i DISTANCE(victims_j, agent_i)$. Then, all agents focus on the target victim, which the global agent decided to visit and save, and ignore the other victims. In this case, an agent neglects the other victims although they could be in its sight. After visiting the first victim, the global agent decides the next target and agents repeat the same procedure until visiting whole victims.

Also we compared our proposed learning algorithm with and without global agent. In this case, no agent knows the global information about the environment and the other agents. So, all agents learn and adapt their behaviors individually based on own imperfect perception of the world. Each agent find a victim, focus on it and try to visit it. In this case, agents might try to save no common victim.

Figure 4 shows the learning curves of the required steps to visit and save the 5 victims (in average). The X-axis indicates the number of episodes and the Y-axis indicates the average of required steps in 10 trials. The case with global agent conditions shows more effective performance than without global agent to visit whole victims because the agents' target is always consistent among them. However, what we notice here is that the required steps to save the whole victims by the case of without global agent algorithm is larger than that in proposed learning algorithm.

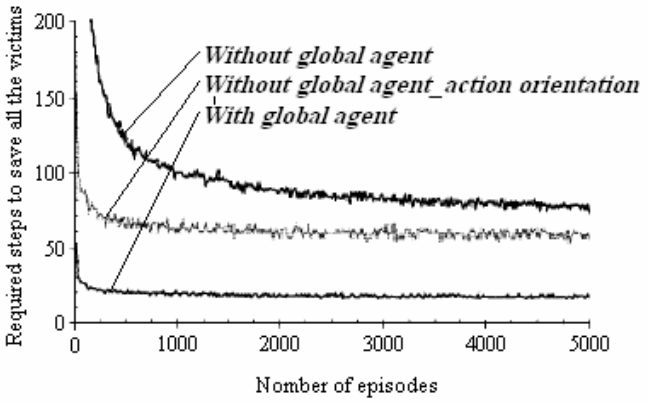


Figure 4: Comparison of with global agent, without global agent and proposed algorithm.

This fact shows that agents in without global agent based on action_orientation concept seem to be thrown into a kind of perception and to be compelled them to move unnaturally way because they are concealed non target victim from their sights.

V. Conclusion

In this paper we addressed a multi agent system including agents using a Reinforcement Learning algorithm to adapt their behavior to the desired global task. To deal with Partial observability and non-stationery problems, we have used profit sharing learning scheme and a reward function based on action_orientation concept learning pointview, which is more efficient than other reinforcement learning methods such as Q-learning or Temporal Difference since it is significantly easier to implement and more applicable to real world problems.

The C++ programs are implemented where agents have to coordinate to plan and schedule order of visiting victims. The experiments showed the large efficiency of action_orientation concept reinforcement learning in decentralized aspect.

References

[1] Y.Shoham, R.Powers , T.Grenager, Multi-agent Reinforcement Learning: a critical survey. Computer Science Department, Stanford, CA, 2003.

[2] R.S.Sutton,A.G. Barto, Reinforcement learning, Bradford Book, MIT Press, Cambridge, MA, 1998.

[3] J. Ferber, Multi-agent systems: an introduction to distributed artificial intelligence, John Wiley & Sons Inc., New York, 1999.

[4] P. Stone, M.Veloso, Multiagent systems: a survey from a machine learning perspective. Autonomous Robotics, 8(3), 2000.

[5] K. L. Clarkson, A Lasvegas algorithm for linear programming when the dimension is small. 29th Annu.IEEE sympos. Found. Comput. Sci., 1988.

[6] G. B. Dantzig, Linear programming and extension, Princeton University press, Princeton, NJ, 1963.

[7] C.Boutilier, Planning, Learning and Coordination in Multiagent Decision Processes, Proc. the 6th Conference on Theoretical Aspects of Rationality and Knowledge (TARK '96), De Zeeuwse Stromen, The Netherlands, 1996.

[8] S.Singh, T.Jaakkola , M.Jordan , Learning Without State Estimation in Partially Observable Markovian Decision Processes, Proc. The Eleventh International Conference on Machine Learning, 1994.

[9] M.Littman, A.Cassandra , L.Kaelbling, Learning Policies for Partially Observable Environments: Scaling Up, Proc. the 12th Int. Conference on Machine Learning, 1995.

[10] D. Dutech, Solving POMDP Using Selected Past-Events, Proc the 14th European Conference on Artificial Intelligence, ECAI2000, 2000.

[11] J. Grefenstette, Credit assignment in rule discovery systems based on genetic algorithms. Machine learning 3:225-245, 1988.

[12] K. Miyazaki; M. Yamamura; and S. Kobayashi, On the rationality of profit sharing in reinforcement learning. In proceedings of the 3rd international conference on fuzzy logic, Neural nets and soft computing, 1994.

[13] C. Watkins; and P. Dayan, Technical note: Q-learning. Machine learning, 8, 1992.

[14] R. Sutton, Learning to predict by the methods of temporal differences, Machine learning, 3, 1988.

[15] S. Arai; K. Miazaki, and S. Kobayashi, Generating cooperative behavior by multi agent reinforcement learning. In proceedings of 6th European workshop on learning robots, 111-120, 1997.

[16] S. Sen, and M. Sekaran, multi agent cooperationwith learning classifier systems. In Weiss, G. and Sen, S., eds., aadaption and learning in multi agent systems. Berlin, Heidelberg: springer Verlog, 218-233, 1995.

[17] R. Sutton, D. Precup, and S. Singh, Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement Learning, artificial intelligence, 112, pp.181-211, 1999.

Feature selection using reinforcement learning

M. Bastanfard and S. D. katebi

***Abstract*—** In this paper, a method for feature selection by reinforcement learning is considered. Reinforcement learning is the problem faced by an agent that learns behavior through trial and error interactions with a dynamic environment. The work described here is a multi agent system in which the agents which try to coordinate to reach common goals, would be learned by considering the ability of the features to separate the sample data in order to discriminate them as well as possible. This discrimination is under Linear Discriminate Analysis; in addition the correlation coefficient does as the weight, perspective to maximize the minimum distance of categories.

***Keywords*—** Feature, agent, reinforcement learning, Linear Discriminate Analysis, correlation coefficient.

I. Introduction

A classification model acts as a function, f, mapping from features, F, to classes, C. But in many cases we don't know the relative importance of the different features, not how they interact. It is very hard to judge the utility of the information provided by a feature. We may run into the problem highlighted by Rosenfeld [1] that we will never have enough data to model rare events – because they are rare. Thus the first task of each classifier system will be to decide which of the very many possible features will be of use in practice.

A feature set with the highest ability for separating the classes as much as possible is needed. Besides, more features mean a larger database/classification model and slower training and classification where is related to the curse of dimensionality, which refers to the fact that the number of data samples required to estimate some arbitrary multivariate probability distribution increases exponentially as the number of dimensions in the data increases linearly.

Simple feature selection algorithms are ad hoc, but there are also more methodical approaches. From a theoretical perspective, it can be shown that optimal feature selection for supervised learning problems requires an exhaustive search of all possible subsets of features of the chosen cardinality. If large numbers of features are available, this is impractical. For practical supervised learning algorithms, the search is for a satisfactory set of features instead of an optimal set. Many popular approaches are greedy hill climbing approaches. Such an approach evaluates a possible subset of features and then modifies that subset to see if an improved subset can be found.

Evaluation of subsets can be done many ways - some metric is used to score the features, and possibly the combination of features.Two popular metrics for classification problems are correlation [7] and mutual information [1]. These metrics are computed between a candidate feature (or set of features) and the desired output category.

In statistics the most popular form of feature selection is called stepwise regression. It is a greedy algorithm that adds the best feature (or deletes the worst feature) at each round [3].In such algorithms, the chance of the features to be selected in each round depends on features which are selected since now. So the algorithm is so related to the order of selections. Besides, forgetting the features which are selected since now is not possible because of lack of back tracking in such algorithms. So the selected feature set might be just a local optimum.

In this paper we are looking towards a 'third way'. Since it is relatively easy to suggest features, but hard to know how useful they are, we leave this task up to the learning system. We simply suggest a large number of possible features and let the system decide which ones to take note of. We considered Multi-Agent Systems composed of simple reactive situated agents with local perceptions in a decentralized [10] way, those try to solve the problem for finding a proper feature set in a decentralized way. Thus the agents adjust and accord its own behaviors based on reinforcement learning and coordinate to each other to learn from the environment.

Here, we describe how we have implemented and tested this application of machine learning within the multi agent system. In section 2 we describe the mathematical framework we have adopted, section 3 is a glimpse at reinforcement learning and in section 4 the learning algorithm is described, while in section 5 a small experiment proposed only as a proof-of concept is considered. In section 6 the results and comparison is reflected.

II. Feature Space

What features of an object would assist in determining its probability of being a correct member of a category? Our aim is to reduce the dimension of feature space to the least possible one and make diagnosis in this. Such dimension should fulfill the following requirements: (i) information lost during reduction should be as small as possible, (ii) resulting

Manuscript received April 15, 2007. Planning and scheduling using reinforcement learning.

M. Bastanfard, and S.D. Katebi are with Computer Science and Engineering department of Shiraz University, Shiraz, Iran (fax: +98-711-6271747; e-mail: bastanfard@cse.shirazu.ac.ir, katebi@shirazu.ac.ir).

space should provide sufficient information for certain purpose.

One scheme is to select features that correlate strongest to the classification variable. This has been called maximum-relevance selection [4]. On the other hand, features can be selected to be mutually far away from each other, while they still have "high" correlation to the classification variable. This scheme, termed as minimum-Redundancy-Maximum-Relevance selection [3], has been found to be more powerful than the maximum relevance selection. As a special case, the "correlation" can be replaced by the statistical dependency between variables. Mutual information can be used to quantify the dependency and redundancy.

A. *Linear discriminate analysis*

The reason is that many parameters (features) are likely to be required to characterize it. In addition, when so many features are involved, it is difficult to decide the ones that are most relevant for recognition. Fisher linear discriminant is a method that provides a measure of information about classes represented by features. The principle of the method can be shown in a feature space. There are clusters of points belonging to different classes. Fisher linear discriminant assumes that each cluster can be represented by its mean value and variance (covariance matrix for more than 1_dimensional feature space). The smaller variance inside clusters and higher distances between them, the more appropriate the features are. Therefore, good features are those for which

$$\frac{\text{inter - classes variance}}{\text{intra - classes variance}}$$

is higher than for others. Since the variance inside the individual classes can be different we can use following:

$$\frac{\text{variance between classes}}{\text{higher intra - class variance}}$$

Variance inside classes was not computed as covariance of mean values [15], but as covariance of all feature points, which is more precise in our case.

Generally, suppose we have data in n classes. Each class is represented by matrix Xi where columns are feature vectors. Suppose the following:

$$A = \mathrm{cov}([X_1 X_2 X_3 ... X_n]) \quad (1)$$

$$B_i = \mathrm{cov}(X_i) \quad (2)$$

$$\lambda_i = \max(eig(B_i^{-1} A)) \quad (3)$$

For deeper insight into values that can be assumed by Fisher linear discriminant, we will do the following. Suppose

$$p = K_i / \sum_{j=1}^{n} K_j$$

where K_i is a number of vectors in matrix X_i , $\sum_{j=1}^{n} K_j$ is a number of vectors in $[X_1\ X_2\ X_3\ ...X_n]$.

Values of parameter p can be from interval < 0, 1 >. It was found experimentally that:

1. In case of perfect overlap of classes (they have equal mean values and variance of one class is zero), then F= p.
2. For identical classes (equal mean values and variances (covariance matrices)), then F = 1.
3. If classes are perfectly separable, then F = 1.
4. In case of only partially overlap, then p< F <1.
5. If classes are merely separable, then -1< F <1.The higher F, the better separability.

B. *Correlation coefficient*

Another way to compare features is multi-correlation coefficient R(i).

$$R(i) = \mathrm{cov}(x_i, y) / \sqrt{\mathrm{var}(x_i)\,\mathrm{var}(y)} \quad (4)$$

Where cov(x,y) is the covariance matrix of two dimensions x , y and var shows the variance.

The coefficient R(i) is also the cosine between vectors xi and y, after they have been centered (their mean subtracted). Although the R(i) is derived from R(i) it may be used without assuming that the input values are realizations of a random variable. In linear regression, the coefficient of determination, which is the square of R(i), represents the fraction of the total variance around the mean value that is explained by the linear relation between xi and y. Therefore, using $R(i)^2$ as a variable ranking criterion enforces a ranking according to goodness of linear fit of individual variables. The use of $R(i)^2$ can be extended to the case of two-class classification, for which each class label is mapped to a given value of y.

Correlation criteria such as R(i) can only detect linear dependencies between variable and target. A simple way of lifting this restriction is to make a non-linear fit of the target with single variables and rank according to the goodness of fit. Because of the risk of over fitting, one can alternatively consider using non-linear preprocessing (e.g., squaring, taking the square root, the log, the inverse, etc.) and then using a simple correlation coefficient.

III. REINFORCEMENT LEARNING

Reinforcement learning is a technique for an autonomous agent that senses and acts in its environment and learns to choose the optimal action by means of a reward value determining how good the action was the agent played. The agent is not programmed to take a certain action in a certain state, instead the agent senses its environment and based on these values and prior experience an action will be chosen. When the action is performed the agent receives feedback from the environment in the form of a numerical reward. High rewards correspond to good actions and low rewards correspond to bad actions. This feedback is an indication for the agent how well it performed by taking that precise action in that precise state. The reward will be taken into account when computing the next action for that same state. This trial-and-error search enables the agent to develop an optimal strategy. The action an agent takes directly influences the environment's state. Thus actions taken early might not only affect the immediate reward but also all subsequent rewards.

Reinforcement learning is different from supervised learning because there the agent is given examples by an external supervisor. From these examples the agent can learn a strategy and whenever it encounters a new case it will be

classified according to this strategy. This kind of learning has two major drawbacks. First the examples provided by the supervisor must be representative for all situations. Second, the domain the agent is learning in must be explored previously the agent can't survive in a completely new environment.

The idea of learning from examples and exploiting the experience afterwards can also be found in reinforcement learning but without a teacher that supervises the learning process. For the agent to obtain high rewards, an agent must prefer actions it has tried before and resulted in a high reward. But to discover these good actions an agent has to explore new, unexplored actions. Finding a good balance between exploration and exploitation turns out to be quite challenge.

As RL suffers from combinatorial explosion when there are a large number of states in the system, using decentralization is beneficial in this way, where we have simple agents with local perception, each learns its own behavior by itself, state space of each agent is not too large and we can avoid this computational bottleneck. However, by using this approach we encounter some problems:

• Partial observability: Agents can only rely on their own imperfect, local and partial perception of their world. The global state of the system is hidden, which prevents classical RL algorithms from finding a globally optimal policy [14]. As such, the problem at hand belongs to the class of partially observed Markov decision models [9].

• Non-stationary world: Reactive agents with a local view of the environment can not use joint actions to solve the problem. In fact, other agents are unpredictable elements of the environment while they are learning too. Moreover transitions from one state of the system to another, as seen by an agent, are non-stationary. To deal with these problems we used RL techniques for finding stochastic policies. Actually, agents with stochastic behaviors are more adapted to the problem of partial perception [2].

IV. LEARNING ALGORITHM

From a theoretical perspective, it can be shown that optimal feature selection [4] requires an exhaustive search of all possible subsets of features of the chosen cardinality. If large numbers of features are available, this is impractical. But by using an insidiously trick for selecting features, aspect to their values in reinforcement learning, the number of really efficient comparisons becomes more practical and can overcome to this difficulty.

All the suggested features would be represented as an environment in which for each three features one agent is added in this environment, uniformly. The agents share the same goal and they have to coordinate to reach the common goal. Our agents don't have any explicit communication with each other and they have their own imperfect and local perception of the environment.

An agent can see just three features with no overlapping other agents' sight. A transition from a state to another can be done by an agent with selecting an action which is catch a feature, or none. This defines a non deterministic finite state automaton, whereas the convergence of reinforcement learning is proven just for finite state automata [11]. In this way the system should be converted to a finite state automaton.

The suggested feature set is divided to subsets with three members randomly. This can be supposed the perception of each agent of the environment. In each subset a dummy member per feature is added to. The i^{th} dummy can be selected if i^{th} feature won't be (F_i is defined as a fraction of feature set F and D_j is a dummy member (Fig 1)).

$$T_i = F_i \bigcup \{D_1, D_2, D_3\}$$

First, when an agent observes current state O_t , it checks its look up table to search the matched state as O_t and gets its action set A_t in a decentralized way, which consists of available features/dummies at time t. The action selection make agent behave under the stochastic policy based on exploration.

Fig 1. Defined actions for an agent.

Although ε-greedy action selection [13] is a popular method for finding equilibrium between exploration and exploitation it has a drawback in the exploration phase. When selecting an action at random the agent gives all actions the same uniformly distributed chance. A consequence is that the worst action is chosen with the same probability of the best action. In environments where bad actions lead to heavy penalties this behavior is unacceptable. A logical solution would be to base the random behavior on the estimated usefulness of the actions.

The action is selected by the roulette selection, soft greedy method [12], in which the selection rate of the action is in proportion to its current value. This selection method makes agent behave under the stochastic policy and explore its strategy. The greedy action is still given the highest probability of getting selected, but the other actions are ranked and weighted according to their estimates. The most common softmax action selection method used a Gibbs or Boltzmann distribution. An action a is selected with probability

$$prob(a) = e^{Qt(a)/T} / \sum_{b=1}^{n} e^{Qt(b)/T} \qquad (5)$$

where T is a temperature factor that determines the amount of exploration. High temperature values will result in a lot of exploration and low values result in the exploitation of experience. Each action's value is initialized by zero.

After agent outputs the selected action a_t it checks if a reward is given or not. If there is no reward after an action a_t, the agent stores the state-action pair (O_t,a_t) into its episodic memory as a rule, and continues the same cycles until

catching the absorbing state. The period from the start to the absorbing state, is called an episode.

When all the agents complete their episodes, share their attainments. The separability of categories would be evaluated by this feature set and features which are stored in the episodic memory of each agent get a reward Ri according to its effect and a credit assignment function f(R, t) which satisfies the "rationality theorem [6]" (6).

$$R_i = \lambda^N_{\max} \times \cos(\alpha_i) \times \gamma^n \times \rho_i \quad (6)$$
$$\lambda^N_{\max} = \lambda_{\max} - \lambda_{mean} / \lambda_{\max} - \lambda_{\min}$$
$$\rho_i = Max_j(\rho_{ij})$$

Where R_i is the reward for selecting the ith feature, λ^N_{max} is the maximum eigen value of covariance matrix of feature set, α_i is the angel of maximum eigen vector with feature i, n is the number of selected features in this episode, ρ_{ij} is correlation coefficient according to ith and jth features in this episode and $0<\gamma<1$.

The gist of this evaluation is that all the selected actions in episodic memory are in the same level and no one has priority over the others. In the other hand, order of selecting the features is not important but the feature set which is attained in this episode and its overall effect on the sample data is the main attendance.

Reward +1 is got by the agent which has selected just the dummies in this episode.

The agent by looking at the effective and ineffective features over large numbers of episodes tries to find features that can separate the categories as much as possible which are the three members in own subset with the highest value function.

V. EXPERIMENTAL RESULTS

In experiment, the learning feature selection system is performed to two different kinds of dataset, symbolic and natural, as materials.

A. Symbolic materials

RL based learning feature selection method is tested on different symbolic datasets such as glass and wine (symbolic datasets) to show the performance of reward function as criterion.

1) Evaluation

For having a comparison between different versions of LDA, we also tested other methods simultaneously. We tested other linear methods such as: Linear Discriminate Analysis and K-Nearest Neighbors (KNN), Linear Discriminate Analysis, Principal Component Analysis [6] and K-Nearest Neighbors classifier.

Results are shown in Table I and II for glass and wine dataset respectively.

TABLE I
COMPARISON OF METHODS FOR GLASS DATASET.

	Prop+KNN	LDA+PCA+KNN	LDA+KNN

TABLE II
COMPARISON OF METHODS FOR WINE DATASET.

	Prop+KNN	LDA+PCA+KNN	LDA+KNN

B. Natural materials

Natural image materials for training and testing are selected from the sonogram images of thyroid gland which are digitized data from ultra sonographical imaging system Toshiba ECCOCEE (console model SSA-340A, transducer model PLF-805ST at frequency 8MHz). Data are captured in longitudinal cross-sections of both lobes with magnification of 4cm and stored with amplitude resolution of 8 bits (256 grey levels). Examples of these images can be seen in Fig 2.

There are two kinds of images: image with normal tissue and image with lymphocyte thyroid. A set of ultrasonic images which is contained 680 sonograms and pathologically proven benign lymphocyte tissue from 350 patients and normal tissue from 330 candidates is considered. The data set contains only one image from each patient. The patients' ages ranged from 28 to 64 years.

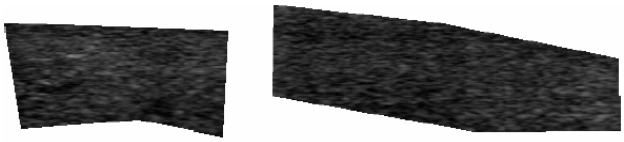


Fig 2. The normal and lymphocyte tissues of thyroid gland.
a | b a) Normal tissue, b) Lymphocyte tissue.

1) Feature generation

Our aim is to classify the thyroid tissue sonogram images in corresponding categories as normal or lymphocyte. In this way, we based our features simply on the extracted features within the images. To do this, a set of 42 features which are listed in Table III are designed. The features were chosen to have approximately exhaustive and miscellaneous information [5][8] about the sonogram image.

TABLE III
EXTRACTED FEATURES FOR INITIAL FEATURE SET.

	Name	Description
1	Mean	Mean
2	Variance	Variance
3	Entropy	Entropy
4 – 13	$Hi_1 – Hi_{10}$	First ten values of Histogram
14 – 23	$F_1 – F_{10}$	First ten coefficients of Fourier
24 – 33	$W_1 – W_{10}$	Wavelet sub bands energy for 3 levels
34 – 42	$H_1 – H_9$	Haralick features (Extracted from co-occurrence matrix d=[11,0])

2) Results

We use the designed features as initial set and perform the RL based proposed algorithm with $\gamma=0.5$ (note that the lower γ causes more exploration) to select the features trough the initial space. In this experiment, agents learn 500 episodes. The result of this proposed method is given comparison with result of a popular greedy method which could be considered as a baseline.

From the list of features which are extracted from the data (Table III), a greedy algorithm such as forward feature selection (following [1]) that considers first the best feature, then the best second feature that can be added to the first, the best third feature that can be added to the first two, and so on. The algorithm halts when the additional benefit of adding another feature falls below some threshold.

TABLE IV
PROPOSED FEATURES USING GREEDY & RL BASED ALGORITHMS.

	Name	Greedy Alg.	RL based Alg.
1	Mean	–	–
2	Variance	–	–
3	Entropy	–	–
4 – 13	$Hi_1 - Hi_{10}$	Hi_1, Hi_2, Hi_3	–
14 – 23	$F_1 - F_{10}$	–	–
24 – 33	$W_1 - W_{10}$	W_1, W_3, W_5–W_9	W_1–W_{10}
34 – 42	$H_1 - H_9$	H_2, H_4, H_9	H_2, H_4

Table IV shows the result of proposed RL based feature selection algorithm as well as result of a greedy algorithm aspect to Linear Discriminate Analysis criterion.

3) Evaluation

To evaluate the selected features, 90% of sonograms are used for training set and the 10% of them reserved for testing. The procedure was then repeated 10 times for each possible division between test and training. The results seem consistent across each rotation of data.

The results of classification of sonograms as normal and lymphocyte using the K-Nearest Neighbors (KNN) classifier are shown in Table V.

TABLE V
RECOGNITION RATE ASPECT TO GREEDY FEATURE SELECTION COMPARISON WITH RL BASED METHOD.

	RL based (Prop)+KNN	Greedy (LDA)+KNN
Normal	96.85%	93.72%
Lymphocyte	95.49%	91.34%

VI. CONCLUSION

In this paper we addressed a multi agent system including agents using a Reinforcement Learning algorithm to adapt their behavior to the desired global task which is finding an optimal feature set with the smallest loss information during reduction and sufficient information for certain purpose. To avoid problems such as combinatorial explosion as it happens in algorithms such as L plus- R mines we used a trick in decentralized approach.

We have proposed a new algorithm based on reinforcement learning according to a powerful criterion, which is more efficient than other methods such as greedy algorithm (Forward or Backward) and significantly more applicable and rapid to execute. This algorithm also makes the consideration of possible permutation of features empirical.

REFERENCES

[1] A. Berger, S. Della Pietra, V. Della Pietra. A maximum entropy approach to natural language processing. In Computational Linguistics, 22, 1996, pp1-36.

[2] D. Dutech, Solving POMDP Using Selected Past-Events, Proc the 14th European Conference on Artificial Intelligence, ECAI2000, 2000.

[3] H.C. Peng, F. Long and Ding, C., "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 8, pp.1226-1238, 2005.

[4] J. Reunanen. Overfitting in making comparisons between variable selection methods. JMLR, 3:1371–1382 (this issue), 2003.

[5] J. Smith and S.-F. Chang, “Transform features for texture classification and discrimination in large image database”, IEEE Intl. Conf. on Image Proc., 1994.

[6] K. Miyazaki; M. Yamamura; and S. Kobayashi, On the rationality of profit sharing in reinforcement learning. In proceedings of the 3rd international conference on fuzzy logic, Neural nets and soft computing, 1994.

[7] K. Torkkola. Feature extraction by non-parametric mutual information maximization. JMLR, 3:1415–1438, 2003.

[8] M. Stricker and M. Orengo, "Similarity of Color Images", Proc. SPIE Storage and Retrieval for Image and Video Databases, 1995.

[9] M. Littman, A. Cassandra , L.Kaelbling, Learning Policies for Partially Observable Environments: Scaling Up, Proc. the 12th Int. Conference on Machine Learning, 1995.

[10] O. Buffet, A. Dutech , F.Charpillet, Multiagent Systems by Incremental Reinforcement Learning, LORIA, BP, 2002.

[11] P.L. Lanzi, S.W. Wilson, "Optimal classifier system performance in non-markov environments", IlliGAL Technical Report 99022, University of Illinois, 1999.

[12] R.S. Sutton,A.G. Barto, Reinforcement learning, Bradford Book, MIT Press, Cambridge, MA, 1998.

[13] S.B. Thrun, "The role of exploration in learning control with neural networks", In D. A. White and D. A. Sofge, editors, Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches, Florence, Kentucky, Van Nostrand Reinhold, 1992.

[14] S. Singh, T. Jaakkola , M. Jordan , Learning Without State Estimation in Partially Observable Markovian Decision Processes, Proc. The Eleventh International Conference on Machine Learning, 1994.

[15] Z. Kotek, I. Br°uha, V. Chalupa, and J. Jel´ınek. Adaptivn´ı a u˘c´ıc´ı se syst´emy. SNTL, 1980.

Instruments and Families Recognition Using neural networks

Hassan Ezzaidi

Abstract— **Recently various features based on the combination of different spectro-temporal parameters were proposed for instruments recognition. In this context, this paper investigated the optimal dimension features that can be extracted from the spectral information. The coding process is obtained by using the neural networks models where the hidden layer compresses the original information. The number of the unit cells corresponds to the optimal dimension of the features vectors. Rather, various architectures of back-propagation and radial basis networks were applied as classifiers and their results were compared. The musical notes used in the test processing were not presented in the previous training session. The discrete Fourier transform vectors extracted from each segment were used as parameters, with the objective to evaluate only the magnitude of the spectral information convoyed to discriminate between musical sounds instrument. The Music Instrument Sample Database (UIOWA) was used for this experiment. The number of instrument was increased from 14 to 19 compared to previous research. With the proposed method, a perfect score is obtained where the recognition is achieved on the basis of the family level (String, Brass, Reed and Flute). However, when the recognition was achieved based on the instrument level, the performance score is decreased. With the spectral information 17 of 19 instruments were identified and all the family was well recognized. The results show that the temporal information is not important for family classification but should be consider for instrument identification.**

Index Terms—**audio, musical instrument, neural network, pattern recognition.**

I. Introduction

SINCE the early 1980s speech signal processing and analysis has been realized and has advanced progressively in various fields of practice. Over the years, many disciplines have emerged and concentrated research in areas such as speech recognition, speaker identification, speaker verification, speaker segmentation, segments clustering, language recognition and so forth.

Recently many studies have been oriented to musical signal analysis and processing in order to respond to the high demand of internet users and to countless multimedia applications. The demand includes audio indexing, automatic transcription, genre classification, singer identification and instrument recognition.

A musical signal can be viewed as random symbol (note) generated by a combination of an output of the hidden sources. When only one source (instrument) is present at any moment of time, we are addressed to the isolated note known as the monophonic recordings. Many previous studies have focused on the monophonic case. Brown et al.[Brown 01] in their studies compared various features in the automatic identification of woodwind musical instruments by using the speaker identification techniques. The examined features included cepstral coefficients, constant-Q coefficients, spectral centroide, autocorrelation coefficients and moments of the time signal. The best accuracies results were identified between 79% and 84%. The higher-order statistics moments were examined by Dubnov et al. [Dubnov 97]. They concluded that these features are adequate to discriminate well between instruments of different families and are not robust to discriminate between instruments within the same family. Krishna and Sreenivas [Krishna 04] proposed Line Spectral Frequencies as features using a Gaussian Mixtures Models (GMM). They reported the best score of 95% at family level and 90% at 14 instrument level compared to MFCC and LPC features. Marques and Moreno [Marques 99] have proposed a GMM model and support vector machines using the FFT based cepstral coefficients, LPC and MFCC features set. A score of 70% was obtained with 9 instruments.

When more than one source can be presented at any moment in time, then the case is qualified as the polyphonic recordings. Research on the polyphonic case has been particularly challenging and complex with limited studies contributed to this field. The following research investigates exclusively a combination of 2 or 3 simultaneous notes. Among other studies, Kashino and Murase [Kashino 99] describe an approach of the sound source identification based on the template adaptation and music stream extraction. A missing features technique in the GMM classifier was introduced by Eggink and Brown [Eggink 03].

In this paper, we proposed the use of different neural networks structures for music instruments recognition. A neural network with magnitude spectral information without any temporal information may be considered to analyze indirectly how the spectral information contributes to

Manuscript received April 17, 2007.

H. Ezzaidi. Author is with the Université du Québec à Chicoutimi, 555 Boulevard de l'université, Chicoutimi, Québec, Canada. G7H2B1CO 80305 USA (corresponding author to provide phone: 418-555-5011 ext 56-42; fax: 418-545-5012; e-mail: hezzaidi@ uqac.ca).

characterize specifically the musical instrument. The hidden layer with a number of unit cells fewer than the input layer has also been used to explore the coding capacity and the dimension reduction of the input vector features. Comparatively to other experiments (studies) the database was augmented to 19 instruments originating from four families. All experiments in this study are focused on the monophonic recordings.

A. Models

Neural network is a formal model inspired from the organization of biological cells to learn many tasks in a diver's field. Neural networks was applied with success in many application as pattern recognition, signal control and time series analysis. The main propriety is that we can learn only from input data (unsupervised learning) or from input-output data (supervised learning). In the following research, only supervised work is addressed.

B. Backpropagation neural network

A Multilayer Perceptron (MLP) neural network with back-propagation was used as classifier (see Fig. 1). The structure is composed from an input layer where the training data vector is injected. The hidden layer is supposed to realize a compression when the number of cells is lower than the input layer cells and an extension when the number of cells is greater than the input layer cells. The output is an encoding process that characterizes which categories are to be recognized. The network is an acyclique form fully connected. In the following study, the number of cells in hidden layer was taken from following numbers: 20, 40, 80 and 120.

The fastest and steepest descent with the momentum algorithm was used in the training process to update the networks weight and biases in the negative direction of the gradient.

The learning occurs according to the following parameters: the learning rate is fixed to 0.1, and the momentum constant is fixed to 0.9. The multilayer network uses the sigmoid transfer function that generates outputs from 0 to 1 at each unit cell.

Effectively, the input goes between negative to positive infinity. In the case of the classification tasks, the output of the sigmoid function can be interpreted as the approximation of posteriors probabilities of classes.

C. Radial basis networks

The Radial Basis Functions (RBF) network is composed of two layers: the hidden radial basis layer and the output competitive layer. The first layer computes distance similarities between the input and each of the prototype vectors representing the classes. This can be interpreted as a projection over some mean or centre mass. The second layer based on the activity of the first layer produces a probability vector as output. The transfer function for the radial basis neuron is: exp(-d2), where d is the similarity distance. All experiments are realized with the neural network toolbox of The MathWorks, Inc.[Matlab 03].

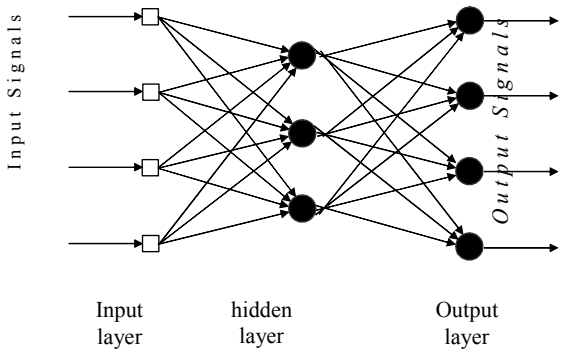


Figure 1: Multilayer Perceptron Neural Network Structure

II. METHODOLOGY

A. Parameters estimation

The energy with a dynamic threshold is done as criterion to detect the start and the end of each note. Indeed the efficiency value of the threshold was verified and adjusted manually to minimize the error detection. The time duration for each note was divided on adjacent window off 1024 points to calculate the magnitude of the Fast Fourier Transform (FFT) vector. By symmetry, only 512 points were kept for the following process. Hence the input of all proposed networks is set to 512 cells. The number of the output cells was also set to 5 where only 19 combinations were used to characterize each instrument. The rest of the other combinations were considered as reject decisions. Rather to coding the resonance of the system (vocal tract), with the magnitude of the FFT all the information related to the excitation source and vocal tract remain here. However the dimension is greater but inside our objective is to explore the compression (dimension reduction) via the hidden layer without evaluating the mapping function.

B. Database

The collection of instruments used in this work was from the database of the [1]University of Iowa, Musical Instrument samples. The collection was composed of 19 instruments coming from four families (String, Brass, Reeds, and Flutes) as follows:

Violin, Sop Sax, Tenor Trombone, Oboe, French Horn, Flute, EbClarinet, Cello, BbClarinet, Bass Trombone, Bassoon, Bass Flute, Alto Sax, Alto Flute, BassClarinet, Bass, Trumpet, Tuba and Viola.

A set of 248 isolated notes for all theses instruments were used to train (first half of data) and to test (rest of data) the proposed systems.

III. RESULTS AND DISCUSSION

The following tables presented in this section illustrate the diagonal score recognition of the confusion matrix. The SR

[1] University of Iowa's Music Instrument Samples, http://theremin.music.uiowa.edu/MIS.html

(19) line corresponds to the score performance with 19 instruments. Precisely, it is the sum of the score obtained respectively by the MAR and MIR criterions. The SR (14) line corresponds to the score performance with 14 instruments added to the score obtained respectively by the MAR and MIR criterions. This last score is proposed mainly to compare the proposed system (with 19 instruments) to previous studies (with 14 instruments) that used different methods. The SR line also reports the score recognition for families as combined the MAR and the MIR scores.

TABLE I
Score Recognition for Instruments Identification Using the Mean of Features and the MLP Networks: SR(19) Is the Score for 19 Instruments and SR(14) Is the Score Recognition for 14 Instruments Using the Criterion MIR and MAR

	Number of unit cells for MLP		
instruments	20	40	60
Violin	20%	40%	40%
Sop Sax	15%	33%	17%
Tenor Trombone	31%	12%	14%
Oboe	19%	33%	30%
French Horn	13%	15%	21%
Flute	10%	6%	8%
EbClarinet	23%	14%	16%
Cello	29%	30%	23%
BbClarinet	67%	66%	66%
Bass Trombone	17%	22%	20%
Bassoon	32%	43%	32%
Bass Flute	14%	23%	17%
Alto Sax	10%	10%	15%
Alto Flute	38%	19%	43%
BassClarinet	7%	9%	10%
Bass	98%	98%	97%
Trumpet	24%	15%	11%
Tuba	34%	42%	42%
Viola	86%	90%	89%
MIR	**5/19**	**5/19**	**5/19**
MAR	**3/19**	**3/19**	**3/19**
SR(19)	**8/19**	**8/19**	**8/19**
SR(14)	**5/14**	**6/14**	**6/14**

TABLE II
Score Recognition for Instruments Identification with MLP Networks: SR(19) is the Score for 19 Instruments and SR(14) is the Score Recognition for 14 Instruments Using the Criterion MIR and MAR

	Number of unit cells for MLP			
	20	60	120	240
String	69%	75%	73%	69%
Brass	40%	45%	40%	40%
Reeds	45%	46%	47%	45%
Flutes	32%	29%	28%	32%
MIR	**3/4**	**3/4**	**3/4**	**3/4**
MAR	**1/4**	**1/4**	**1/4**	**1/4**
SR	**4/4**	**4/4**	**4/4**	**4/4**

TABLE III
Score Recognition for Instruments Identification Using the Mean of Features and the MLP Networks: SR (19) Is the Score for 19 Instruments and SR (14) Is the Score Recognition for 14 Instruments Using the Criterion MIR and MAR

	Number of unit cells for MLP			
instruments	20	60	120	240
Violin	32%	72%	71%	70%
Sop Sax	29%	31%	36%	41%
Tenor Trombone	62%	56%	62%	57%
Oboe	32%	29%	37%	34%
French Horn	20%	27%	23%	19%
Flute	18%	19%	20%	22%
EbClarinet	20%	27%	31%	29%
Cello	62%	65%	68%	71%
BbClarinet	66%	68%	74%	73%
Bass Trombone	19%	23%	23%	23%
Bassoon	49%	49%	51%	49%
Bass Flute	26%	33%	36%	39%
Alto Sax	32%	43%	46%	47%
Alto Flute	32%	37%	38%	39%
BassClarinet	12%	14%	24%	24%
Bass	74%	82%	78%	79%
Trumpet	50%	44%	52%	52%
Tuba	59%	93%	95%	96%
Viola	79%	83%	85%	85%
MIR	**8/19**	**9/19**	**7/19**	**8/19**
MAR	**7/19**	**7/19**	**9/19**	**8/19**
SR(19)	**15/19**	**16/19**	**16/19**	**16/19**
SR(14)	**11/14**	**12/14**	**12/14**	**12/14**

A. Single vector parameter by each notes

In this experiment and under the assumption that the spectral properties remain unchanged, one codes each note by only one vector feature which is the average of the FFT vectors estimated from all frames. The interest of this procedure is the reduction of the computing time. However, the results show that this strategy can be applicable for classification by family and unfortunately none interesting for the instruments classification.

Tables I and II illustrate the score recognition with this strategy. With the efficient MAR criterion only 3 instruments over 19 are recognized which is very lower. This means that the mean of the FFT is not appropriate to discriminate well the

instrument music individually. However, with families' instruments a good score is obtained so this technique can be used for this purpose.

B. Multiple vectors parameters by each notes

In this experiment the parameters are estimated as illustrated in the (subsection II.A) .

Table III illustrates the score recognition of the instruments for various architectures of multilayer perceptron neural networks. With 19 instruments, the best performance was obtained with the architecture containing respectively 60, 120 and 240 unit cells in the hidden layer. With 120 unit cells, 7 instruments were recognized with MIR criterion and 9 instruments with the MAR criterion to produce a performance rate of 16/19. Some instruments were identified with a higher rate and the others were identified with a weaker score. This fact, can be interpreted as some instruments convey a specific spectral information that enhances the inter-variability between different instruments. The instruments then recognized with lower score are affected principally by the higher similarity related to instruments coming from the same family. One can remark that the individual performance rate increases from the neural structure with 20 nodes in the hidden layer to the structure with 60 nodes. Then the rate performance remains almost stable between the 120 and 240 nodes structures. We can conclude that optimal structure is there with 120 nodes in hidden layer in order to reduce the time calculation and the memory capacity.

Table IV illustrates the diagonal score recognition of the confusion matrix for various architectures of the multilayer perceptron networks with the family instruments accuracy. Only one family is recognized with MIR score and 3 out of 4 are identified with MAR. This shows that there is more similarity between instruments of the same family and therefore best performance is attained.

Table V and VI illustrate the impact of modifying the training and testing data set. The collection of database was divided in two different ways in order to construct a new set of the data training and testing. Then, two neural networks models with 120 unit cells in hidden layer were used to learn from each data distribution. The performance score was increased for 19 instruments from a rate of 16/19 to 18/19 at the instrument level and remained with the same performance at instrument level family. The results reveals for having an optimal performance, we should take into account the way of choosing the training data.

TABLE IV
Score Recognition for Families Instruments with MLP Network. SR Is the Score Recognition Using the Criterion MIR and MAR

	Number of unit cells for MLP			
	20	60	120	240
String	73%	83%	83%	84%
Brass	59%	64%	68%	64%
Reeds	51%	50%	53%	51%
Flutes	39%	45%	43%	45%
MIR	**1/4**	**1/4**	**1/4**	**1/4**
MAR	**3/4**	**3/4**	**3/4**	**3/4**
SR	**4/4**	**4/4**	**4/4**	**4/4**

TABLE V
Impact of Training Data Set. Score Recognition Basis Instruments with MLP Networks: SR(19) Is the Score for 19 Instruments and SR(14) Is the Score Recognition for 14 Instruments Using the Criterion MIR and MAR

	Number of unit cells for MLP	
instruments	120	120
Violin	71%	77%
Sop Sax	36%	39%
Tenor Trombone	62%	50%
Oboe	37%	29%
French Horn	23%	16%
Flute	20%	19%
EbClarinet	31%	32%
Cello	68%	66%
BbClarinet	74%	73%
Bass Trombone	23%	41%
Bassoon	51%	43%
Bass Flute	36%	40%
Alto Sax	46%	40%
Alto Flute	38%	26%
BassClarinet	24%	17%
Bass	78%	77%
Trumpet	52%	53%
Tuba	95%	91%
Viola	85%	83%
MIR	**7/19**	**10/19**
MAR	**9/19**	**8/19**
SR(19)	**16/19**	**18/19**
SR(14)	**12/14**	**14/14**

TABLE VI

Impact of Training Data Set. Score Recognition for Families Instruments with MLP Networks. SR Is the Score Recognition Using the Criterion MIR and MAR

	Number of unit cells for MLP	
families	120	120
String	83%	85%
Brass	68%	67%
Reeds	53%	51%
Flutes	43%	41%
MIR	**1/4**	**1/4**
MAR	**3/4**	**3/4**
SR	**4/4**	**4/4**

With 14 instruments the score increased for 19 from a rate of 12/14 to 14/14. Compared to a previous work [Krishna 04] using the same database with 14 instruments, the Line Spectral Frequencies (LSF) as parameters and 46 Mixtures Gaussians Models (GMM), approximately the same performance were observed.

TABLE VII
Score Recognition for Instruments with RBF Network. SR Is the Score Recognition Using the Criterion MIR and MAR

instruments	RBF
Violin	100%
Sop Sax	39%
Tenor Trombone	14%
Oboe	50%
French Horn	17%
Flute	24%
EbClarinet	24%
Cello	62%
BbClarinet	74%
Bass Trombone	21%
Bassoon	47%
Bass Flute	37%
Alto Sax	47%
Alto Flute	25%
MIR	**7/14**
MAR	**4/14**
SR(14)	**11/14**

.

TABLE VIII
Score Recognition for Instruments Identification Using the Mean of Features and the MLP Networks: SR(19) Is the Score for 19 Instruments and SR(14) Is the Score Recognition for 14 Instruments Using the Criterion MIR and MAR

Families	RBF
String	41%
Brass	28%
Reeds	46%
Flutes	44%
MIR	**4/4**
MAR	**0/4**
SR	**4/4**

Tables VII and VIII report the score recognition respectively for the 14 instruments and the 4 families with the Radial Basis Functions. The total family score is the same as the multilayer perceptron neural network but the score elements of the diagonal confusion matrix decrease significantly. However with the instruments only 11/14 are registered as score performance with the RBF network.

IV. CONCLUSION

Music processing, recognition, identification, segmentation is now an immense challenge and practical reality. In this study, we are interested in the instrument identification task in the monophonic context. We used only the spectral information as parameters by applying a discrete Fourier transform. Different architectures and models of neural networks were experimented as classifiers. We found that the spectral information is sufficient to discriminate well the family instrument but it cannot only work well to recognize each instrument particularly those coming from the same family. The best score is obtained with back-propagation neural network with 60 and 120 unit cells at the hidden layer. It was also noted that that the choice of the training and testing data play a crucial role and must be considered as another factor for enhancing performance of the instruments recognition cases.

REFERENCES

[1] Brown J. C., Houix O. and McAdams S., Feature dependence in the automatic identification of musical woodwind instruments, J. Acoust. Soc. Am., Vol. No. 3, pp. 1064-1072, 2001.
[2] Dubnov S. , Tishby N. and Cohen D., Polyspectra as measures of sound texture and timbre, J. New Music Res.26, 277-314, 1997.
[3] Eggink Jana and Brown Guy, A missing feature approach to instrument identification in polyphonic music, Proc. of ICASSP, pp. V-553-556, 2003.
[4] Kashino K. and Murase H., A sound source identification system for ensemble music stream extraction. Speech Comm. 27, pp. 337-349, 1999.
[5] Marques J. and Moreno P., A study of musical instrument classi.cation using gaussian mixture models and support vector machines, Cambridge Research Labs Technical Report Series CRL/4, 1999.

[6] Matlab, the language of technical computing, version 6.0.0.88 Release 12. Copyright Mathworks, inc., 2003.
[7] Krishna A.G and T.V Sreenivas, Music instruments recognition: from isolated notes to solo phrases, Proc of ICASSP, pp. IV-265-268, 2004.

ISAST Transactions on Intelligent Systems, No. 1, Vol. 1, 2008
J.M. Blackledge and D.A. Dubovitskiy: Object Detection and Classification with Applications to Skin Cancer Screening

Object Detection and Classification with Applications to Skin Cancer Screening

Jonathan M Blackledge, Fellow, IET and Dmitry A Dubovitskiy, Member, IET

***Abstract*— This paper discusses a new approach to the processes of object detection, recognition and classification in a digital image. The classification method is based on the application of a set of features which include fractal parameters such as the Lacunarity and Fractal Dimension. Thus, the approach used, incorporates the characterisation of an object in terms of its texture.**

The principal issues associated with object recognition are presented which includes two novel fast segmentation algorithms for which C++ code is provided. The self-learning procedure for designing a decision making engine using fuzzy logic and membership function theory is also presented and a new technique for the creation and extraction of information from a membership function considered.

The methods discussed, and the 'system' developed, have a range of applications in 'machine vision'. However, in this publication, we focus on the development and implementation of a skin cancer screening system that can be used in a general practice by non-experts to 'filter' normal from abnormal cases so that in the latter case, a patient can be referred to a specialist. A demonstration version of the application developed for this purpose has been made available for this publication which is discussed in Section IX.

***Index Terms*— Computer vision, Segmentation, Object recognition, Contour tracing, Decision making, Self-learning, Fuzzy logic, Image morphology, Skin cancer screening.**

I. Introduction

IMAGE analysis involves the use of image processing methods that are often designed in an attempt to provide a machine interpretation of an image, ideally, in a form that allows some decision criterion to be applied [1], [2]. Pattern recognition uses a range of different approaches that are not necessarily based on any one particular theme or unified theoretical approach. The main problem is that, to date, there is no complete theoretical model for simulating the processes that take place when a human interprets an image generated by the eye, i.e. there is no fully compatible model, currently available, for explaining the processes of visual image comprehension. Hence, machine vision remains a rather elusive subject area in which automatic inspection systems are advanced without having a fully operational theoretical framework as a guide. Nevertheless, numerous algorithms for understanding two- and three-dimensional objects in a digital image have and continue to be researched in order to design systems that can provide reliable automatic object detection, recognition and classification in an independent environment, e.g. [3], [4], [5], [7].

Vision can be thought of as the process of linking parts of the visual field (objects) with stored information or 'templates' with regard to a pre-determined significance for the observer. There are a number of questions concerning vision such as: (i) what are the goals and constraints? (ii) what type of algorithm or set of algorithms is required to affect vision? (iii) what are the implications for the process, given the types of hardware that might be available? (iv) what are the levels of representation required to achieve vision? The levels of representation are dependent on what type of segmentation can and/or should be applied to an image. For example, we may be able to produce primal sketches from an image via some measure of the intensity changes in a scene which are recorded as place tokens and stored in a database. This allows sets of raw components to be generated, e.g. regions of pixels with similar intensity values or sets of lines obtained by isolating the edges of an image scene, computed by locating regions where there is a significant difference in the intensity. However, such sets are subject to inherent ambiguities when computed from a given input image and associated with those from which an existing data base has been constructed. Such ambiguities can only be overcome by the application of high-level rules, based on how humans interpret images, but the nature of this interpretation is not always clear. Nevertheless, parts of an image will tend to have an association if they share size, colour, figural similarity, continuity, shading and texture, for example. For this purpose, we are required to consider how best to segment an image and what form this segmentation should take.

The identification of the edges of an object in an image scene is an important aspect of the human visual system because it provides information on the basic topology of the object from which an interpretative match can be achieved. In other words, the segmentation of an image into a complex of edges is a useful pre-requisite for object identification. However, although many low-level processing methods can be applied for this purpose, the problem is to decide which object boundary each pixel in an image falls within and which high-level constraints are necessary. Thus, in many cases, a principal question is, which comes first, recognition or segmentation?

Compared to image processing, computer vision (which incorporates machine vision) is more than automated image processing. It results in a conclusion, based on a machine

Manuscript received November 1, 2007. This work was supported by the Council for Science and Technology, by Microsharp Limited and by Management and Personnel Services Limited.

Jonathan Blackledge is Visiting Professor, Advanced Signal Processing Research Group, Department of Electronic and Electrical Engineering, Loughborough University, England and Extraordinary Professor of Computer Science, Department of Computer Science, University of the Western Cape, Cape Town, Republic of South Africa (e-mail: jon.blackledge@btconnect.com). Dr Dmitry Dubovitskiy is Director of Oxford Recognition Limited (e-mail: dda@oxreco.com).

performing an inspection of its own. The machine must be programmed to be sensitive to the same aspects of the visual field as humans find meaningful. Segmentation is concerned with the process of dividing an image into meaningful regions or segments. It is used in image analysis to separate features or regions of a pre-determined type from the background; it is the first step in automatic image analysis and pattern recognition. Segmentation is broadly based on one of two properties in an image: (i) similarity; (ii) discontinuity. The first property is used to segment an image into regions which have grey (or colour) levels within a predetermined range. The second property segments the image into regions of discontinuity where there is a more or less abrupt change in the values of the grey (or colour) levels.

In this paper, we consider an approach to object detection in an image scene that is based on a new segmentation algorithm for edge recognition using a Contour Tracing Algorithm. This algorithm differs from conventional 'edge detection' techniques in two respects: (i) it is not based on detecting first or second order gradients in an image using conventional FIR filters [2]; (ii) it is independent of any binarisation process through application of a threshold. After detection, the object is analysed in terms metrics derived from both a Euclidean and fractal geometric perspective, the output fields being used to train a fuzzy inference engine. The recognition structure is based on some of the image processing, analysis and machine vision techniques reported in [6], for example. The approach considered is generic in that it can, in principle, be applied to any type of imaging modality for which there are numerous applications that include speech and image recognition where self-calibration and learning is often mandatory. Example applications may include remote sensing, non-destructive evaluation and testing and other applications which specifically require the classification of objects that are textural. However, in this paper we focus on one particular application, namely, the diagnosis of skin cancer for screening patients through a general practice. The system reported is, in principle, just one of a number of variations which can be used for medical image analysis and classification in general. This is because the system includes features that are based on the textural properties of an image (defined in terms of fractal geometric parameters including the Fractal Dimension and Lacunarity) which is an important theme is medical image analysis.

II. Feature Detection and Classification

Suppose we have an image which is given by a function $f(x, y)$ and contains some object described by a set of features $S = \{s_1, s_2, ..., s_n\}$. We consider the case when it is necessary to define a sample which is somewhat 'close' to this object in terms of a matching set. This task can be reduced to the construction of some function determining a degree of proximity of the object to a sample - a template of the object. Recognition is the process of comparing individual features against some pre-established template subject to a set of conditions and tolerances. This process commonly takes place in four definable stages: (i) image acquisition and filtering (as required for the removal of noise, for example); (ii) object location (which may include edge detection); (iii) measurement of object parameters; (iv) object class estimation. We now consider aspects of each step, details of which are discussed in the following sections. In particular, we consider the design features and their implementation together with their advantages, disadvantages and proposals for a solution whose application, in this paper, focuses on the problem of designing a skin cancer screening system. It is for this reason, that the examples given to illustrate the steps proposed, are 'system related'.

Image acquisition depends on the technology that is best suited for integration with a particular application. For pattern recognition in histopathology, for example, high fidelity digital images are required for image analysis whose resolution is, at least, compatible by the image acquisition equipment used for human inspection, e.g. an optical microscope. The colour images used in the current application discussed in this paper are, in general, relatively noise free and are digitised using a standard CCD camera. Nevertheless, it is important that good quality images are obtained that are homogeneous with regard to brightness and contrast through application of well diffused light sources. Unless consistently high quality images can be generated that are compatible with the sample images used to design a given computer vision system, then that same system can be severely compromised. The system discussed in this paper is based on an object detection technique that includes a novel segmentation method and must be adjusted and 'fine tuned' for each area of application. This includes those features associated with an object for which fractal models are well suited [1], [2], [14].

The system described in this paper provides an output (i.e. a decision) using a knowledge database which generates a result (a decision) by subscribing different objects. The 'expert data' in the application field creates a knowledge database by using supervised training with a number of model objects [9]. The recognition process is illustrated in Figure 1, a process that includes the following steps:

1) Image Acquisition and Filtering.
   A physical object is digitally imaged and the data transferred to memory, e.g. using current image acquisition hardware available commercially. The image is filtered to reduce noise and to remove unnecessary features such as light flecks.
2) Special Transform: Edge Detection.
   The digital image function $f_{m,n}$ is transformed into $\tilde{f}_{m,n}$ to identify regions of interest and provide an input dataset for segmentation and feature detection operations [8]. This transform avoids the use of conventional edge detection filters which have proved to be highly unreliable in the present application.
3) Segmentation.
   The image $\{f_{m,n}\}$ is segmented into individual objects $\{f^1_{m,n}\}, \{f^2_{m,n}\}, \ldots$ to perform a separate analysis of each region. This step includes such operations as thresholding, morphological analysis, edge or contour tracing (Section IV) and the convex hull method (Section V).
4) Feature Detection.

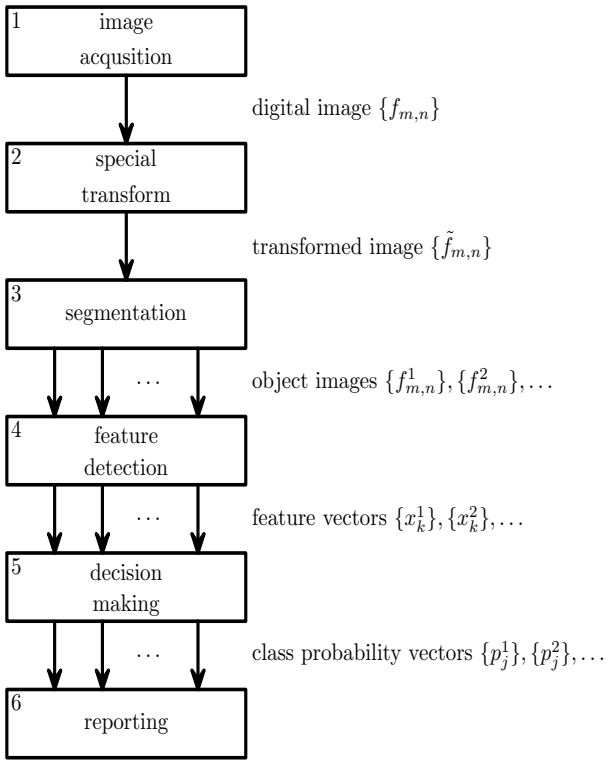


Fig. 1. Recognition process

Feature vectors $\{x^1_k\},\{x^2_k\},\dots$ are computed from the object images $\{f^1_{m,n}\},\{f^2_{m,n}\},\dots$ and corresponding transformed images $\{\tilde{f}^1_{m,n}\},\{\tilde{f}^2_{m,n}\},\dots$. The features are numeric parameters that characterize the object inclusive of its texture. The feature vectors computed consist of a number of Euclidean and fractal geometric parameters together with statistical measures in both one- and two-dimensions. The one-dimensional features correspond to the border of an object whereas the two-dimensional features relate to the surface within and/or around the object.

5) Decision Making.
This involves assigning a probability to a predefined set of classes [12]. Probability theory and fuzzy logic [10] are applied to estimate the class probability vectors $\{p^1_j\},\{p^2_j\},\dots$ from the object feature vectors $\{x^1_k\},\{x^2_k\},\dots$. A fundamental problem has been to establish a quantitative relationship between features and class probabilities, i.e.

$$\{p_j\} \leftrightarrow \{x_k\}$$

where $\leftrightarrow$ denotes a transformation from class probability to feature vector space. A 'decision' is the estimated class of the object coupled with the probabilistic accuracy [11].

This paper reports on a number of new algorithms that have been designed to solve problems associated with the above steps. Two new morphological algorithms for object segmentation have been considered which include auto-threshold selection. One of these algorithms - a contour tracing algorithm - extracts parameters associated with the spatial distribution of an object's border. This algorithm is also deployed in the role of feature detection. Another algorithm, that is concerned with computing a boundary with the 'convex hull' property, has been designed for operation in an environment where we do not have preliminary information about object position and orientation.

With regard to the decision making engine, the approach considered is based on establishing an expert learning procedure in which a Knowledge Data Base (KDB) is constructed based on answers that an expert makes during normal manual work. Once the KDB has been developed, the system is ready for application in the field and provides results automatically. However, the accuracy and robustness of the output depends critically on the extent and completeness of the KDB as well as on the quality of the input image, primarily in terms of its compatibility with those images that have been used to generate the KDB. The algorithm discussed in Section IV has no analogy with previous contour tracing algorithms and has been designed to trace the contour of an object with any level of complexity to produce an output that consists of a consecutive list of coordinates. The algorithm is optimised in terms of computational efficiency and can be realised in a compact form suitable for hardware implementation.

III. Segmentation

Segmentation is implemented by adaptive thresholding and morphological analysis. The adaptive image threshold is given by

$$T = \begin{cases} T_x, & T_x \geq T_y; \\ T_y, & \text{otherwise.} \end{cases}$$

where

$$T_x = \frac{1}{2}\left(\min_y\left(\max_x f(x,y)\right) - \langle \max_x f(x,y)\rangle_y\right)$$

$$+\langle \max_x f(x,y)\rangle_y,$$

$$T_y = \frac{1}{2}\left(\min_x\left(\max_y f(x,y)\right) - \langle \max_y f(x,y)\rangle_x\right)$$

$$+\langle \max_y f(x,y)\rangle_x.$$

Here, $\langle\cdot\rangle_x$ and $\langle\cdot\rangle_y$ are the means within column x and row y, respectively. This approach provides a solution for extracting the most significant features associated with a well defined object in the image frame. Thus, if an object covers an extensive image space, then this 'filter' provides the fastest compact solution. For example, in the skin cancer screening application considered here, there is preliminary information based on the fact that there is just one object on the image (as shown in the example given in Figure 2). In order to obtain a clear boundary, the morphological analysis applied here selects objects with a predefined area.

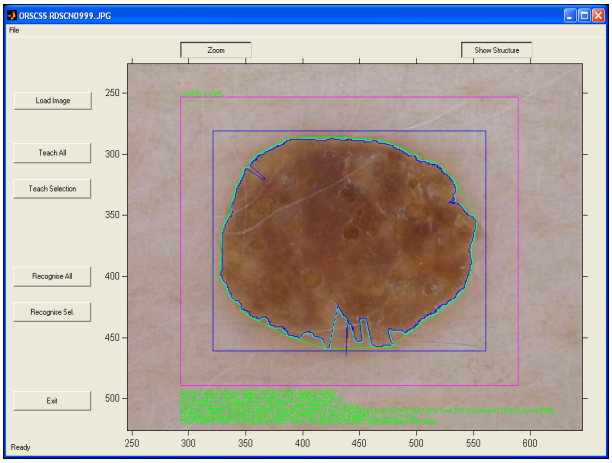


Fig. 2. Example of object segmentation applied to a skin cancer screening system.

IV. Detour Over an Object Contour: Contour Tracing Algorithm

After application of the segmentation algorithm described in the previous section and subsequent binarisation, the two dimensional (binary) representation of the object is the index map $f_{\rm bin}[m,n]$. This map has the same dimensions as the initial image $f[m,n] \equiv f_{m,n}$ where '1' corresponds to the object and '0' corresponds to the background image. It is then necessary to generate a serial list of boundary coordinates associated with the edge in which the inscribed object is set. Here, we use a novel detour algorithm on an object contour to derive this list of coordinates. The algorithm is both efficient and accurate and is profile independent when compared to other published algorithms, e.g. [13].

Consider the image in Figure 3. The start point (point 'A' in Figure 3) is not significant and, if necessary, can be determined from previous processing stages. For simplicity,

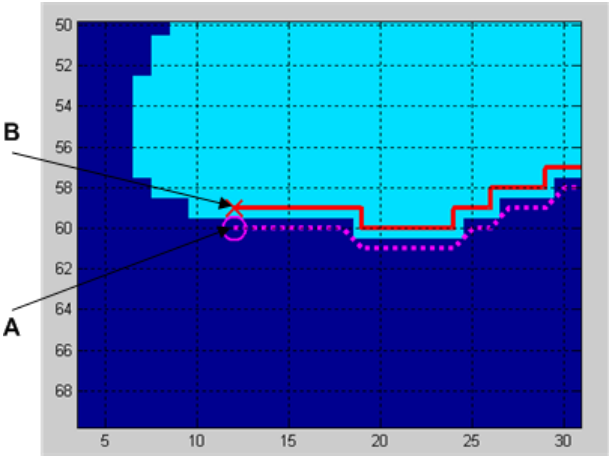


Fig. 3. Detour along an object contour

let the detour algorithm for evaluating an objects contour be named 'Sprocket wheel' because this virtual sprocket is rolled on to a virtual contour. Let us now zoom in on the image and observe how this sprocket wheel looks together with the binary map of an object as in Figure 4. We take the minimum radius of the wheel equivalent to the distance between two points on the image corresponding to a surface consisting of 3×3 elements. Let point 'A' correspond to a wheel axle, with the dashed-line curve, as given in Figure 3, showing its track. One of the points of the wheel will be connected to the objects edge at point 'B' (Figure 3). From the initial conditions, the coordinates of arbitrary points 'A' and 'B' are known. These coordinates can be recovered from preliminary processing or can be found by scanning for the nearest transition from 0 to 1. Thus, the coordinate of a point 0 will correspond to the coordinate of a point 'A', and 1 according to a point 'B'. The

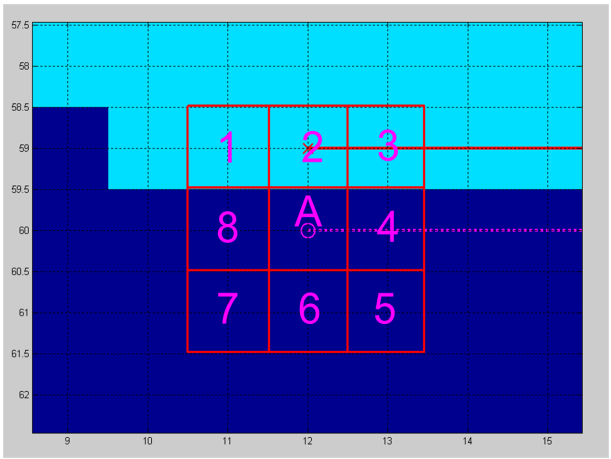


Fig. 4. Structure of a sprocket wheel

direction of movement has no value in the example above and so we consider a counter-clockwise motion. The motion of the virtual sprocket continues along the boundary with the current position of the axis conforming to the initial conditions. For simplicity, we assume that the object does not involve the image boundary. The C++ code for this algorithm is given in Figure 5 which computes the list of coordinates of the edge points of the segmented and binarised object. An example of implementing this algorithm is given in Figure 6 for the object given in Figure 9. The red line of connected points in the figure shows the edge of the object. With reference to the C++ code given in Figure 5, the coordinate data are contained in arrays $ListDotsX[0...ks]$ and $ListDotsY[0...ks]$ for X and Y, respectively.

The advantage of this algorithm over conventional edge detection techniques is that the system considers not only the brightness gradient but also the spatial distribution in terms of the object as a whole. The benefit of this approach is that the movement of axial coordinates occurs less often than the change of edge points and therefore, the computational costs are reduced on average by a factor 2-3 and depend only on the complexity of the object.

The contour generating algorithm described above, whose details are compounded in the C++ code given, is of value in determining the edges of a binarised image. However, in the application considered in this paper, it is applied to produce a contour signal whose fractal properties are used to compute one of a number of features which are discussed later.

Fig. 5. C++ algorithm for contour generation (object edge recognition)

```
int ks=0;
ListDotsX[0]=StartX;
ListDotsY[0]=StartY;
long DotX[9]={0,-1,0,1,1,1,0,-1,-1};//Extend surface of
long DotY[9]={0,-1,-1,-1,0,1,1,1,0};//wheel from axle.
int Ox=StartX;// Set position of points 'A' and 'B'.
int Oy=StartY+1;
int ht,HaveToch=2;
do {  // Cycle while not returning to initial coordinates.
  for (nl=1;nl<=7;nl++){ // Cycle surface of wheel.
    ht=HaveToch+nl;
    if (ht>8) ht=ht-8;
    x1=Ox+DotX[ht]; //Calculate coordinates for
    y1=Oy+DotY[ht]; //surface of the wheel.
    if (*(pp + x1*h + y1)==0){ //If 0 then move the wheel
      Ox=x1; // axle and calculate the point of tangency
      Oy=y1; // of surface with object edge.
      if((ht==1)||(ht==3)||(ht==5)||(ht==7)) HaveToch=ht+5;
      if((ht==2)||(ht==4)||(ht==6)||(ht==8)) HaveToch=ht+6;
      if (HaveToch>8) HaveToch=HaveToch-8;
      break;
    }
    if (*(pp + x1*h + y1)==1){ //If 1 then check the
      if ((x1==StartX)&&(y1==StartY)) break;// initial
      ks=ks++;//conditions
      ListDotsX[ks]=x1;// and save the edge
      ListDotsY[ks]=y1;// coordinate of the object.
    }
  }
} while ((x1!=StartX)||(y1!=StartY));
```

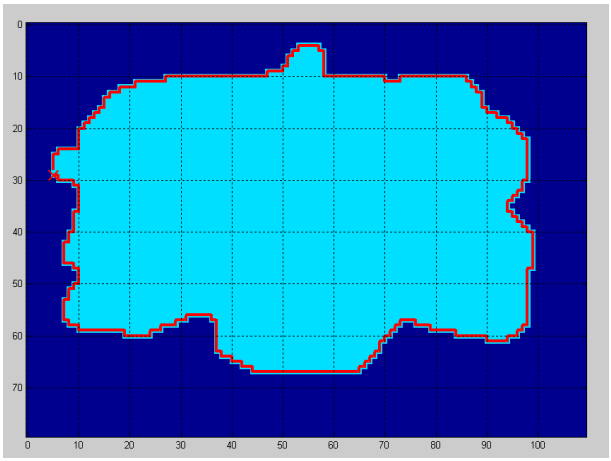


Fig. 6. Result generated by the Contour Walk Algorithm

V. Convex Hull Algorithm: 'Spider'

We now consider the task of obtaining the coordinates of a convex polygon for a binarised image. The binary image has been selected for explanatory purposes only. However, in general, this algorithm can be used as a segmentation procedure for image recognition. This task is given in the MathWorks MATLAB function 'Qhull'. However, the algorithm designed for this application differs from that available in MATLAB in terms of its simplicity, reliability and computational speed. The reason for this is that the number of cycles performed is limited and equal only to the total border length of the object.

The main idea can be thought of in terms of a 'Spider' walking over a contour and pulling a thread behind it. This thread is attached to the object. At the 'point of curvature', the thread stores the coordinates of the outer polygonal point. Thus, the path of the perimeter around the object provides the coordinates of all the outer polygonal points as illustrated in Figure 7. For the initial conditions, we select a position of

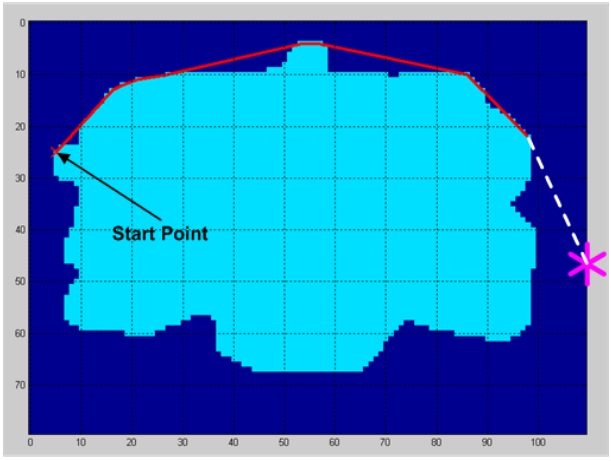


Fig. 7. Coordinate determination for a Convex Hull

Fig. 8. C++ algorithm for Convex Hull

```
{NListDotsX[02*((maxX-minX)+(maxY-minY))] //Create dot list
 NListDotsY[02*((maxX-minX)+(maxY-minY))]}//of bound object.
ListDotsX[0]=StartX; // Set the initial coordinates
ListDotsY[0]=StartY; // for end of thread.
int nc=0,x4,y4,Mx4,My4;
double fi,cs,sn,step,r,RR,bz,sz;
for(nt=0;nt<(2*((maxX-minX)+(maxY-minY)));nt++){//Begin walk
  fi=atan2(NListDotsY[nt]-StartY,...          // around object.
  NListDotsX[nt]-StartX);
    RR=sqrt(pow((NListDotsX[nt]-StartX),2)+...
                          +pow((NListDotsY[nt]-StartY),2));
    cs=cos(fi);
    sn=sin(fi);
    if (fabs(sn)>fabs(cs)){ //Calculate the step length.
      bz=fabs(sn);
      sz=fabs(cs);
    }else{
      bz=fabs(cs);
      sz=fabs(sn);
    }
    step=sqrt(pow(((sz*(1-bz))/bz),2)+pow((1-bz),2))+1;
    for (r=0;r<=RR;r+=step){ // Search for all objects
      x4=round((double)StartX + r*cs);//in line of thread.
      y4=round((double)StartY + r*sn);
      if (*(ppg + x4*h + y4) == 1){
        Mx4=x4; // Save last coordinate
        My4=y4; // in temporary variables.
      }
    }
    if (((Mx4!=StartX)&&(My4!=StartY)) || //Last dot check.
 ((Mx4==StartX)&&(Mx4==NListDotsX[nt])&&(Mx4!=NListDotsX[nt+1]))||
 ((My4==StartY)&&(My4==NListDotsY[nt])&&(My4!=NListDotsY[nt+1]))){
      StartX=Mx4; // Assign new start coordinates.
      StartY=My4;
      nc=nc++;
      ListDotsX[nc]=StartX; // Save list of coordinates
      ListDotsY[nc]=StartY; // for polygon.
    }
}
```

a thread. Clearly, this will be along one of the four image boundaries. The direction of a detour and the selection of the initial conditions does not depend on these conditions. In the example considered here, the detour is clockwise and starts along the left vertical boundary of the image. The C++ code for this algorithm is given in Figure 8.

This algorithm is also useful for defining the geometric location of separated points or objects and can be applied to the development of computer recognition systems, in general. An example of computing this type of polygon for the object is given in Figure 9, the output being represented by the green line. The convex hull algorithm provides information on the basic geometry of the object which yields information on

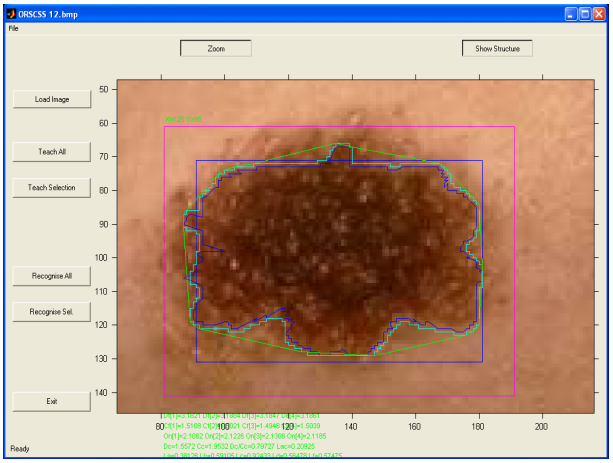


Fig. 9. Object with Contour and Convex Hull

the boundary area, the perimeter and so on. These Euclidean metrics are used to derive features which are discussed in the following section.

VI. Feature Determination

Features (which are typically compounded in a set of metrics - floating point or decimal integer numbers) describe the object state in an image and provides the input for a decision making engine (Figure 1). The features considered in this paper are computed in the spatial domains of the original image $\{f_{m,n}\}$ and transformed image $\{\tilde{f}_{m,n}\}$. Further, these features are extracted from different colour channels - Red (R), Green (G) and Blue (B) - captured by the CCD array. The issue of what type, and how many features should be used to develop a computer vision system, is critical in the design. The system considered here has been developed to include features associated with the texture of an object, features that are compounded in certain parameters associated with the field of fractal geometry. Texture is particularly important in medical image classification and of primary importance in the application (skin cancer screening) considered in this paper. The following features and their derivatives have been considered (primarily through numerical experimentation) in the recognition system reported in this paper:

Average Gradient G

describes how the intensity changes when scanning from the object center to the border. The object gradient is computed using the least squares method compounded in the following result:

$$g = \frac{N\sum\limits_{(m,n)\in S} r_{m,n}\tilde{f}_{m,n} - \sum\limits_{(m,n)\in S} r_{m,n} \sum\limits_{(m,n)\in S} \tilde{f}_{m,n}}{N\sum\limits_{(m,n)\in S} r_{m,n}^2 - \left(\sum\limits_{(m,n)\in S} r_{m,n}\right)^2},$$

where N is the number of pixels defining an object of compact support S and $r_{m,n}$ is the distance between (m,n) and the center (m',n'), i.e.

$$r_{m,n} = \sqrt{(m-m')^2 + (n-n')^2}.$$

The center coordinates (m',n') correspond to the local maximums of $\tilde{f}_{m,n}$ within the cluster. The cluster gradient is the average of object gradients,

$$G = \langle g_i \rangle_{i\in S}$$

where $i \in S$ is the object index.

Colour Composites Υ and Υ^D

characterise the relationship between the R, G and B layers of the transformed image. The triangle formula

$$r(a,b,c) = \sqrt{\frac{(s-a)(s-b)(s-c)}{s}},$$

$$s = \frac{1}{2}(a+b+c)$$

is applied to the 'colour triangle' RGB such that the following pixel colour composite is obtained

$$\upsilon_{m,n} = r(a,b,c)$$

where

$$a = \tilde{f}^R_{m,n}, \quad b = \tilde{f}^G_{m,n}, \quad c = \tilde{f}^B_{m,n}$$

and $\upsilon^D = r(a,b,c)$ with

$$a = |\tilde{f}^R_{m,n} - \tilde{f}^G_{m,n}|, \quad b = |\tilde{f}^G_{m,n} - \tilde{f}^B_{m,n}|$$

and

$$c = |\tilde{f}^R_{m,n} - \tilde{f}^B_{m,n}|.$$

The average colour composites are then given by

$$\Upsilon = \langle \upsilon_{m,n} \rangle_{(m,n)\in S}, \Upsilon^D = \langle \upsilon^D_{m,n} \rangle_{(m,n)\in S}.$$

Fourier Dimension q

determines the frequency characteristics of the object and is related to the fractal dimension D by $q = 4 - D$ [1], [2]. It represents a measure of texture [14] and describes a random fractal image with a power spectrum of the form

$$P^2(k_x,k_y) = c|k|^{-2q},$$

where $|k| = \sqrt{k_x^2 + k_y^2}$ is the spatial frequency and c is a constant. Both q and c can be computed using a least squares method [14].

Lacunarity (Gap Dimension) Λ_k

characterizes the way the 'gaps' are distributed in an image [2], [14]. The gap dimension is, roughly speaking, a measure of the number of light or dark regions in an image. It is defined for a degree k by

$$\Lambda_k = \left\langle \left| \frac{f_{m,n}}{\langle f_{m,n} \rangle} - 1 \right|^k \right\rangle^{\frac{1}{k}},$$

where $\langle f_{m,n} \rangle = \frac{1}{N}\sum f_{m,n}$ denotes the mean value. In the system described in this paper, an average of local Lacunarities of the degree $k = 2$ is measured.

Symmetry Features S_n and M

are estimated by morphological analysis in a three-dimensional space, i.e. two-dimensional spatial coordinates and intensity. A symmetry feature S_n is measured for a given degree of symmetry n (currently

$n = \{2, 4\}$). This value shows the deviation from a perfectly symmetric object, i.e. S_n is close to zero when the object is symmetric and $S_n > 0$ otherwise. Feature M describes the fluctuation of the centre of mass for pixels with different intensities; $M = 0$ for symmetric objects and $M > 0$ otherwise.

Structure γ

provides an estimation of the 2D curvature of the object in terms of the following:

$\gamma < 0$, if object bulging is less than a threshold,

$\gamma = 0$, if the object has standard bulging,

$\gamma > 0$, if object has a higher level of bulging.

Geometrical Features

include the minimum $R_{\min}$ and maximum $R_{\max}$ radius of the object (or ratio $R_{\max}/R_{\min}$), object area S, object perimeter P (or ratio S/P^2) and the coefficient of infill S/S_R, where S_R is the area of the bounding polygon which, in this application, is determined using the convex hull algorithm given in Section V.

The present solution detects objects by computer analysis using mixed mode features that are based on Euclidean and fractal metrics. The procedure of object detection is performed at the segmentation stage and needs to be adjusted for each area of application. The recognition algorithm then makes a decision using a knowledge database and outputs a result by subscribing objects based on the features defined above. The 'expert data' associated with a given application creates a knowledge database by using the supervised training system with a number of model objects as described in the following section.

VII. Object Recognition

In order to characterize an object, the 'system' has to know its mathematical representation. Here, this representation is based on the features considered in the previous section which are used to create an image of the object in the 'electronic mind'. This includes the textural features (Fractal Dimension and Lacunarity) for the object coupled with the Euclidean and morphological measures defined. In the case of a general application, all objects are represented by a list of parameters for implementation of supervised learning - Section VII(B) - in which a fuzzy logic system automatically adjusts the weight coefficients for the input feature set.

The methods developed represent a contribution to pattern recognition based on fractal geometry (at least in a partial sense), fuzzy logic and the implementation of a fully automatic recognition scheme as illustrated in Figure 10 for the Fractal Dimension D (just one element of the feature vector used in practice). The recognition procedure uses the decision making rules from fuzzy logic theory [9], [10], [11], [12] based on all, or a selection, of the features defined and discussed in Section VI which are combined to produce a feature vector $\mathbf{x}$.

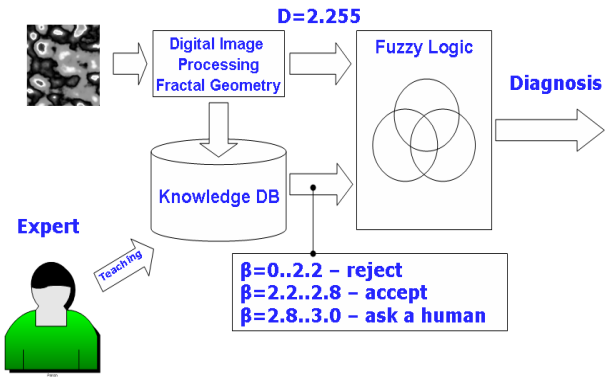


Fig. 10. Basic architecture of the diagnostic system based on the Fractal Dimension D (a single feature) and decision making criteria β.

A. Decision Making

The class probability vector $\mathbf{p} = \{p_j\}$ is estimated from the object feature vector $\mathbf{x} = \{x_i\}$ and membership functions $m_j(\mathbf{x})$ defined in a knowledge database. If $m_j(\mathbf{x})$ is a membership function, then the probability for each j^{th} class and i^{th} feature is given by

$$p_j(\mathbf{x}_i) = \max\left[\frac{\sigma_j}{|\mathbf{x}_i - \mathbf{x}_{j,i}|} \cdot m_j(\mathbf{x}_{j,i})\right]$$

where σ_j is the distribution density of values $\mathbf{x}_j$ at the point $\mathbf{x}_i$ of the membership function. The next step is to compute the mean class probability given by

$$\langle p \rangle = \frac{1}{j}\sum_j \mathbf{w}_j p_j$$

where $\mathbf{w}_j$ is the weight coefficient matrix. This value is used to select the class associated with

$$p(j) = \min\left[(p_j \cdot \mathbf{w}_j - \langle p \rangle) \geq 0\right]$$

providing a result for a decision associated with the j^{th} class. The weight coefficient matrix is adjusted during the learning stage of the algorithm.

The decision criterion method considered here represents a weighing-density minimax expression. The estimation of the decision accuracy is achieved by using the density function

$$d_i = |\mathbf{x}_{\sigma_{\max}} - \mathbf{x}_i|^3 + [\sigma_{\max}(\mathbf{x}_{\sigma_{\max}}) - p_j(\mathbf{x}_i)]^3$$

with an accuracy determined by

$$P = \mathbf{w}_j p_j - \mathbf{w}_j p_j \frac{2}{\pi}\sum_{i=1}^{N} d_i.$$

B. Supervised Learning Process

The supervised learning procedure is the most important part of the system for operation in automatic recognition mode. The training set of sample objects should cover all ranges of class characteristics with a uniform distribution together with a universal membership function. This rule should be taken into account for all classes participating in the training of the

system. An expert defines the class and accuracy for each model object where the accuracy is the level of self confidence that the object belongs to a given class. The Graphical User Interface (GUI) designed for the training procedure is shown in Figure 13. During this procedure, the system computes and transfers to a knowledge database, a vector $\mathbf{x} = \{x_i\}$, which forms the membership function $m_j(\mathbf{x})$. The matrix of weight factors $w_{j,i}$ is formed at this stage accordingly for the i^{th} parameter and j^{th} class using the following expression:

$$w_{i,j} = \left| 1 - \sum_{k=1}^{N} \left(p_{i,j}(\mathbf{x}_{i,j}^k) - \langle p_{i,j}(\mathbf{x}_{i,j}) \rangle \right) p_{i,j}(\mathbf{x}_{i,j}^k) \right|.$$

The result of the weight matching procedure is that all parameters which have been computed but have not made any contribution to the characteristic set of an object are removed from the decision making algorithm by setting $w_{j,i}$ to null.

VIII. Discussion

The methods discussed in the previous sections represent a novel approach to designing an object recognition system that is robust in classifying textured features, the application considered in this paper, having required a symbiosis of the parametric representation of an object and its geometrical invariant properties. In comparison with existing methods, the approach adopted here has the following advantages:

Speed of operation. The approach uses a limited but effective parameter set (feature vector) associated with an object instead of a representation using a large set of values (pixel values, for example). This provides a considerably higher operational speed in comparison with existing schemes, especially with composite tasks, where the large majority of methods require object separation. The principal computational effort is that associated with the computation of the features defined in Section VI given the fast algorithms discussed in Sections IV and V.

Accuracy. The methods constructed for the analysis of sets of geometrical primitives are, in general, more precise. Because the parameters are feature values, which are not connected to an orthogonal grid, it is possible to design different transformations (shifts, rotational displacements and scaling) without any significant loss of accuracy compared with a set of pixels, for example. On the other hand, the overall accuracy of the method is directly influenced by the accuracy of the procedure used to extract the required geometrical tags. In general, the accuracy of the method will always be lower, than, for example, classical correlative techniques. This is primarily due to padding, when errors can occur during the extraction of a parameter set. However, by using precise parametrisation structures based on the features defined in Section VI, remarkably good results are obtained.

Reliability. The proposed approach relies first and foremost on the reliability of the extraction procedure used to establish the geometrical and parametric properties of objects, which, in turn, depends on the quality of the image; principally in terms of the quality of the contours. It should be noted that the image quality is a common problem in any vision system and that in conditions of poor visibility and/or resolution, all vision systems will fail. In other words, the reliability of the system is fundamentally dependent on the quality of the input data.

An additional feature of the system discussed in this paper, is that the sub-products of the image processes can be used for tasks that are related to image analysis such as a search for objects in a field of view, object identification, maintaining an object in a view field, optical correction of a view point and so on. These can include tasks involving the relative motion of an object with respect to another or with respect to background for which the method considered can also be applied - collision avoidance tasks, for example.

Among the characteristic disadvantages of the approach, it should be noted that: (i) The method requires a considerable number of different calculations to be performed and appropriate hardware requirements are therefore mandatory in the development of a real time system; (ii) the accuracy of the method is intimately connected with the required computing speed - an increase in accuracy can be achieved but may be incompatible with acceptable computing costs. In general, it is often difficult to acquire a template of samples under real life or field trial conditions which have a uniform distribution of membership functions. If a large number of training objects are non-uniformly distributed, it is, in general, not possible to generate accurate results.

The original approach to the decision process proposed includes the following important steps: (i) the estimation of the density distribution is accurately determined from the original samples in the membership function during a supervised learning phase which improves the recognition accuracy under non-ideal conditions; (ii) the pre-filtering procedures provide a good response to the required features of the object without generating noise; (iii) the segmentation procedures discussed in Sections IV and V efficiently select only those objects required; (iv) computation of fractal parameters, in particular, the Lacunarity, helps to characterize the textural features (in terms of their classification) associated with the object.

The integration of Euclidean with fractal geometric parameters provides a more complete 'tool-kit' for pattern recognition in combination with supervised learning through fuzzy logic criteria. In the following section, we consider the application of our approach for the design of a skin cancer screening system. Other applications that have been considered to date include a surface inspection system for quality control in the manufacture of steel, details of which will be considered in a future publication.

IX. Application to Skin Cancer Screening: ORSCSS

In this section, we describe the basis and operational performance associated with the Oxford Recognition Skin Cancer Screen System (ORSCSS) developed by Oxford Recognition Limited (ORL) in collaboration with Loughborough University.

Malignant Melanomas are increasingly common and a potentially fatal form of skin cancer, the incidence of which is

increasing at a rate greater than any other form of cancer. It is often difficult to visually differentiate a normal mole from abnormal and general practitioners do not usually have significant expertise to diagnose skin cancers. Skin cancer specialists can improve the identification rate by over 80% but are often severely overloaded by referrals from regional general practices. It is possible for a general practitioner to take a high quality digital image of the suspect region on a patients skin and email the result to a remote diagnosis center. However, this can also lead to a (remote) overload and it is for this reason that the system discussed here has been considered in response to developing a screening method that can 'filter' benign melanomas in a general practice.

The system developed has been designed for use with a standard PC with input from a good quality digital camera using Commercial Off-The Shelf hardware. It analyses the structure of a mole or other skin 'defects', detects cancer-identifying features, makes a decision using a knowledge database and outputs a result. Skin cancer experts create a knowledge database by training the system using a number of case-study images. This produces a KDB which 'improves' with the use of the system.

The current system is composed of the following basic steps:

1) Filtering
   The image is Wiener filtered [2] to reduce noise and remove unnecessary and obtrusive features such as light flecks.
2) Segmentation
   The image is segmented to perform a separate analysis of each object (moles and/or other skin features). Two segmentation modes are available:
   - Automatic Mode
     The software identifies a mole as the largest and darkest object in the image. This mode is applicable in most cases.
   - Manual Mode
     The area of interest is manually selected by the user. This is most useful in cases when multiple moles and/or foreign objects are present in the image with possible overlapping features, for example.
3) Feature Detection
   For each object, a set of recognition features are computed based on those discussed in Section VI. The features are numeric parameters that describe the object in terms of a variety of Euclidean and fractal geometries and statistical features in one- and two-dimensions. The one-dimensional features correspond to the border of a mole and the two-dimensional features relate to the surface within the object boundary. In addition, a recognition algorithm is used to analyse the mole *structure* as illustrated in Figure 11. This provides information on the possible growth of the object when an inspection is undertaken over a period of time.
4) Decision Making
   The system uses fuzzy logic to combine features into a decision. A decision is the estimated class of the object and its accuracy. In this particular application, the output is designed to give two classes: *normal* and *abnormal*. This provides the simplest output with regard to the use of the system in a general practice in which abnormal cases are immediately referred to a specialist.

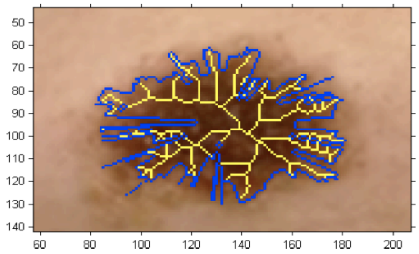


Fig. 11. Analysis of the structure of a mole for comparative growth analysis.

A. Key Advantages

The technology delivers high accuracy and automation which has been made possible by the following innovations:

Fractal analysis
: Biological structures (such as body tissues) have natural fractal properties. Numeric measurements of these properties enables efficient and effective detection of abnormalities.

Extended set of detectable features
: High accuracy is achieved when multiple features are measured together and combined into a single result.

Advanced fuzzy logic engine
: The knowledge-based recognition scheme used enables highly accurate diagnosis and offers significant improvements over current diagnostic methods.

B. Knowledge Database

ORSCSS is a knowledge-based system and requires extensive training before clinical operation. The training process includes a review and probabilistic classification of appropriate images by experts who can input results using the interface shown in Figure 13. The minimal number of training images depends on the number of classes and the diversity of objects within each class. An analysis and estimation of the number of (normal and abnormal) training images required is given in Section IX(H). The following sections describe how this application can be downloaded, installed and implemented. The demo version, which has been made available for this publication, includes documentation which is itemised in the following sections.

C. Platform Requirements

System Requirements

- Windows 98/ME/2000/XP
- CD-ROM Drive
- 256 MB RAM
- 30 MB hard disk space

Image Requirements

- Input format: JPEG, BMP or TIF
- Image size: 640x480 to 1024x728
  (higher image resolution requires RAM of 512 Mb and more)
- Good focus with no motion blur
- Uniform lighting
- Capture of the object which is well centred in the image frame and does not, for example, extend beyond the image boundaries

D. Installation

1) The downloadable demo version of ORSCSS is available from hppt://www.oxreco.com/setup.zip.
2) Installation is initiated through `setup.exe` from the root folder in which the downloaded application has been placed (after unzipping the downloaded file setup.zip).
3) Follow the instructions on screen.

E. Recognition Mode

1) Click **Load Image** and select an image of a mole or other skin 'defect'. Samples can be found in folder `Pictures`, which, by default, reside in

   `...\ORSCSS␣Demo\Pictures\`.

2) Click **Recognise All**. If the object(s) is not located automatically, then click **Recognise Selection** and select the area of interest.
3) Recognition and class estimation takes approximately 20 seconds (for a typical modern PC operating under an XP windows environment) producing an output of the type given in (Figure 12).

F. Teaching

1) The default knowledge database is loaded from

   `...\ORSCSS␣Demo\bin\def.kdb`.

   To create a new database, select **New knowledge DB** from the **File** menu (see Figure 14).
2) Click **Load Image** and select a picture of a mole, for example.
3) Click **Teach All**. If the mole is not found automatically click **Teach Selection** and select the area of interest.
4) ORSCSS analyses the mole for 10-30 seconds whereupon the Teaching Dialog (Figure 13) pops up. Enter your estimation:
   a) Class: number 1 (for Abnormal) or 2 (for Normal),
   b) Probability: a number between 0.0 and 1.0. 1.0 means you are absolutely sure, whereas zero should not normally be used. Typical values are 0.90-0.95.
5) Repeat Steps 1-4 above to process all training images.
6) Select **Save knowledge DB...** from **File** (see Figure 14) and enter a file name for the knowledge database.

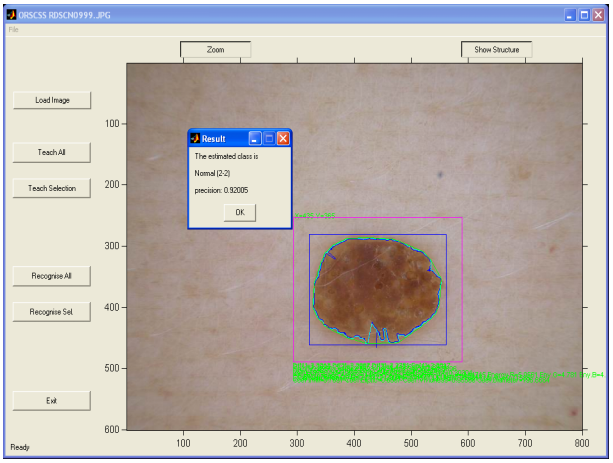


Fig. 12. Recognition result

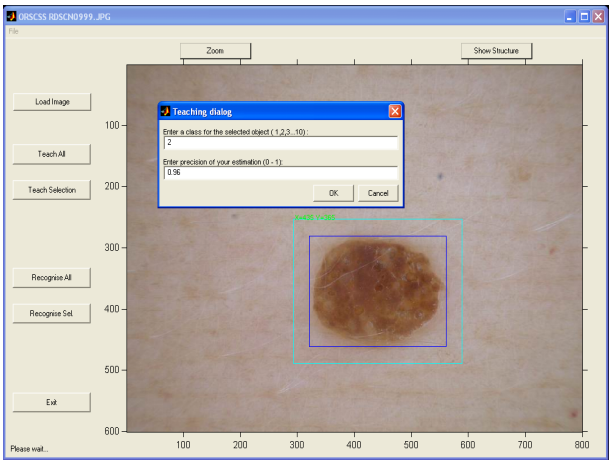


Fig. 13. Teaching dialog

G. User Interface

Main Window

The commands available from the main window (see Figure 2) are summarised in Table 1.

File Menu

The file menu is given in Figure 14 whose menu items and actions are summarised in Table II.

Command Line Execution

To launch the system in automatic mode type:

`ORSCSS.exe␣"LoadGraf"␣%1`

where `%1` is an image name (JPEG, BMP or TIFF formats are supported).

H. Estimation of the Minimal Number of Samples

There are approximately 65,000 new cases of skin cancer each year in the UK, which is about 5% of the total number of patients examined annually [15]. Let p be the probability of unrecognized cases. Then $q = 1 - p$ is the probability of recognized cases and the number of mistakes is determined by $S_n^k = C_n^k p^k q^{n-k}$ where n is number of experiments, k is the number of misidentifications and $C_n^k = \frac{n!}{k!(n-k)!}$.

Button	Action
Load Image	loads image in JPEG, BMP or TIF formats
Teach All	performs teaching in automatic mode
Teach Selection	performs teaching in manual mode
Recognise All	performs recognition in automatic mode
Recognise Selection	performs recognition in manual mode
Exit	closes the application
Zoom	switches on/off the zoom mode (use the left mouse button to zoom in and the right button to zoom out)
Show Structure	displays the mole boundary and features for growth analysis

TABLE I
BUTTONS AND ASSOCIATED ACTIONS ON THE MAIN WINDOW OF ORSCSS.

Menu Item	Action
New knowledge DB	resets knowledge database
Open knowledge DB...	loads knowledge database from a kdb file)
Save knowledge DB...	saves knowledge database to a kdb file
Open Image...	loads image in JPEG, BMP or TIF formats
Exit	closes the application

TABLE II
MENU ITEMS AND ASSOCIATED ACTIONS OF ORSCSS FILE MENU.

Our problem is to compute the number of images (tests) n required to estimate the error probability within the range $\pm 1\%$ (0.01) and degree of confidence $\alpha = 0.99$. Assuming the error is normally distributed with a standard deviation of σ, the probability of estimating the sampling fraction w within a degree of confidence Δ is defined by

$$P(|w - p| \leq \Delta) = \Phi(t),$$

where

$$\Phi(t) = \frac{2}{\sqrt{2\pi}} \int_0^t e^{-x^2/2} dx,$$

p is the probability of the error and $t = \Delta/\sigma$. Suppose we find p for 0.01 and let p' be the real probability. Setting the confidence interval at $\alpha = 0.99$, our minimal error is

$$P\left(|p' - p| < 0.01\right)$$

so that

$$P\left(|S_k^n - pn|0.01n\right) < \alpha$$

which, by the law of large numbers yields

$$0.01\frac{\sqrt{n}}{\sqrt{pq}} \geq 2.58$$

or

$$\sqrt{n} \geq 2580\sqrt{pq}$$

For all p, $pq \leq 1/4 \Rightarrow n \geq 16641$ and if $p < 0.1$ then $q > 0.9$ and $pq < 0.1$. We can then evaluate the minimal requirement n, i.e.

$$n \geq 6656$$

However, in practice, this number may not be enough to assess the accuracy of the recognition system due to the following reasons: (i) the assumption that p and w are constant for all types of moles is very doubtful and it is necessary to carry tests with a variety of skin defects; (ii) the recognition quality will significantly vary in time during the test process since the knowledge database is constantly updated. Nevertheless, the value of n given here provides an order of magnitude of the number of images required to train the system effectively.

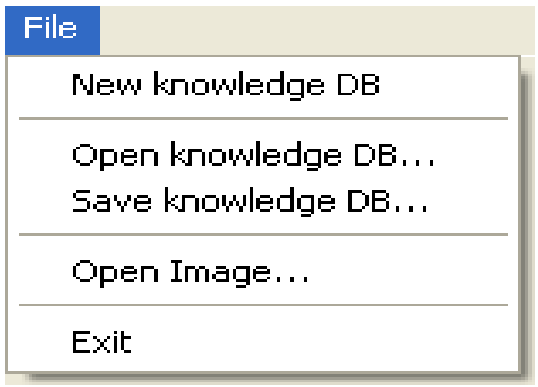


Fig. 14. File menu

I. Comparison with other approaches

There are a number of commercially available products which offer a range of aids and tools for skin cancer detection. Some of them use an extensive database to estimate the pathology and may require a relatively significant amount of time to make a decision. Other products calculate several properties and represent them graphically. Medical staff are then used to make a final decision. More interesting techniques involve the capture of images using different sensors or a multiplicity of different images. However, these systems are as yet, not approved for clinical diagnosis and are not a referenced form of dermatoscopy. The following list provides some of the more common products in the field: (i) MoleMAX - http://www.molechecks.com.au; (ii) DermLite - http://www.dermlite.com/mmfoto.html; (iii) DermoGenius Lite - http://www.dermogenius.de; (iv) MelaFind - www.melafind.com. Comparing these products with the methods developed for this paper, it is clear that there are no other automatic recognition systems with self-adjusting procedures and self-controlled functions. The tests undertaken to date, have established the capacity for ORSCSS to be used in routine clinical conditions provided extensive training of the system has been undertaken.

X. CONCLUSION

This paper has been concerned with the task of developing a methodology and implementing applications that are concerned with two key tasks: (i) the partial analysis of an image in terms of its fractal structure and the fractal properties that characterize that structure; (ii) the use of a fuzzy logic engine

to classify an object based on both its Euclidean and fractal geometric properties. The combination of these two aspects has been used to define a processing and image analysis engine that is unique in its modus operandi but entirely generic in terms of the applications to which it can be applied.

The work reported in this paper is part of a wider investigation into the numerous applications of pattern recognition using fractal geometry as a central processing kernel. This has led to the design of a new library of pattern recognition algorithms including the computation of parameters in addition to those that have been reported here such as the information dimension, correlation dimension and multi-fractals [14]. The inclusion or otherwise of such parameters in terms of improving vision systems such as the one considered here remains to be understood. However, from the work undertaken to date, it is clear that texture based analysis alone is not sufficient in order to design a recognition and classification system. Both Euclidean and fractal parameters need to be combined into a feature vector in order to develop an operational vision system which includes objects that have textural properties such as those associated with medical imaging.

The creation of logic and general purpose hardware for artificial intelligence is a basic theme for any future development based on the results reported in this paper. The results of the current system can be utilized in a number of different areas although medical imaging would appear to be one of the most natural fields of interest because of the nature of the images available, their complex structures and the difficulty of obtaining accurate diagnostic results which are efficient and time effective. A further extension of our approach is to consider the effect of replacing the fuzzy logic engine used to date with an appropriate Artificial Neural Network (ANN). It is not clear as to whether the application of an ANN could provide a more effective system and whether it could provide greater flexibility with regard to the type of images used and the classifications that may be required.

ACKNOWLEDGMENTS

The authors are grateful for the advice and help of Professor V Deviatkov and Professor A Chernikov (Moscow State Technical University), Professor Jonathan Brostoff (Kings College, London University), Dr Alastair Deery (Royal Free Hospital, London) and Professor Irina Shabalova (Russian Medical Academy of Postgraduate Education, Moscow).

REFERENCES

[1] J. M. Blackledge, *Digital Signal Processing*, 2nd Edition, Horwood Publishing, 2006.
[2] J. M. Blackledge, *Digital Image Processing*, Horwood Publishing, 2005.
[3] E. R. Davies, *Machine Vision: Theory, Algorithms, Practicalities*, Academic press, London, 1997.
[4] H. Freeman, *Machine vision. Algorithms, Architectures, and Systems*, Academic press, London, 1988.
[5] J. Louis and J. Galbiati, *Machine vision and digital image processing fundamentals*, State University of New York, New-York, 1990.
[6] R. Boyle, M. Sonka and V. Hlavac, *Image Processing, Analysis and Machine Vision*, PWS, USA, 1999.
[7] W. E. Snyder and H. Qi. *Machine Vision*, Cambridge University Press, England, 2004.
[8] V. S. Nalwa and T. O. Binford, *On detecting edge* IEEE Trans. Pattern Analysis and Machine Intelligence, (PAMI-8), 699-714, 1986.
[9] L. A. Zadeh, *Fuzzy sets and their applications to cognitive and decision processes*, Academic Press, New York, 1975.
[10] E. H. Mamdani, *Advances in linguistic synthesis of fuzzy controllers*, J. Man. Mach., 8, 669-678, 1976.
[11] E. Sanchez, *Resolution of composite fuzzy relation equations* Int. Control, 30, 38-48, 1976.
[12] N. Vadiee, *Fuzzy rule based expert system-I*, Prentice Hall, Englewood, 1993.
[13] Contour Tracing Algorithms http://www.cs.mcgill.ca/ aghnei/alg.html
[14] M. J. Turner, J. M. Blackledge and P. R. Andrews. *Fractal Geometry in Digital Imaging*. Academic Press, London, 1998.
[15] Cancer research uk, http://www.cancerresearchuk.org/ aboutcancer/reducingyourrisk/9314.
[16] J. S. Lim, *Two-Dimensional Signal and Image Processing*, Prentice-Hall, 1990.

Jonathan Blackledge received a BSc in Physics from Imperial College, London University in 1980, a Diploma of Imperial College in Plasma Physics in 1981 and a PhD in Theoretical Physics from Kings College, London University in 1983. As a Research Fellow of Physics at Kings College (London University) from 1984 to 1988, he specialized in information systems engineering undertaking work primarily for the defence industry. This was followed by academic appointments at the Universities of Cranfield (Senior Lecturer in Applied Mathematics) and De Montfort (Professor in Applied Mathematics and Computing) where he established new post-graduate MSc/PhD programmes and research groups in computer aided engineering and informatics. In 1994, he co-founded Management and Personnel Services Limited where he is currently Executive Director. His work for Microsharp (Director of R & D, 1998-2002) included the development of manufacturing processes now being used for digital information display units. In 2002, he co-founded a group of companies specializing in information security and cryptology for the defence and intelligence communities, actively creating partnerships between industry and academia. He currently holds academic posts in the United Kingdom and South Africa.

Dmitry Dubovitskiy received a BSc and Diploma in Aeronautical Engineering from Saratov Aviation Technical College in 1993, an MSc in Computer Science and Information Technology for the Moscow State Technical University 1999 and a PhD in Computer Science from De Montfort University in 2005. As a project leader in medical imaging at Microsharp Limited from 2002 to 2005, he specialized in information systems engineering, developing image recognition systems for medical applications for real time operational diagnosis. He is Executive Director and Principal Engineer at Oxford Recognition Limited which he founded in 2005 to undertake research and development, specialising in the application of artificial intelligence for computer vision. He has developed a range of computer vision systems for industry including applications for 3D image visualisation, face recognition, information security and digital rights management and is currently principal coordinator for the INTAS project in distributed automated systems for acquiring and analysing eye tracking data.

ISAST Transactions on Intelligent Systems, No. 1, Vol. 1, 2008
N. Le-Khac et al.: Knowledge Map Layer for Distributed Data Mining

Knowledge Map Layer for Distributed Data Mining

Nhien-An Le-Khac, Lamine M. Aouad and M-Tahar Kechadi

Abstract — Nowadays, massive amounts of data which are often geographically distributed and owned by different organisations are being mined. As consequence, a large mount of knowledge is being produced. This causes the problem of efficient knowledge management in distributed data mining (*DDM*). The main aim of *DDM* is to exploit fully the benefit of distributed data analysis while minimising the communication overhead. Existing *DDM* techniques perform partial analysis of local data at individual sites and then generate global models by aggregating the local results. These two steps are not independent since naive approaches to local analysis may produce incorrect and ambiguous global data models.

To overcome this problem, we introduce "knowledge map" approach to represent easily and efficiently the knowledge mined in a large scale distributed platform such as Grid. This approach is developed and integrated in a *DDM* framework. This will also facilitate the integration/coordination of local mining processes and existing knowledge to increase the accuracy of the final model. Our "knowledge map" is being tested on real large datasets.

Keywords — distributed data mining, knowledge map, knowledge management

I. Introduction

While massive amounts of data are being collected and stored from not only science fields but also industry and commerce fields, the efficient mining and management of useful information of this data is becoming a scientific challenge and a massive economic need. This led to the development of distributed data mining (*DDM*) techniques [16][17] to deal with huge and multi-dimensional datasets distributed over a large number of sites. This phenomenon leads to the problem of managing the mined results, so called knowledge, which becomes more complex and sophisticated. This is even more critical when the local knowledge of different sites are owned by different organisations. Existing *DDM* techniques is based on performing partial analysis on local data at individual sites and then generating global models by aggregating these local results. These two steps are not independent since naive approaches to local analysis may produce incorrect and ambiguous global data models. In order to take the advantage of mined knowledge at different locations, *DDM* should have a view of the knowledge that not only facilitates their integration but also minimises the effect of the local results on the global models. Briefly, an efficient management of distributed knowledge is one of the key factors affecting the outputs of these techniques.

Recently, many research projects on knowledge management in data mining were initiated [29][13][1]. Their goals are to tackle the knowledge management issues as well as present new approaches. However, most of them propose solutions for centralised data mining and only few of them have attempted the issues of large scale *DDM*. Moreover, some recent research works [4] have just provided a manner of managing knowledge but not the integration and coordination of these results from local results.

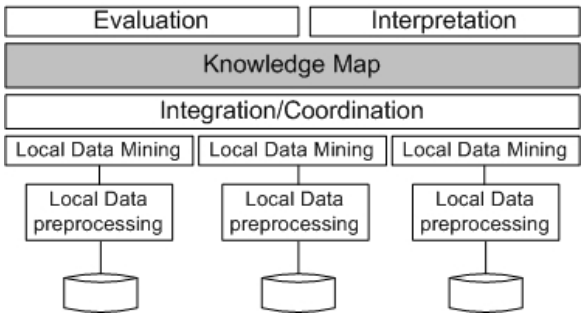


Fig. 1. ADMIRE's core architecture

In this paper, we propose a "knowledge map", an approach for managing knowledge of *DDM* tasks on large scale distributed systems and also supporting the integration views of related knowledge. The concept of knowledge map has been efficiently exploited in managing and sharing knowledge [24] in different domains but not yet in *DDM* techniques. Our main goal is to provide a simple and efficient way to handle a large amount of knowledge built from *DDM* applications in Grid environments. This knowledge map helps to retrieve quickly any results needed with a high accuracy. It will also facilitate the merging and coordination of local results to generate global models. This knowledge map is one of the key layers of ADMIRE [18] (Fig.1), a framework based on Grid platform for developing *DDM* techniques to deal with very large and distributed heterogeneous datasets.

The rest of this paper is organised as follows: In section II, we give some backgrounds of knowledge representation and knowledge map concept as well as related projects. We present the architecture of our knowledge map in section III. Section IV presents knowledge map's operations. Implementation issues of knowledge map are presented in section V and an evaluation of this approach is presented in section VI. Finally, we conclude in section VII.

II. Background

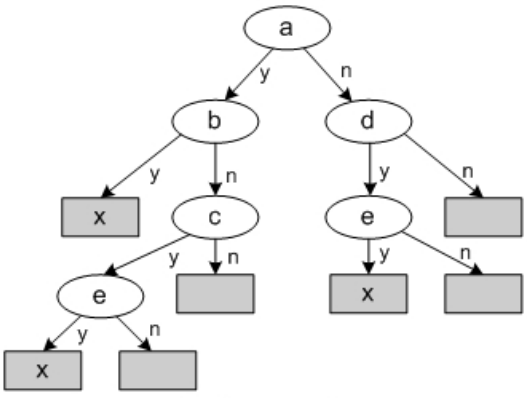
In this section, we present some methods for representing knowledge in data mining. We discuss the concept of knowledge map and its use in managing the knowledge. This section will be ended by related work on knowledge map and knowledge management.

Manuscript received October 10, 2007

Nhien-An Le-Khac, Lamine M. Aouad and M-Tahar Kechadi are with the School of Computer Science and Informatics, University College Dublin, Dublin 4, Ireland (email: an.lekhac, lamine.aouad, tahar.kechadi@ucd.ie)

weather	weekday	holiday	action
sunny	Y	N	go to work
rain	N	Y	stay at home
sunny	N	Y	go on a picnic

(a) Decision table



(b) Decision tree

Fig. 2. Knowledge representations

A. Knowledge representation

There are many different ways of representing mined knowledge in literature, such as decision tables, decision trees (Fig.2), classification rules, association rules, instance-based and clusters. *Decision table* is one of the simplest ways of representing knowledge. The columns contain set of attributes including the decisions and the rows represent the knowledge elements. This structure is simple but it can be sparse because of some unused attributes. *Decision tree* approach is based on "divide-and-conquer" concept where each node tests a particular attribute and the classification is given at the leaves level. However, it has to deal with missing value problem. *Classification rules* [9] are a popular alternative to decision tree. *Association rules* [9] are kind of classification rules except that they can predict any attribute and this gives them the flexibility to predict combinations of attributes too. Moreover, association rules are not intended to be used together as a set as classification rules are.

Classification rules as well as association rules are a kind of production rules [2] that are widely used in knowledge representation [12]. A rule is a knowledge representation technique and a structure that relates one or more causes, or a situation, to one or more effects (consequents) or actions. It is also called cause-effect relationships represented by an "IF {cause expression} THEN {conclusion expression}". The IF part of the rule is an cause expression composed of causes, and the effects are contained in the conclusion expression of THEN, so that the conclusions may be inferred from the causes when they are true. A rule may also be extended to an *uncertain rule* or a *fuzzy rule* by adding appropriate attributes. Briefly, the knowledge of an intelligent system could be represented by using a number of rules. In this case, these rules are usually grouped into sets and each set contains rules related to the same topic. In the data mining, rules can be used in the representation of knowledge learnt from classification tasks, association rules tasks, etc. It is also called rule-based classification [14] in classification problems where a set of "IF, THEN" rules including attributes such as *coverage* and *accuracy* is applied. Moreover, rules can be extracted from other kinds of model representations such as decision tree, neural network, etc. In association rule tasks, knowledge is represented by a set of rules with two attributes: confidence and support.

The instance-based knowledge representation uses the instances to represent what is mined rather than inferring a rule set and store it instead. The problem is that they do not make explicit the structures of the knowledge. In the cluster approach, the knowledge can take the form of a diagram to show how the instances fall into clusters. There are many kinds of cluster representations such as space partitioning, Venn diagram, table, tree, etc. Clustering [9] is often followed by a stage in which a decision tree or rule set is inferred allocating each instance to its cluster. Other knowledge representation approaches, such as Petri net [26], Fuzzy Petri nets [5] and G-net [8] were also developed and used.

B. Knowledge map concept

A knowledge map is generally a representation of "knowledge about knowledge" rather than of knowledge itself [7][10][30]. It basically helps to detect the sources of knowledge and their structures by representing the elements and structural links of the application domains. Some kind of knowledge map structure that can be found in literature are: hierarchical/radial knowledge map, networked knowledge map, knowledge source map and knowledge flow map.

Hierarchical knowledge map, so-called concept map [24], provides one model for the hierarchical organization of knowledge: top-level concepts are abstractions with few characteristics. Concepts of the level below have detailed traits of the super concept. The link between concepts can represent any type of relations as "is part of", "influences", "can determine", etc. A similar approach is radial knowledge map or mind map [3], which consists of concepts that are linked through propositions. However, it is radially organised. Networked knowledge map is also called causal map which is defined as a technique "for linking strategic thinking and acting, making sense of complex problems, and communicating with others what might be done about them" [3]. This approach is normally used for systematizing knowledge about causes and effects. Knowledge source map [10] is a kind of organisational charts that does not describe functions, responsibility and hierarchy, but expertise. It helps experts in a specific knowledge domain. The knowledge flow map [10] represents the order in which knowledge resources should be used rather than a map of knowledge.

C. Related works

Little research work on knowledge map is given in [11][22]. However, these few projects were not in the context of *DDM*. The Knowledge Grid project [4] proposed an approach

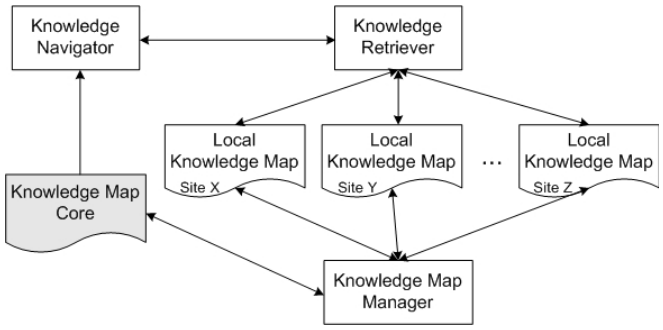


Fig. 3. Knowledge map system

to manage the knowledge by using Knowledge Discovery Service. This module is responsible for handling meta-data of not only knowledge obtained from mining tasks but also all kinds of resources such as hosts, data repositories, used tools and algorithms, etc. All metadata information is stored in a Knowledge Metadata Repository. However, this approach does not provide a management of meta-data of knowledge in their relationships to support the integration view of knowledge as well as the coordination of local the mining process. There is moreover no distinct separation between resource, data, and knowledge.

Until now, to the best of our knowledge, in spite of the popularity of *DDM* applications, there is only our system [19][21] that provides knowledge map layer for *DDM* applications on a Grid type platforms. This constitutes one of the motivations of our research to provide a fully integrated view of knowledge to facilitate the coordination of local mining processes and increase the accuracy of the final models.

III. Architecture of knowledge map

The knowledge map (*KM*) does not attempt to systematize the knowledge itself but rather to codify "knowledge about knowledge". In our context, it facilitates *DDM* by supporting users coordination and interpretation of the results. The objectives of our *KM* architecture are: provide an efficient way to handle a large amount of data collected and stored in large scale distributed system; retrieve easily, quickly, and accurately the knowledge; and support the integration process of the knowledge. We propose an architecture of the *KM* system as shown in Fig.3, 4, 5 and 6 to achieve these goals. *KM* consists of the following components: knowledge navigator, knowledge map core, knowledge retrieval, local knowledge map and knowledge map manager (Fig.3). From now on, we use the term "mined knowledge" to represent for knowledge built from applications.

A. Knowledge navigator

Usually, users may not exactly know the mined knowledge they are looking for. Thus, knowledge navigator component is responsible for guiding users to explore the *KM* and for determining the knowledge of interest. The result of this task is not the knowledge but its meta-data, called ***meta-knowledge***, which includes related information such as data mining task used, data type, and a brief description of this knowledge and its location. For example, a user may want to retrieve some knowledge about tropical cyclone. The application domain "meteorology" allows the user to navigate through tropical cyclone area and then a list of knowledge related to it will be extracted. Next, based on this meta-knowledge and its application domain, the users will decide which knowledge and its location are to be retrieved. It will interact with knowledge retrieval component to collect all mined knowledge from chosen locations.

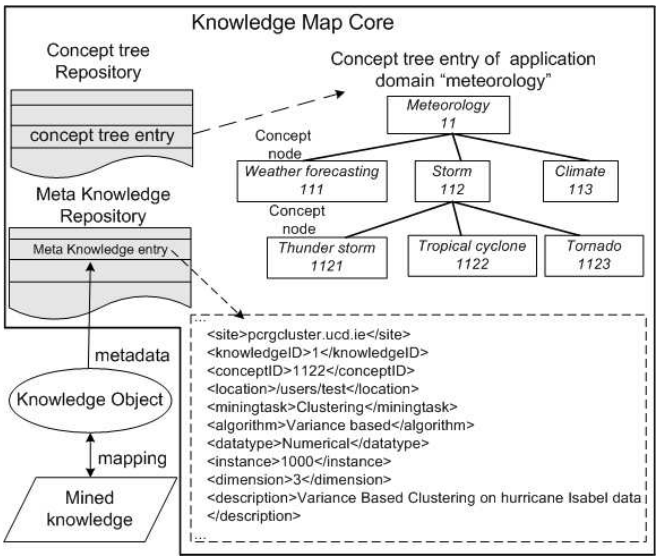


Fig. 4. Knowledge map core structure

B. Knowledge map core

This component (Fig.4) is composed of two main parts: *concept tree repository* and *meta-knowledge repository*. The former is a repository storing a set of application domains. Each application domain is represented by a *concept tree* that has a hierarchical structure such as a concept map [24]. A node of this tree, so called *concept node* represents a sub-application domain and each *concept node* includes a unique identity, called *concept identity*, in the whole *concept tree* repository and a name of its sub-application domain. The content of each *concept tree* is defined by the administrator before using the *KM* system. The concept tree repository could also be updated during the runtime. In our approach, a mined knowledge is assigned to only one sub-application domain and this assignment is given by the users.

As shown in Fig.4 for example, the ***concept tree repository*** contains an application domain named "meteorology" which includes sub-application domains such as "weather forecasting", "storm" and "climate". And then, "thunder storm", "tropical cyclone" and "tornado" are parts of "storm". By using *concept tree*, we can deal with the problem of knowledge context. For instance, given the distributed nature of the knowledge, some of them may have variations depending on the context in which it is presented locally. Moreover, we can also extend the concept tree of each application domain to an ontology of this domain in order to increase the accuracy in retrieving knowledge in different contexts. At that moment,

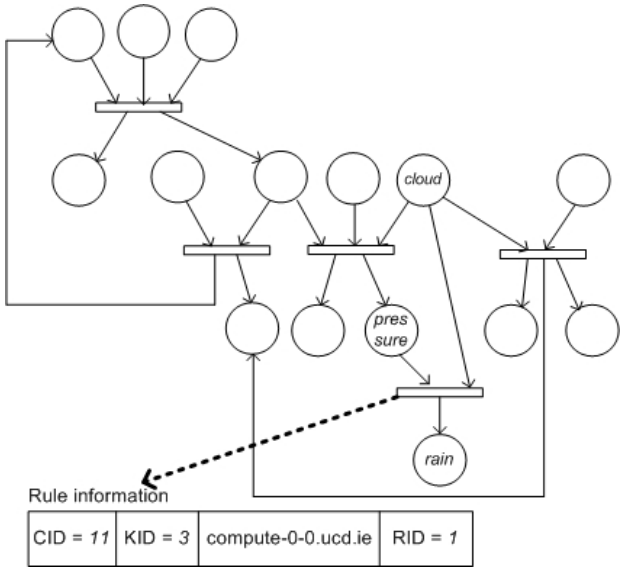


Fig. 5. Rule Net architecture

the concept tree will become a taxonomy tree and a list of term as well as slots [12] will be added. The ontology-based architecture of this repository will be applied in the next version of our *KM*.

Meta-Knowledge repository (Fig.4): this handles meta-data of the mined knowledge from different sites. A knowledge is mapped to a ***knowledge object*** and its meta-data is represented by a meta-knowledge entry in this repository. Figure 4 also shows an example of a meta-knowledge entry in XML format. In this example, this knowledge is built from "pcrgcluster.ucd.ie" (*knowledge location*) and its local identity (*knowledge identity KID*) is 1; its concept identity (*CID*) is 1122 (sub-application domain is *tropical cyclone*); the location of datasets is "/users/test/"; the used mining task is "clustering" and its algorithm is "variance-based" [20]. Other related information are data type of mined datasets, number of instances, dimension of data and a brief description about this knowledge. Based on this information, users could determine which mined knowledge they want to extract.

As presented in the section II-A, production rules are widely used in knowledge representation. So, each application domain is also linked to a distinct ***Rule-Net*** (Fig.5) which is a representation of relationships between rules and their items related to this domain. A production rule is of the form "IF {cause expression} THEN {conclusion expression}" and an expression (cause or conclusion) also contains a set of items. So, a ***Rule-Net*** is a triple-tuple *(I, R, A)* where *I* is a set of items participating in expressions of one or many rules, *R* is a set of rules, *A* is a set of edges called rule item relations. A should satisfy the following: $A \subseteq (I \times R) \bigcup (R \times I)$. A rule of a set *R* is part of mined knowledge of which their representative is a set of production rules. These mined knowledge are discovered from datasets of this application domain. A rule moreover contains its related information such as its identity,(*knowledge identity KID*) of mined knowledge containing this rule, location of this knowledge. The purpose of this ***Rule-Net*** is to provide a larger index of rules and items in order to support the knowledge retrieving process. For example, as shown in the Fig.5, a knowledge of the application domain "tropical cyclone" includes a rule as "IF *cloud, pressure* THEN *rain*" and the rule identity (*RID*) is equal *1*, the knowledge identity (*KID*) is equal *3*, its location is *compute-0-0.ucd.ie* and its application domain is *meteorology"* (*CID=11*). Based on information of *Rule Net*, users can search knowledge by a request such as "searching all meta-knowledge (represented by production rules) that contains the item "cloud" of application domain meteorology". In this case, links of item "cloud" in the *Rule Net* are used to determine the identity and location of related knowledge satisfying the request and then their meta-knowledge will be retrieved. Next, based on these meta-knowledge, the user can retrieve knowledge needed. Users can also use the narrower criteria such as reducing the search within the "tropical cyclone" domain.

The goal of *KM* core, is not only to detect the sources of knowledge and information but also represent their relationships with concepts of application domains. The location of this component depends on the topology of the system. It could be, for example, implemented in a master site assigned to a group of sites. The creation and maintenance of this component as well as its operations such as retrieving knowledge will be presented in section IV.

C. Knowledge retrieval

The role of this component is to seek the knowledge that is potentially relevant. This task depends on the information provided by the users after navigating through application domains and getting the meta-knowledge needed. This component is similar to a search engine which interacts with each site and returns knowledge acquired.

D. Local knowledge map

This component (Fig.6) is located in each site of the system where knowledge are built from datasets. *Local knowledge map* is a repository of knowledge entries. Each entry, which is a knowledge object, represents a mined knowledge and contains two parts: *meta-knowledge* and a *representative*. *Meta-knowledge* includes information such as the identity of its mined knowledge that is unique in this site, its properties, and its description. Theses attributes are already explained in the section ***Knowledge map core*** above. This *meta-knowledge* is also submitted to the ***Knowledge map core*** and will be used in *meta-knowledge entry* of its repository at the global level. The *representative* of a knowledge entry depends on a given mining task. *KM* supports two kinds of representatives: one for knowledge mined from clustering tasks and another for mined knowledge represented by production rule. Our system has however the capacity of adding more representative types for other mining tasks.

In the clustering case (Fig.6a), a representative of a mined knowledge stands in one or many clusters. A cluster has one or more representative elements and each element consists of fields filled by the user. The number of fields as well as data type of each field, which is also defined by the user, depends on the clustering algorithm used. The meta-data of these fields

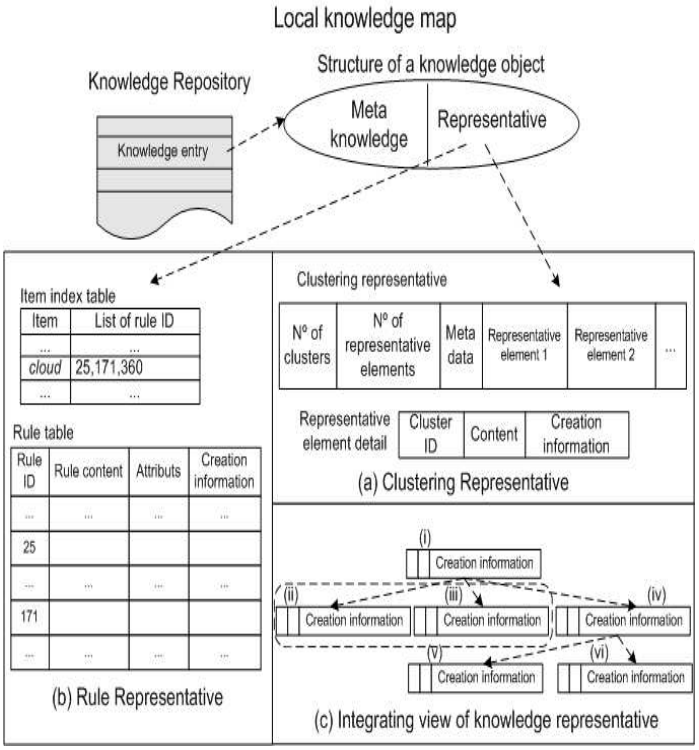


Fig. 6. Organisation of local knowledge map

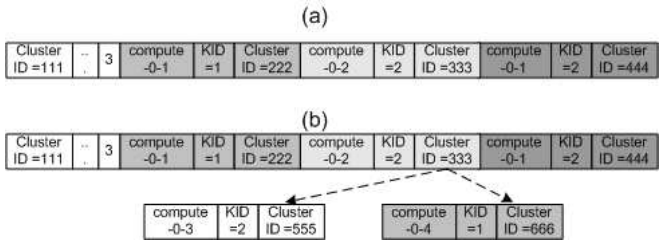


Fig. 7. An example of integration link of creation information: (a) one level; (b) multi-level

```
<representative type="Clustering">
  <numcluster>3</numcluster>
  <numrep>1</numrep>
  <repmetadatafields>Id,Counts,Centers,Variances </repmetadatafields>
  <repmetadatatype>int;long;vector,3,double;matrix,3,double</repmetadatatype>
  <cluster>
    <clusterID>0</clusterID>
    ......
  </cluster>
  <cluster>
    <clusterID>1</clusterID>
      <rep>1;1500;0.005,0.006,0.007;0.0051,0.0061,0.0071,0.0052,0.0062,0.0072,0.0053,0.0063,0.0073</rep>
      <creating type="Integrating">
        <numelement>3</numelement>
        <level0>
          ......
        </level0>
        <level0>
          ......
        </level0>
        <level0>
          <site>compute-0-1</site>
            <knowledgeID>1</knowledgeID>
          <clusterID>2</clusterID>
          <numelement>2</numelement>
          <level1>
            <site>compute-0-0</site>
            <knowledgeID>1</knowledgeID>
            <clusterID>1</clusterID>
            <numelement>0</numelement>
          </level1>
          <level1>
              <site>compute-0-2</site>
              <knowledgeID>0</knowledgeID>
              <clusterID>0</clusterID>
              <numelement>0</numelement>
          </level1>
        </level0>
      </creating>
  </cluster>
  <cluster>
    <clusterID>2</clusterID>
    ...
  </cluster>
</representative>
```

Fig. 8. A representative of Clustering in XML format

is also included in each representative. *KM* allows the user to define this meta-data with both scalar and vector data type. A cluster also contains information about its creation. This information shows how this cluster was created: by clustering or integration process. In the former case, the information is a tuple of (*hostname, cluster filename, cluster identity*) and in the latter, it is a tuple of (*hostname, knowledge identity, cluster identity*). *hostname* is the location where clustering results are stored in files called cluster files with their *cluster filenames*. Each cluster has a *cluster identity* and it is unique in its knowledge entry. For example, a knowledge entry which is created by a variance-based clustering algorithm [20] on test datasets, has its representative in XML format as shown in Fig.8. In this example, there are three clusters, each one has only one representative. A cluster representative consists of three fields: *cluster identity*, *counts*, *centres* and *variances* with their data types which are *integer*, *long*, *vector 3 of doubles* and *matrix 3x3 of doubles* respectively. The content of a cluster representative is presented after its meta-data. Besides, another important information of cluster representative is the creation type which shows how this cluster was created: by either a clustering process or an integration process which merges sub-clusters from different sources to build this cluster. In the integration case, the cluster representative shows its integration link representing all information needed to build this cluster. Fig.6c and Fig.7b show an example of integration link. In this figure, the cluster at the root level is integrated from three other sub-clusters where the last one is also integrated from two others. Note that in Fig.6c, representatives (ii) and (iii) belong to the same knowledge.

In the rule case (Fig.6b), the mined knowledge is represented as a set of the production rules [2]. As mentioned above, a rule is of the form "IF {cause expression} THEN {conclusion expression}" and an expression (cause or conclusion) contains a set of items. A rule also includes its attributes such as *support* and *confidence* [9] in association rules task or *coverage* and *accuracy* [14] in classification task, etc. In order to represent these rules by their items, a representative in our approach consists of two parts: a *rule table* and an *item index table*. The former is a table of rules where each line represents a rule including its identity, content, attributes and creation information. The *item index table* is a data structure that maps items to the rule table. For example, the index of a book maps a set of selected terms to page numbers. There are many different types of index described in literature. In our approach,

the index table is based on *inverted list* [31] technique because it is one of the most efficient index structures [32]. This index table consists of two parts: items and a collection of lists, one list per item, recording the identity of the rule containing that item. For example (Fig.6a), we assume that the term *"cloud"* exists in rules of which identities are *25, 171, 360*, so its list is {*25, 171, 360*}. This index table also expresses the relationship between items and their corresponding rule. By using this table, rules which are related to the given items will be retrieved by the intersection of their lists, e.g. the list of term *"pressure"* is *20, 171* so the identity (ID) of rule that contains *"cloud"* and *"pressure"* is *171*. This ID is then used to retrieve the rule and its attributes. In addition, a rule can be created by using one or more other rules, so its creation information keeps this link (Fig.6c).

E. Knowledge map manager

Knowledge map manager is responsible for managing and coordinating the local knowledge map and the knowledge map core. For *local knowledge map*, this component provides primitives to create, add, delete, update knowledge entries and their related components (e.g. *rule net* and *item index table*) in knowledge repository. It also allows to submit local meta-knowledge to its repository in *knowledge map core*. This component provides also primitives to handle the meta-knowledge in the repository as well as the concept node in the concept tree repository. A key role of this component is to keep the coherence between the *local knowledge map* and the *knowledge map core*.

IV. Knowledge Map operations

In this section, we present basic operations of knowledge map system including adding, retrieving, and maintenance. We firstly create the *concept tree* repository with some predefined *concept trees*. The administrator can then update its content.

A. Adding new knowledge

For any new mined knowledge, its corresponding meta-data and its representative are generated and mapped to a *knowledge object*. This object will be added to the *local knowledge repository* with an appropriate *concept identity*. Its meta-knowledge is then submitted to the *meta-knowledge repository* of *knowledge map core*. The adding operation is realized via the primitive ***"put"***. The Fig.9 shows a flowchart of the adding process.

B. Update/Delete knowledge

KM allows users to update or to delete an existing knowledge meta-data via *"update"* and *"delete"* primitives. These operations are executed at local site and then the system will automatically update *knowledge map core* to ensure the coherence between core and *local knowledge map*. This operation is moreover atomic.

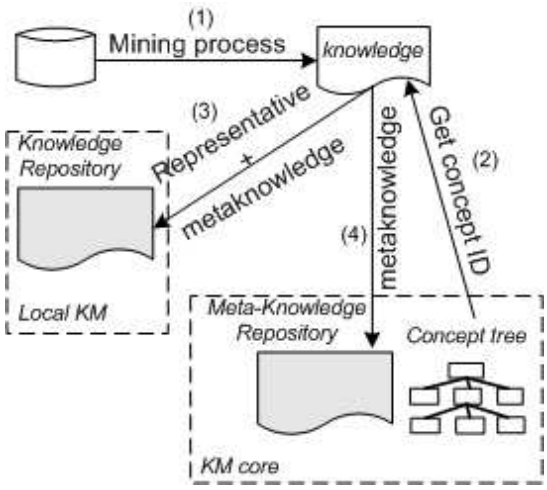


Fig. 9. Adding a new knowledge: (1) knowledge built by a mining process; (2) get an appropriate Concept Identity; (3) knowledge object is added to *local knowledge repository*; (4) Metaknowledge is submitted to *meta-knowledge repository*

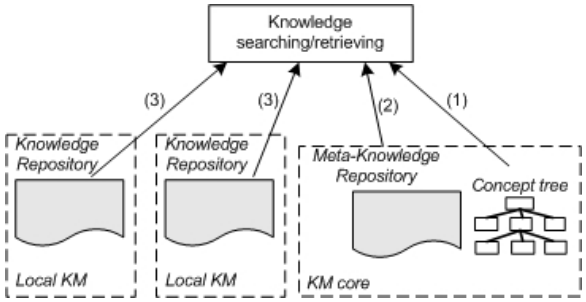


Fig. 10. Retrieving knowledge: (1) Concept (ID or name) retrieving; (2) Meta knowledge retrieving; (3) Knowledge Retrieving from different local KMs

C. Knowledge searching/retrieving

These operations are functions of *find/retrieve* primitives (Fig.10). *KM* supports different levels of search: concepts or meta-data of mined knowledge. At the concept level, *KM* allows the user to search and retrieve concepts acquired through their identity or name. The search operation can be done using different criteria such as concept (e.g. search all *meta knowledge* of a selected concept), mining task and algorithm used to build its knowledge. The retrieve operation is performed through the *knowledge identity* and the *location* of the knowledge needed. This process returns a knowledge object. This operation is executed both locally and globally, i.e. users can retrieve the knowledge needed at its local site or from a group of sites of the system.

V. Implementation and Exploitation

We have implemented a prototype of *KM* and in the current version, the topology of distribution is a flat tree where one local site is elected as the host. The *meta-knowledge repository* of *KM core* is located at this host while its *concept tree* resides in every site. In this case, only the administrator can define and update the content of this *concept tree* at one site and KM system will then update every replicas. The advantages

```
...
universe = java
executable = KM.class
arguments = KM nocom -c put -p KO1.xml
jar_files = KM_Lib.jar
output = KMout.xml
error = KM.error
queue
...
```

Fig. 11. An example of a Condor submit file

and disadvantages of this approach will be discussed in the section VI.

In order to exploit mined knowledge, these knowledge should be managed by *KM* system. If it is not, then the first step is to create knowledge objects including meta-knowledge and representatives, and then add it in each appropriate *local KM*. In the current implementation, a knowledge object has XML format as shown in Fig.8. Their *meta-knowledge* will be automatically submitted to the *meta-knowledge repository* at the *knowledge core map* as an adding operation of a new mined knowledge. Next, users can exploit these meta-knowledge and knowledge object in their integration process or explore the knowledge. In this version, repositories of *KM core* and *Local KM* are also in XML format.

Communication Our aim is to provide an efficient *KM* for distributed environments. Our approach provides a flexible solution so that *KM* can be carried on or interact with different communication system (e.g. RMI [15]) as well as workload management systems on cluster or grid platforms (e.g. Condor [6], PBS or OpenPBS [27]). We present a scenario, as an example, where *KM* system is cooperating with Condor. In this case, each *KM* operation is an independent executable job with its appropriate parameters including input, output files and others. Users write the submit description file including resources needed and then use the Condor system to submit it. An example of a submitted file is shown in Fig.11. In this file, a user adds knowledge objects, which are stored in the file *KO1.xml*, of mined knowledge to a *local KM* at a remote site. This mined knowledge already exists on that site or has just built after a mining process. The output file *KMout.xml* contains the meta-knowledge of knowledge objects added and the user uses this information to submit to *meta-knowledge repository* of *KM core*. In this case, the user is responsible for the coherence between *KM core* and *local KM*. In addition, the parameter *nocom* in the argument line shows that user does not use the communication module of the *KM* system.

An alternative way of exploiting the *KM* system is to use its communication module with different communication middleware. The current *KM* uses Java RMI but it can easily use other communication middleware. In this version, the *KM* runtime includes a set of *KM Daemon* (Fig.12). Each local site has one *KM Daemon* that is responsible for processing local/remote requests. These *KM Daemon* are created at the start by using the primitive ***"init"***. The primitive ***"stop"*** will terminate all the *KM Daemons*. A *KM* application can send request to one or many remote sites. As shown in Fig.12, for

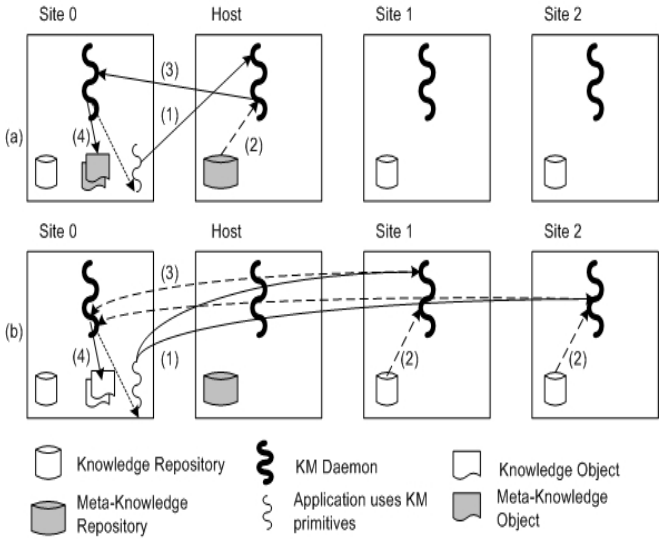


Fig. 12. An example of using Knowledge Map

example, the application find firstly all the meta-knowledge needed via primitive "find" (Fig.12a). This action is composed of four steps: a request is sent to the Host (1) to look for the meta-knowledge needed. Then, this will be retrieved (2) and sent back to the source site (3), and it extracts the results as meta-knowledge objects (4). The application extracts knowledge via primitive "retrieve" (Fig.12b). This action is also composed of four steps: (1) requests are sent to the appropriate sites; (2) retrieve the knowledge found at each site; (3) sent back to the source site via *KM Daemon*; (4) extracts results as knowledge objects.

Another issue of the *KM* implementation is the creation information of representatives in the integration case (c.f. III-D). Normally there are two kinds of links: one level integration and multi-level integration. In the first kind, the creation information of a representative only contains integration information from one sub-level that is its direct children. For example, as shown in Fig.7a, cluster 111 is integrated from three sub-clusters (222, 333, 444) of which information are registered in creation information of cluster 111. Meanwhile, the cluster 333 is also integrated from two other clusters (555 and 666) but this information is not shown in the creation information of the cluster 111. The advantage of this approach is its simplicity and the saving of storage capacity used of creation information. However, a global search in each related local *KM* is needed to retrieve all sub-levels of integration in this case. Our *KM* system is implemented with a multi-level integration of creation information. In this approach, all integration levels of a representative are in its creation information (Fig.7b).

VI. Evaluation and Discussion

We are using this *KM* in our framework [19][20] as shown in the Fig.13. It is difficult to evaluate our approach by comparing it to other systems because it is unique so far. Therefore, we evaluate different aspects of the system architecture for supporting the management, mapping, representing and retrieving the knowledge.

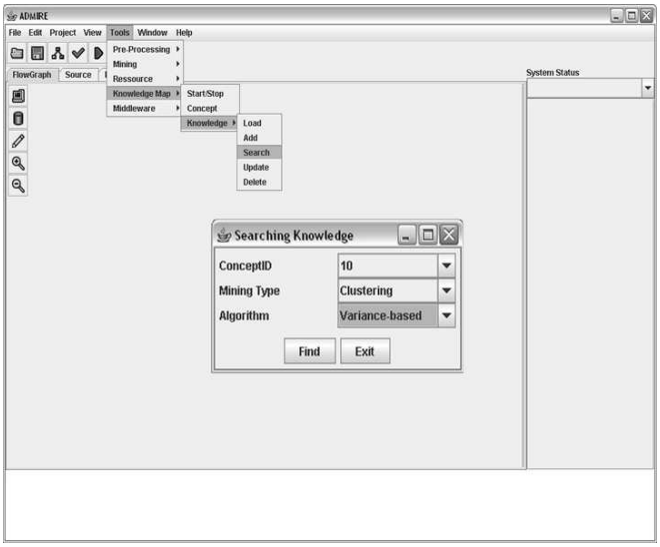


Fig. 13. Using Knowledge Map in the ADMIRE framework

First, we evaluate the complexity of search/retrieve the knowledge object of the system. This operation includes two parts: searching relative concept and search/retrieve the knowledge. Let N be the number of *concept tree* entries and n be the number of *concept nodes* for each *concept tree*. The complexity of the first part is $O(\log N + \log n)$ because the concept tree entries are indexed according to the B+tree model. However, the number of concept entries as well as of concept nodes of a concept tree is smaller compared to the number of knowledge entries. So this complexity depends strongly on the cost of search/retrieve operations. Let M be the number of meta-knowledge entries in the *KM core*, so the complexity of searching a meta-knowledge entry at this level is $O(\log M)$. The complexity of retrieving a knowledge object depends on the number of knowledge entries m in *local KM*. Therefore, this complexity is $O(\log M + C \log m)$, where C is the communication cost.

Next, we discuss the knowledge map architecture. Firstly, the structure of *concept tree* is based on the concept map [24], which is one of the advantages of this model. We can avoid the problem of semantic ambiguity as well as reduce the domain search to improve the speed and accuracy of the results. In our current version, the concept tree is implemented at each site. The advantage of this approach is to reduce the communication cost of searching/retrieving task but the communication cost is high for updating task. However, the frequency of this updating task is very low compared with the frequency of the searching/retrieving tasks. Secondly, the division of knowledge map into two main components (local and core) has some advantages: (i) the core component acts as a summary map of knowledge and it is a representation of knowledge about knowledge when combined with local *KM*; (ii) avoiding the problem of having the whole knowledge on one master site (or server), which is not feasible in very large distributed system such as Grid. By representing meta-knowledge in their relationship links, the goal is to provide an integration view of this knowledge. Furthermore, the use of *rule net* for representing the rules mined creates a map of knowledge elements by representing relationships between items and rules.

Moreover, our *KM* system offers a knowledge map with flexible and dynamic architecture where users can easily update the *concept tree* repository as well as meta-knowledge entries. The current index technique used in a rule representative is an inverted list. However, we can improve it without affecting to whole system structure by using other index algorithms as in [23] or applying compressed technique as discussed in [33]. Moreover, flexible and dynamic features are also reflected by mapping a knowledge to a *knowledge object*. The goal here is to provide a portable approach where knowledge object can be represented by different techniques such as an entity, an XML-based record, or a record of database, etc.

Although the implementation of the creation information of a representative might not be optimal for the storage capacity used, it takes an important advantage in the communication cost compared to one-level approach in retrieving the whole integration links. For example, we analyse these costs for two topologies (Fig.14): flat tree and binary tree. We assume that one representative is integrated by *N* elements and the information size of each element is 32 bytes. In the first topology, information of all elements are stored in its creation information for both cases: one level and multi-level. There is no communication cost and the storage size is $32 \times N$ bytes. In the second topology, the storage size at this representative (root of the tree) is only 32×2 bytes for one-level and $32 \times N$ bytes for multi-level. Furthermore, in one-level case, the storage size at each site (not root) in the tree is always 32×2 bytes except sites at the leaf level. In the multi-level case, this size of a site at the level *h* is $32 \times n_h$ bytes with $n_h = 2(n_{h-1} + 1)$ and $n_0 = 0$. However, there is no communication cost needed for multi-level case, all integration links are in this representative. In order to evaluate this cost for one-level, we assume that the communication is executed in parallel at each level of binary tree with the same latency between two sites and the searching time at each site is negligible compared with communication time. This means that all sites at the same level, each one sends two requests to its two children (one request/child), will receive their replies at the same time. So the communication cost is evaluated by: $2 \times l \log_2 N$ where *l* is the communication latence between two sites or more general is $O(l \log_p N)$ with *p* depends on the chosen topology. In the Grid environment, the communication latency and the number of participating sites are important factors affecting the overall performance of distributed tasks.

VII. Conclusion

In this paper, we presented an architecture of the knowledge map layer. This new approach aims at managing effectively the mined knowledge on large scale distributed platforms. The purpose of this research is to provide a knowledge map to facilitate the management of the results as well as to provide a viable environment for the *DDM* applications. Throughout evaluations of each component and it function, we

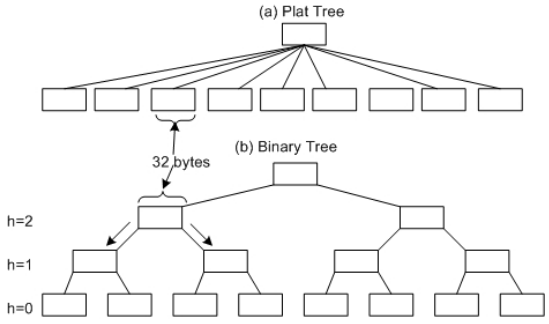


Fig. 14. Different topology of integration

can conclude that knowledge map is an efficient and flexible system in a large and distributed environment. It satisfies the needs for managing, exploring, and retrieving the mined knowledge of *DDM* in large distributed environment.

This knowledge map is integrated in the ADMIRE framework. Experimental results on real-world applications are also being produced [20] and this will allow us to test and evaluate deeply the system robustness and the distributed data mining approaches at very large scale.

REFERENCES

[1] S.S. R. Abidi and Cheah Yu-N, *A Convergence of Knowledge Management and Data Mining: Towards "Knowledge-Driven" Strategic Services*, 3rd International Conference on the Practical Applications of Knowledge Management, Manchester, 2000

[2] B.G. Buchanan and E.H. Shortliffe, *Rule-Based Expert Systems: The MYCIN Experiments of The Standford Heuristic Programming Projects*, Reading, MA: Addison-Wesley, 1984.

[3] T. Buzan and B. Buzan, *The Mind Map Book*, Plume, 1996.

[4] M. Cannataro, D. Talia and P. Trunfio, *Distributed Data Mining on the Grid*, Future Generation Computer Systems, North-Holland vol. 18, no. 8, pp. 1101-1112.

[5] S.M Chen, J-S. Ke and J-F. Chang, *Knowledge Representation Using Fuzzy Petri Nets*, IEEE transaction on Knowledge and Data Engineering vol.2, no.3, 1990, 311-319

[6] http://www.cs.wisc.edu/condor/

[7] T.H Davenport and L. Prusak, *Working Knowledge*, Havard Business School Press, 1997.

[8] Y. Deng, S-K. Chang, *A G-Net model for Knowledge Representation and Reasoning*, IEEE transaction on Knowledge and Data Engineering vol.2, no.3, 1990, 295-310

[9] M.H. Dunham, *Data Mining Introductory and Advanced Topics*, 1st ed. Prentice Hall, 2002.

[10] M. J. Eppler, *Making Knowledge Visible through Intranet Knowledge Maps: Concepts, Elements, Cases* Proceedings of the 34th Hawaii International Conference on System Sciences, 2001.

[11] C. Faloutsos and K. Lin, *FastMap: A Fast Algorithm for Indexing, Data-Ming and Visualization of Traditional and Multimedia Datasets*, Proceedings of SIGMOD'95 Conference, 1995, 163-174.

[12] D. Gasevic, D. Djuric and V. Devedzic, *Model Driven Architecture and Ontology Development*, Springer-Verlag, 2006, 46-57

[13] S.K. Gupta, V. Bhatnagar and S.K. Wasan, *A propasal for Data Mining Management System*, Integrating Data Mining and Knowledge Management Workshop, IEEE ICDM, 2001.

[14] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2nd ed. Morgan Kaufmann Publishers, 2006.

[15] http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp

[16] H. Kargupta and P. Chan, *Advances in distributed and Parallel Knowledge Discovery*, 1st ed. AAAI Press/The MIT Press, London, 2000.

[17] J-C. Silva, C. Giannella, R. Bhargava, H. Kargupta, and M. Klusch, *Distributed Data Mining and Agents* International Journal of Engineering Applications of Artificial Intelligence, 18 (7), Elsevier Science, 2005.

[18] N-A. Le-Khac, M.T. Kechadi, J. Carthy *ADMIRE framework: Distributed data mining on data grid platforms*, Proceedings of 1st International Conference on Software and Data Technologies ICSOFT'06, 2006, 67-72

[19] N-A. Le-Khac, Lamine M. Aouad and M.T. Kechadi *Knowledge Map: Toward a new approach supporting the knowledge management in Distributed Data Mining*, KUI Workshop, IEEE International Conference on Autonomic and Autonomous Systems ICAS'07, Athens, Greece, 2007.

[20] Lamine M. Aouad, N-A. Le-Khac and M.T. Kechadi *Variance-based Clustering Technique for Distributed Data Mining Applications*, International Conference on Data Mining (DMIN'07), USA, 2007

[21] N-A. Le-Khac, Lamine M. Aouad and M.T. Kechadi *An Efficient Support Management Tool for Distributed Data Mining Environments*, 2nd IEEE/ACM International Conference on Digital Information Management (ICDIM'07), October 28-31, Lyon, France, 2007.

[22] F. Lin and C.M. Hsueh, *Knowledge map creation and maintenance for virtual communities of practice*, Journal of Information Processing and Management, vol. 42, 2006, 551-568.

[23] M. Martynov and B. Novikov, *An Indexing Algorithm for Text Retrieval*, Proceedings of the International Workshop on Advances in Databases and Information system (ADBIS'96), Moscow, 1996, 171-175.

[24] J.D. Novak and D.B. Gowin, *Learning how to learn*, Cambridge University Press, 1984.

[25] M.C.F. Oliveira and H. Levkowitz, *From Visual Data Exploration to Visual Data Mining: A survey*, IEEE transaction on visualization and computer graphics vol.9, no.3, 2003, 378-394

[26] J-L. Peterson, *"Petri Nets"*, Journal of ACM Computing Surveys vol. 9 no.3, 1977, 223252.

[27] http://www.openpbs.org/

[28] P-N. Tan, M. Steinbach and V. Kumar, *Introduction to Data Mining*, 1st ed. Pearson Education, 2006.

[29] A. Veloso, B. Possas, W. Meira and M. B. Carvalho, *Knowledge Management in Association Rule Mining*, Integrating Data Mining and Knowledge Management Workshop, IEEE ICDM, 2001

[30] M.N. Wexler, *The who, what and why of knowledge mapping*, Journal of Knowledge Management, vol. 5, 2001, 249-263.

[31] J. Zobel, A. Moffat, R. Sacks-Davis *An efficient indexing technique for full-text database systems*, Proceeding of the 18th VLDB Conference Vancouver, British Columbia, Canada, 1992, 352-362.

[32] J. Zobel, A. Moffat, R. Sacks-Davis *Searching Large Lexicons for Partially Specified Terms using Compressed Inverted Files*, Proceeding of the 19th VLDB Conference Dublin, Ireland, 1993, 290-301.

[33] J. Zobel and A. Moffat *Inverted Files for Text Search Engines*, Journal of ACM Computing Surveys, Vol. 38, No.2, Article 6, 2006

Dr. Nhien An Le Khac is a Postdoctoral Fellow at the School of Computer Science and Informatics, University College Dublin, Ireland. He obtained his M.Sc. in Computer Science in 2000 at National University of Vietnam(VNU-HCM), and Ph.D. in Computer Science in 2005 at the INP (Institut National Polytechnique) Grenoble, France. His research interest spans the area of Parallel Computing, Grid computing, heterogeneous clusters and Distributed Data Mining. He is a member of the IEEE computer society.

Dr. Lamine M. Aouad has joined the University College Dublin, Ireland, in 2006 as a postdoctoral research fellow. He received his PhD from the University of Lille I, and his MSc (DEA) in distributed computing from the University of Paris XI, France. His research interests lie in the field of parallel and distributed paradigms and algorithms and application design for grid systems, specially for large scale linear algebra applications and the data mining area.

Dr. M-Tahar Kechadi was awarded PhD and a DEA (Diplome d'Etude Approfondie) - Masters degree - in Computer Science from University of Lille 1, France. He joined UCD School of Computer Science & Informatics in 1999. His research interests span the areas of optimization techniques, Data mining, heterogeneous distributed systems and Grid computing. He is a member of the communication of the ACM journal and IEEE computer society.

ISAST Transactions on Intelligent Systems, No. 1, Vol. 1, 2008
V. Bhatnagar et al.: Gradation Framework for Anomaly Detection in Streams

Gradation Framework for Anomaly Detection in Streams

Vasudha Bhatnagar, Sharanjit Kaur and Ajit Chaturvedi

***Abstract*—Anomaly detection is an application centric task, which is heavily dependent on user interaction for its usefulness. Though the extent of "outlying-ness" of a data point is computed objectively, its semantic interpretation is highly context dependent, making the task somewhat subjective with respect to the user and application.**

In this paper we propose a framework for grading data regions as patterns or anomalies in a stream within defined data space. Such streams commonly arise in scientific applications. We distinguish an anomaly as either a noisy data region or an outlier with quantified characteristics. Signature of anomalous regions in data space are delivered to facilitate better understanding of the underlying process that generated the data. The gradation framework is useful in not only identifying unexpected events, but also finds application in noise removal/reduction, leading to improved quality of discovered patterns. We report experiments that demonstrate the efficacy of the framework and compare the performance with some recent algorithms with similar functionality.

***Index Terms*—Data stream, Data gradation, Noise, Outlier, Grid**

I. Introduction

In several commercial and scientific applications, rare or unexpected events are more interesting than the common ones, as they often lead to more meaningful discoveries [14]. Example applications in commercial domain include credit card fraud detection, criminal activities in e-commerce, video surveillance etc.. In scientific domain, unknown astronomical objects, unusual astronomical or seismic phenomena, unexpected values of vital parameters in patient monitoring etc. manifest as *anomalies* in observed data. Thus anomaly detection is an important task in the process of knowledge discovery. Further, since most data mining algorithms lack robustness, presence of anomalies results in deterioration of the quality of discovered patterns. Consequently the task of anomaly reduction/removal in the KDD process becomes very important for the overall success of the KDD endeavor [11], [23].

Most clustering algorithm either selectively report clusters based on user defined parameter set and leave out the remaining data as anomalies [11], or accommodate anomalies in clusters deteriorating the quality of clustering [13]. Both actions are undesirable in several scientific applications where the data space is bounded. Often, in such applications it is important to detect and report any unusual behavior as early as possible.

An ideal clustering algorithm is expected to view the data space D as consisting of three components:

$$D = P + N + O \tag{1}$$

where P is the set of patterns, N is the noise and O is the set of outliers. Even though both *noise* and *outliers* are considered as *anomalies*, former is an anomaly exhibited by a set of data values, while the latter is an unexpected data value which indicates an aberration. Faulty data capturing mechanisms and inadequate data representations are some of the common causes of noise. *Noise* must be removed or considered separately to improve accuracy of the discovered model [13]. *Outliers* provide additional insight in the data generating process and often need immediate corrective action or investigation.

A. Motivation for Gradation

In several scientific applications including patient monitoring systems, seismic observations, process monitoring systems, astronomical observations etc., the observed data is expected to fall with in defined normative bounds. Observations with in the normative bounds need to be mined for patterns, while those which are outside are clear anomalies. However, some values that lie with in the bounds may defy expectations and hence are considered anomalous. Such data also need to be investigated, explained and understood.

Detection of anomalies in a data stream with bounded data space is challenging because of the dynamic nature of the data stream. In extreme situations this may influence the bounds itself. Anomaly detection in such environments requires partitioning of the stream into three constituents as mentioned above while taking into account the recency of data. User subjectivity involved in categorizing data (within the bounded space) as patterns or anomalies increases the challenge multi fold.

We propose a framework to grade the streaming data into above mentioned components, quantify the characteristics of each component and report to the user for interpretation. We consider this as a desirable functionality for stream monitoring applications since automated mechanisms can only highlight potential anomalies, confirmation of which requires further investigation and correlation by human experts in the domain. Since gradation facilitates segregation of data space into

Manuscript received December 8, 2007; accepted January 24, 2008.

V. Bhatnagar is with the Department of Computer Science, University of Delhi, Delhi, India. e.mail: vbhatnagar@cs.du.ac.in

S. Kaur is with the Department of Computer Science, University of Delhi, Delhi, India. (Corresponding author to provide Phone: 91-11-27667591; Fax: 91-11-27662553; e-mail: skaur@cs.du.ac.in)

A. Chaturvedi is with the Department of Statistics, University of Delhi, Delhi, India. e.mail: achaturvedi@stats.du.ac.in

regions with different characteristics, it provides opportunity to focus on the component of immediate interest. In data stream with noise, gradation can also be used to *de-noise* the stream in order to discover better quality patterns.

B. Our Contribution

The objective of this work is to understand different types of anomalies that can arise in a bounded data space and propose a framework for *explicit* gradation (Section II). A grid based data driven algorithm has been designed to implement the proposal (Section III). The salient features of the proposed algorithm are listed below :

1) On-the-fly detection of distinctly outlying data points in constant time. This permits immediate action in case of *serious* outliers.
2) Distinction between two types of anomalous regions in data space, viz. outlying noisy regions and neighborhood noisy regions.
3) Reporting signature of anomalous regions in data space. This helps in understanding the nature of anomaly thereby augmenting the knowledge of the data generating process.

Experiments on synthetic datasets demonstrate the functionality of the scheme, while those on public dataset prove its efficacy and scalability.

C. Related Works

Some of the recent algorithms for outlier detection in static dataset appear in [2], [14], [18], [20]. Recently, algorithms have been proposed for outlier detection in streaming environment [1], [5], [9], [19]. Our work particularly draws strength from the algorithms that aim to find clusters of various shapes, sizes and densities [4], [11], [15], [24].

Earlier works for detecting outliers in non-stationary data include [1], [5], [9]. The algorithm proposed in [9] incrementally builds a probabilistic model of incoming points by using an on-line discounting learning method, and computes the extent of deviation for each point from learned model. Algorithm LOADED [1] handles dynamic data with mixed attributes by combining a lattice data structure for categorical attributes with correlation statistics maintained for continuous attributes. Efficiency of LOADED decreases with increase in categorical attributes and number of attributes. Incremental Local Outlying Factor algorithm [5] employs an indexing structure to support k-nearest neighbors and reverse k-nearest neighbors queries. This algorithm heavily depends upon indexing structure and hence does not work well for high dimensional data.

CLAD (Clustering for Anomaly Detection) [15] uses a distance based approach which determines suspicious clusters at both local and global level, and handles noise. It dynamically computes cluster-width using the smallest distance between pairs of data points drawn randomly from the dataset, instead of taking input from the user. This strategy works when entire dataset is available, hence cannot be used with streams.

CDDD (Clusters Derived with Dyadic Decomposition) algorithm [4] delivers multi-scale clusters by decomposing geospatial feature space into hierarchical set of cubes. The algorithm bisects one or more dimensions recursively and then on the basis of some user defined input parameters distinguishes between cluster, noise and outlier. Approach used in CDDD works efficiently for static datasets having structure at different scales. Scheme presented in [11] finds clusters of different densities, shapes and sizes and is shown to be extremely effective for NASA earth science data. However, it works on a static dataset and does not handle outliers.

ExCC [24] does exclusive and complete clustering for data stream. It reports clusters of different densities with their signature.

II. Framework for Gradation

Consider a d-dimensional dataset D in $\mathcal{R}^n$, with dimension set $A = \{A_1, \ldots, A_d\}$. Let l_i and h_i be the two known boundary data values along dimension A_i and let $r_i = (h_i - l_i)$ represents the range of A_i. This leads to a d-dimensional real data space $S = [l_1, h_1] \times \ldots \times [l_d, h_d]$, which defines the bounds of normalcy. Examples where S can be crisply defined as above include critical limits of vital parameters in patient monitoring, concentration of gases, pressures and temperature captured by sensors in coal mines, movement and pressure inputs of seismic sensors etc..

As per Eq.1, we aim to segregate regions corresponding to P, N and O in S. *Noise* is an anomalous region in data space with more than one data point and is identified by a signature, while a single anomalous data point is referred as an *outlier*. We make the following observations before proceeding further.

i) A data point outside S is clearly unexpected and illegitimate as per the domain knowledge and is an outlier. Deviation of its data values outside the normal bounds, along one or more dimensions characterizes and quantifies its severity. Depending on the severity of an outlier we grade it either as $\mathcal{M}$-outlier or $\mathcal{O}$-outlier (M and O respectively in Figure 2). While $\mathcal{M}$-outlier is more severe and needs no further characterization, the severity of an $\mathcal{O}$-outlier is quantized.

ii) A set of data points well inside S could connote *noise* either because of its isolated location in the data space or unexpectedly low density. Depending on its characteristics, a noisy region is graded as either *outlying* (O-noise) or *neighborhood* noise (N-noise) (OR, NR respectively in Figure 2). A noisy outlying region with a single point is notable and is designated as a *succinct* outlier ($\mathcal{S}$-outlier, S in Figure 2).

iii) Data points which are high on similarity form the patterns - clusters of varied shapes, sizes and densities (C in Figure 2).

Consequent to this discussion we revise Eq. 1 to Eq. 2, and our goal is revised to grade data into six distinct components (Figure 1).

$$D = P + \{OR + NR\} + \{\mathcal{M} + \mathcal{O} + \mathcal{S}\} \qquad (2)$$

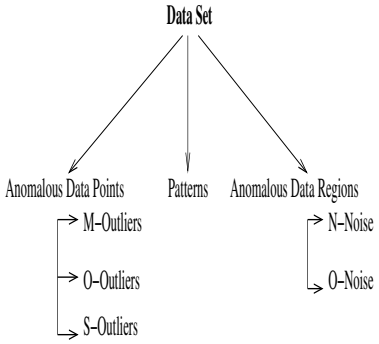


Fig. 1. Scheme for gradation of anomalies

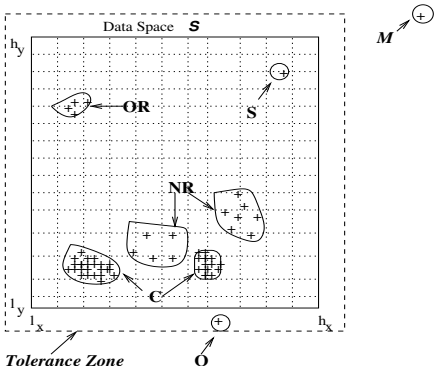


Fig. 2. Grading in two dimensional data space S

A. Detecting Anomalous Data Points: Outliers

A point $p(v_1, \ldots, v_d) \notin S$ can safely be considered as anomalous and hence an outlier. It is characterized by assessing its distance outside the boundary of S, which indicates its *extent of outlying*. Let δ_i be the normalized distance of p outside the boundary of S along dimension A_i.

$$\delta_i = \begin{cases} 0 & if \quad l_i \leq v_i \leq h_i \\ \frac{1}{r_i} * min(|v_i - l_i|, |v_i - h_i|) & otherwise \end{cases} \tag{3}$$

Data point p is an outlier if there exists a dimension A_i $(i = 1, d)$, such that $\delta_i > 0$. A_i is called the outlying dimension and OD designates the set of all outlying dimensions for p. This mechanism of detecting outliers is trivial and may deliver large number of false positives if the bounds are not defined accurately (e.g. in case the user is inexperienced).

To make the scheme robust, a tolerance vector $\Gamma = \{\tau_1, \ldots, \tau_d\}$ of size d is defined. Here $\tau_i = [l'_i, h'_i] \supseteq [l_i, h_i]$ indicates the revised range including the user tolerance for deviation outside the bounds for dimension A_i. Intuitively Γ connotes a tolerance zone (Figure 2), and a data point that falls in this zone is quantified for its *outlyingness*. The tolerance vector Γ is periodically computed from the current distribution of the data in the stream (Section III-D) and aids in making a decision whether a data value along a dimension is *severely* or *tolerably* outlying.

To compute Γ, the range $r_i = [l_i, h_i]$ of dimension A_i is divided into k equi-width intervals $(I_{i,1}, \ldots, I_{i,k})$, such that the t^{th} interval $I_{i,t} = [l_{i,t}, h_{i,t})$ and $l_{i,1} = l_i,\ h_{i,k} = h_i$. Here k is a user defined parameter. Let w_i be the width of the interval of i^{th} dimension. $[l'_i, h'_i]$ is determined using one of the three approaches depending on the data distribution.

1) Modal Heuristics: Each dimension is categorized as spike, uniform or asymmetric on the basis of point distribution in the intervals. The dimension is said to be spike if majority of the points lie in single interval. For each spike dimension A_i, τ_i is computed based on the location of the modal interval i.e. the interval having the maximum number of points. Let m_i $(1 \leq m_i \leq k)$ be the location of the modal interval for A_i.

$$[l'_i, h'_i] = \begin{cases} [l_i, h_i] \quad if \ \ m_i = k/2 \ or \ m_i = (k+1)/2 \\ [l_i - \frac{(k-m_i)w_i}{k},\ h_i] \ \ if \ \ m_i < k/2 \\ [l_i,\ h_i + \frac{(m_i+1)w_i}{k}] \ \ if \ \ m_i > k/2 \end{cases} \tag{4}$$

If m_i is the middle interval, then it is less likely that points in near future will lie outside the known bounds and hence no deviation outside the bounds is tolerated. In all other cases, tolerance is computed relative to the location of m_i. Closer the m_i towards an extreme, higher is the tolerance for deviation on that side. In case data is nearly uniformly distributed over all the intervals (uniform dimension) then $[l'_i, h'_i] = [l_i - w_i, h_i + w_i]$

2) Estimating Tolerance using Order Statistics: In case the dimension is neither uniform nor spike, a more rigorous approach using order statistics may be employed to estimate the tolerance limit/revised range.

Suppose n points arrive in a stream during a prespecified period. Let $x_{(1)}$ and $x_{(n)}$ be the minimum and maximum values (order statistic of order 1 and n respectively) along dimension A_i such that $l_i \leq x_{(1)} \leq x_{(n)} \leq h_i$. Let v_i be the value of the newly received point. For two specified values of γ and θ $(\in [0, 1])$, probability that $\theta * 100\%$ points lie within $(x_{(1)}, x_{(n)})$ is γ is given by:

$$P(P(x_{(1)} \leq v_i \leq x_{(n)}) \geq \theta) = \gamma \tag{5}$$

We rewrite Eq. 5 using the distribution function $F(x)$ of the variable under study as follows.

$$P[F(x_{(n)}) - F(x_{(1)}) \geq \theta)] = \gamma \tag{6}$$

Depending on the semantics of A_i, θ and γ can be specified by the domain expert. In an ideal situation $\gamma = \theta = 1$ meaning that 100% points fall within $(x_{(1)}, x_{(n)})$ with probablity one. However, in practical real life situations $0 < \gamma, \theta < 1$. We need to estimate allowable deviation ϵ using the current distrbution of A_i for given γ and θ such that:

$$P(P(x_{(1)} - \epsilon \leq v_i \leq x_{(n)} + \epsilon) \geq \theta) = \gamma \tag{7}$$

OR

$$P[F(x_{(n)} + \epsilon) - F(x_{(1)} - \epsilon) \geq \theta)] = \gamma \tag{8}$$

This is a problem of tolerance interval estimation in a distribution free situation and is solved using the distribution of the difference between the largest and the smallest order statistics. Using the fact that the distribution of distribution function is uniform [21], we can write Eq. 8 as

$$P[U_{(n)} - U_{(1)} \geq \theta)] = \gamma \tag{9}$$

where $U_{(n)}$ and $U_{(1)}$ are, respectively, order statistics of order n and 1 from $U(\epsilon, 1+\epsilon)$ and $U(-\epsilon, 1-\epsilon)$ distributions. Thus from Eq. 9, we get

$$n \int_{y=\theta}^{1+\epsilon} \int_{x=-\epsilon}^{y-\theta} (y - x + 2\epsilon)^{n-2} dx \; dy = \gamma \tag{10}$$

We make the transformations $U = U_{(n)} - U_{(1)}$ and $V = U_{(n)}$. The Jacobian of transformation is unity. The joint distribution of U and V is

$$f_{U,V}(\upsilon, \nu) = \begin{cases} n(n-1)\upsilon^{(n-2)}; \; -1 + 2\epsilon < \upsilon < \nu < 1 + 2\epsilon \\ 0; \quad otherwise \end{cases} \tag{11}$$

From Eq. 11, the marginal distribution of U is given by

$$f_U(\upsilon) = \begin{cases} n(n-1)\upsilon^{(n-2)}(1 + 2\epsilon - \upsilon); \; 0 \leq \upsilon \leq 1 + \epsilon \\ 0; \quad otherwise \end{cases} \tag{12}$$

From Eq. 9 and Eq. 12, we have

$$n(n-1) \int_{\theta}^{1+\epsilon} \upsilon^{n-2}(1 + 2\epsilon - \upsilon) d\upsilon = \gamma \tag{13}$$

Solving this integral, we get

$$n[(1+\epsilon)^{n-1} - \theta^{n-1}](1 + 2\epsilon) - (n-1)[(1+\epsilon)^n - \theta^n] = \gamma \tag{14}$$

Given n, θ and γ, we can solve Eq. 14 for ϵ, which is used to define the revised range $[l'_i, h'_i] = [l_i - \epsilon, h_i + \epsilon]$. This computation requires constant time and is done periodically for dimension A_i.

3) User-centric approach: In case the user is knowledgeable and confident about the tolerance limits for each dimension and wants to completely control the anomaly detection mechanism, Γ can be administered by him. This flexibility is extremely useful in applications where the user can judge the implications of deviations on different dimensions. For example, while monitoring vital parameters of a critical patient, Γ can be set according to the criticality of the parameters. This acceptable deviation in blood pressure of a critical patient may be very different from that of blood sugar level.

B. Outlier Detection

Once Γ is in place, $\mathcal{M}$-outlier and $\mathcal{O}$-outlier are identified as per the definitions given below.

Definition 2.1: $\mathcal{M}$-outlier: An anomalous data point $p(v_1, \ldots, v_d)$ with OD as the set of outlying dimensions is a $\mathcal{M}$-outlier if $\exists\, v_i$ such that $v_i \notin [l'_i, h'_i]$ and $1 \leq i \leq d$.

Definition 2.2: $\mathcal{O}$-outlier: Consider a data point $p(v_1, \ldots, v_d)$. If $\exists\, v_i$ s.t. $v_i \in [l'_i, h'_i]$ and $v_i \notin [l_i, h_i]$, then p is categorized as an $\mathcal{O}$-outlier with degree $\mathcal{O}(p) = (|OD|, \frac{1}{d}\sum_i^d \delta_i)$

An $\mathcal{M}$-outlier is a severe anomaly in data and needs no characterization. However, an $\mathcal{O}$-outlier is characterized by i) number of outlying dimensions and ii) the extent of deviation outside the tolerance zone. The knowledge of these two characteristics of an $\mathcal{O}$-outlier helps the user to assess the severity and hence interestingness of the outlier.

Since detection of $\mathcal{M}$-outlier and $\mathcal{O}$-outlier needs simple inspection, they can be identified on-the-fly in $O(d)$ time. Detection of $\mathcal{S}$-outliers requires processing and is discussed in Section III-C.

C. Identifying Patterns and Anomalies

Noise in a dataset is loosely defined as data points that are either not contributing to any pattern and/or are undesirable. Presence of $Noise$ can often distort pattern discovery and hence deteriorate quality of decision support.

In order to distinguish patterns and anomalies, we abstract the data space as consisting of regions of d-dimensional equi-sized cells (Figure 2). A data point $p(v_1, \ldots, v_d)$ can be localized in a cell depending on its data values. Adjacent non-empty cells are joined to form two types of regions viz. Clusters or Patterns (P), and Anomalous Regions (NR, OR). This differentiation is made on the basis of number of points (weight) in the cell. The weight threshold is determined from the current data distribution in the stream whenever gradation is demanded. Gradation of anomalous regions into *O*-noise, *N*-noise and $\mathcal{S}$-outlier is based on inter-region distances as discussed in Section III-B.

Let n_σ and n_μ denote the standard deviation and the average number of points in a cell respectively. Assuming a Gaussian distribution for the number of points in cells, a cell with at least $\rho = abs(n_\mu \pm x * n_\sigma)$ number of data points is heavy cell. Otherwise it is termed as light cell. The heavy cells form patterns whereas light cells join to form noisy regions. Depending on the desired quality of gradation, x can be set to $1, 2, 3$. Choosing $+$ sign instead of the $-$ sign will will lead to patterns with better quality. An experiment to demonstrate this effect is reported in Section IV-E.

III. GRADS Algorithm

The $GrADS$ algorithm is completely data driven and uses grid data structure to maintain the details of the distribution of data values along all dimensions. It consists of an on-line component which processes the incoming stream and detects $\mathcal{M}$ and $\mathcal{O}$ outliers on-the-fly. The offline component is invoked when gradation is demanded by the user to detect the anomalous regions in the stream.

A. Grid Structure

Grid based approaches process a data point in constant time and have been used extensively for clustering solutions [8], [16], [25].

As mentioned in Section II-A, each dimension is divided into k equi-width intervals where k determines the granularity of the grid. Finer granularity of grid results into more and smaller sized regions and allows multi-scale analysis with consequences on memory requirement [4].

A data point is stored in a grid using d internal nodes at d levels, corresponding to each of the d dimensions. A leaf node (cell C^m) in the grid corresponds to the region in the data space determined by the intersection of one interval from each dimension i.e $C^m = (I^m_{1,q1} \wedge I^m_{2,q2} \wedge \ldots \wedge I^m_{d,qd})$ where $I^m_{i,qj}$ refers to qj^{th} interval on i^{th} dimension of cell C^m and $1 \leq qj \leq k$. Though the actual memory requirement of the grid will depend on the data distribution, memory bound $O(k^d)$ may be used as a broad guideline to decide the granularity of the grid [24].

Once the grid is constructed, adjacent cells are merged to identify regions in S. Two cells C^m and C^n are merged iff for each dimension A_i, condition $|I^m_{i,qi} - I^n_{i,qi}| \leq 1$ is satisfied. Regions containing heavy cells correspond to clusters (Section II-C), while the remaining regions need to be further graded as either outlying or neighborhood noise.

B. Gradation of Anomalous Regions

A noisy region is graded as either *O*-noise or *N*-noise depending on its location in the data space relative to discovered clusters. Noisy regions close to clusters are designated as *N*-noise and those which are located far-off constitute *O*-noise. Determining the appropriate distance threshold (Δ), which is the basis of gradation, is crucial to the efficacy of the scheme.

The distance between two regions R^1 and R^2 is computed as distance between the two region centers. Let $I^j_{i,l}$ and $I^j_{i,h}$ respectively denote least (l) and highest (h) interval on i^{th} dimension for region R^j. For R^j, region center is a cell $RC^j < rc^j_1 \ldots rc^j_d >$, where $rc^j_i = \lceil I^j_{i,l} + I^j_{i,h} \rceil / 2$. RC^j denotes the logical center of R^j. Though RC^j may not exist inside the region because of the way adjacency is computed, it does not affect the distance computation. Manhattan distance between the region centers is used to determine the distance threshold.

Let PR denotes set of all patterns regions/clusters discovered in the data space. Let $MaxD(P^i)$ denote the distance of a cluster P^i from the farthest cluster in PR. Then $\Delta = \frac{1}{|PR|} \sum_{\forall P^i \in PR} MaxD(P^i)$, denotes the average distance between a cluster and its farthest neighbor. This value is used as threshold for distinguishing between *N*-noise and *O*-noise.

In order to grade a noisy region N^i, its distance from the closest cluster ($MD(N^i)$) is computed. If this distance is more than Δ, then it is graded as *O*-noise otherwise it is considered *N*-noise. Signatures of the regions are computed from the cells that constitute the boundary of the regions.

C. $\mathcal{S}$-outlier

An outlying noisy region that has only one point is said to have a *succinct* outlier ($\mathcal{S}$-$outlier$). An $\mathcal{S}$-$outlier$ lies within the bound of normalcy and yet is the lone point with the nearest neighborhood at least Δ units away. A *succinct* outlier is delivered as a data point by the algorithm after gradation.

D. Pruning of Grid

Unbounded nature of data stream and discounting the effect of older data are two prime motivations for pruning the grid. The grid pruning is done immediately after gradation, resulting into removal of cells constituting the anomalous regions and outdated patterns. Pruning not only keeps the grid size under control, but also increases the efficacy of the scheme by ignoring the older data while computing discriminating thresholds ρ and Δ (Sections II-C, III-B). This ensures that the anomalous regions in stream do not interfere with data gradation at later point in time and old patterns do not overshadow the clustering scheme. We accomplish this in two stages.

Firstly we prune away the noisy regions, taking care that a possibly evolving cluster is preserved. This de-noises the stream while ensuring that correct clustering scheme is revealed at a later time. The following volume based methodology is used for this purpose.

Let R^j be a region with n data points and v^j_{il} and v^j_{ih} be the minimum and maximum data values along i^{th} dimension. Then the volume of R^j $(vol(R^j)) = \prod^d_{i=1} v^j_{ih} - v^j_{il}$ $where$ $v^j_{il} \neq v^j_{ih}$

The density of the region is computed as $\varphi = \frac{n}{vol(R^j)}$. Noisy regions with density greater than the minimum density of a pattern in PR, are retained in the grid in the anticipation of evolving patterns. Similar functionality is supported by the algorithms proposed in [3], [6], [24], though in a different context.

After pruning of the noisy regions, the remaining cells are further examined for their age. Age based pruning of cells ensures that the historical patterns do not interfere with computation of ρ. We use the following pruning criterion, which removes cells that have not received data points since long [24].

$$\frac{t_{curr} - t^C_l}{NumP} \geq aat^C \tag{15}$$

where t_{curr} is the current time, t^C_l is the time when the last point was received in the cell, aat^C is the average arrival rate of data points in the cell and $NumP$ is the number of points received in the data stream since last gradation. The offline component of $GrADS$ algorithm is given in Figure 3.

Input: All cells in the grid G
Output: Gradation scheme GS = {P, $\mathcal{S}$, O-$noise$, N-$noise$}
Process:
1. Compute Γ as given in Section II-A.
2. Compute weight threshold ρ and mark cells in G as heavy cells or light cells.
3. Generate Pattern Regions P using heavy cells and Noisy Regions N using light cells.
4. Compute $\Delta = \frac{1}{|P|} \sum_{\forall P^i \in P} MaxD(P^i)$
5. For each region N^i in N
6. Compute its distance $MD(N^i)$ from the nearest pattern.
7. If $MD(N^i) \geq \Delta$)
8. If (n == 1) Report $\mathcal{S}$-$outlier$
9. Else grade N^i as O-$noise$ and report its signature.
10. Endif
11. Else grade N^i as N-$noise$ and report its signature.
12. Endif
13. Endfor
EndProcess

Fig. 3. The offline component of $GrADS$ Algorithm

IV. Experimental Study

In this section we present experiments to demonstrate functionality and scalability of $GrADS$ algorithm on two synthetic datasets and a real dataset. All experiments are performed on Intel Centrino processor and 256 MB RAM, running stand alone Linux (kernel 2.4.22-1). The algorithm is implemented

in ANSI C with no optimizations, and compiled using g++ compiler (3.3.2-2).

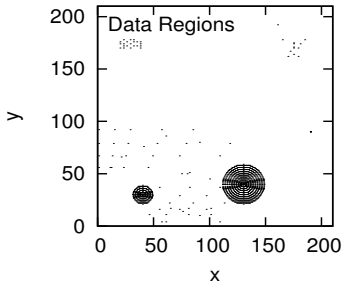
A. Demonstration of functionality

We executed $GrADS$ algorithm on two and three dimensional datasets and found that the outputs matches the corresponding inputs for different grid granularity. Both datasets consisted of clusters and three types of anomalies viz. succinct outlier ($\mathcal{S}$), *O*-noise (OR) or *N*-noise (NR). The records were labeled as per the type of anomaly to enable the evaluation of the quality of graded regions by computing cluster/region purity [3].

The two dimensional dataset, shown in Figure 4, consisted of two clusters (covering more than 90% of data) and 4 noisy regions viz. succinct outlier ($\mathcal{S}$), two outlying noisy regions (OR) and neighbourhood noisy region (NR). Table I shows that four noisy regions are always revealed and match with noisy regions shown in Figure 4.

Table I reveals that the purity of the noisy regions increases with grid granularity. The computed distance of neighborhood noisy region (MD) from its nearest cluster is less than that of outlying noisy regions, as proposed in the framework.

The three dimensional synthetic dataset (Figure 5) has two clusters (covering 97% of data) and four noisy regions viz. two $\mathcal{S}$, one OR and one NR. Table II shows that four anomalies are always detected and match with those shown in Figure 5.



Region	Data Points	Centroid	Region Type
1	350	(40,28)	Cluster
2	1100	(130,40)	Cluster
3	53	-	N-Noise
4	12	-	O-Noise
5	21	-	O-Noise
6	1	(190,90)	S-Outlier

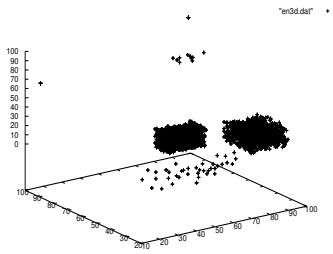
Fig. 4. GnuPlot of two dimensional synthetic dataset and its description

Region purity of three dimensional synthetic data increases with increase in grid granularity (Table II).

B. De-noising and Quality of Patterns

We performed experiments to examine the effectiveness of the algorithm in de-noising the static and stream datasets.

We executed WEKA K-Means on the 2D static dataset (Figure 4) with K=6 and noted the clusters descriptions in terms of centroids (Table III). The value of K = 6 was chosen to match the total regions in the dataset. The small cluster



Region	Data Points	Centroid	Region Type
1	1000	(10,10,9)	Cluster
2	1000	(20,50,45)	Cluster
3	40	-	N-Noise
4	10	-	O-Noise
5	1	(98,99,98)	S-Outlier
6	1	(16,97,65)	S-Outlier

Fig. 5. GnuPlot of three dimensional synthetic dataset and its description

towards the origin was accurately reported with nearly 90% cluster purity. However the bigger cluster was split because of noise and points were distributed into four clusters. With K=2 on the same dataset, the centroid of the bigger cluster was found to be reasonably close to the actual centroid, while that of the smaller cluster was distorted. Subsequently, we removed the noise as per signature reported by the algorithm (k=10, Table I) and clustered again with K=2. Results show that centroids of both clusters are closer to the original clusters (Table III).

The next experiment was performed using the real-life dataset, Intrusion Detection dataset (KDDcup 99 training dataset) [10]. This dataset has 494021 records, each having 42 attributes (34 continuous and 8 categorical). Each record corresponds to either normal class or an attack class. We performed experiment with 23 classes using 34 continuous attributes. We simulated KDDcup data as a stream (1000 points per time unit) and executed the algorithm twice on the same dataset - with and without de-noising. Cluster purity was computed in both cases and a distinct improvement in the quality of clusters was found after noise removal using $GrADS$ algorithm. The result of the experiment is shown in Figure 6. Cluster purity is same in first window for both runs, as expected. But in all subsequent windows, there is a distinct improvement in cluster purity after pruning the noisy regions.

C. Scalability

Since we are not aware of any algorithm which provides functionality similar to $GrADS$ algorithm, we compare its execution time with two recent outlier detection algorithm viz ORCA [18] and RBRP [2]. Comparison of execution times on KDDcup data for the three algorithms is shown in Figure 7. Execution time of $GrADS$ includes grid (k=8) construction time and grading time. $GrADS$, which provides wider functionality, takes lesser execution time compared to $ORCA$. Initially it takes a little more time compare to $RBRP$ because of grid construction. After 1,50,000 records, when the

		k=6			k=8			k=10		
RN	RT	Signature	MD	RP	Signature	MD	RP	Signature	MD	RP
1	$\mathcal{S}$	(190, 90)	3	-	(190, 90)	4	-	(190, 90)	5	-
2	OR	$20 \leq x \leq 38$ $170 \leq y \leq 178$	5	100%	$20 \leq x \leq 38$ $170 \leq y \leq 178$	6	100%	$20 \leq x \leq 38$ $170 \leq y \leq 178$	7	100%
3	OR	$165 \leq x \leq 185$ $162 \leq y \leq 178$	5	100%	$165 \leq x \leq 185$ $162 \leq y \leq 178$	7	100%	$165 \leq x \leq 185$ $162 \leq y \leq 178$	9	100%
4	N	$1 \leq x \leq 33.5$ OR $67 \leq x \leq 100$ $4 \leq y \leq 10$ OR $59 \leq y \leq 92$	1	70%	$1 \leq x \leq 31.5$ OR $49 \leq x \leq 100$ $4 \leq y \leq 11$ OR $67 \leq y \leq 92$	2	82%	$1 \leq x \leq 31.5$ OR $55 \leq x \leq 100$ $4 \leq y \leq 21$ OR $59 \leq y \leq 92$	2	100%

TABLE I
SIGNATURE OF NOISY REGIONS FOR TWO DIMENSIONAL SYNTHETIC DATASET; RN: REGION NUMBER; RT: REGION TYPE; MD: DISTANCE FROM THE NEAREST CLUSTER, RP: REGION PURITY

		k=6			k=8		
RN	RT	Signature	MD	RP	Signature	MD	RP
1	$\mathcal{S}$	(99,98,33)	8	-	(99,98,33)	10	-
2	$\mathcal{S}$	$8 \leq x \leq 8$ $99 \leq y \leq 99$ $98 \leq z \leq 98$	7	100%	$8 \leq x \leq 8$ $99 \leq y \leq 99$ $98 \leq z \leq 98$	8	100%
3	OR	$86 \leq x \leq 90$ $75 \leq y \leq 84$ $65 \leq z \leq 74$	9	100%	$86 \leq x \leq 90$ $75 \leq y \leq 84$ $65 \leq z \leq 74$	10	100%
4	NR	$18 \leq x \leq 70$ $18 \leq y \leq 70$ $1 \leq z \leq 20$	4	93%	$32 \leq x \leq 70$ $45 \leq y \leq 70$ $1 \leq z \leq 20$	5	100%

TABLE II
SIGNATURE OF NOISY REGIONS IN THREE DIMENSIONAL DATASET; RN: REGION NUMBER; RT: REGION TYPE; MD: DISTANCE FROM THE NEAREST CLUSTER, RP: REGION PURITY

Clusters with Noise				Clusters after de-noising	
K=6		K=2		K=2	
Centroid	Data Points	Centroid	Data Points	Centroid	Data Points
(39,31)	385	(11,17)	411	(41,30)	360
(136,31)	304	(130,42)	1128	(130,40)	1109
(139,48)	250	-	-	-	-
(119,31)	225	-	-	-	-
(123,47)	342	-	-	-	-
(82,172)	33	-	-	-	-

TABLE III
IMPROVEMENT IN K-MEANS CLUSTERING SCHEME AFTER DE-NOISING TWO DIMENSIONAL DATASET;K: NUMBER OF CLUSTERS

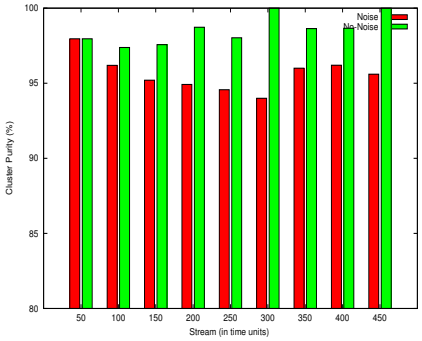


Fig. 6. Improvement in cluster purity after de-noising the KDDcup dataset as stream

number of leaves stabilizes (Figure 8), its execution time is more or less stable.

D. Effective Gradation Technique

We performed extensive testing with KDDcup data to explore the efficacy of gradation framework. We simulated a stream and applied the method after every 50K records. Since KDDcup data has classes varying from size 2,00,000 recs (biggest) to size 3 records (smallest) with same attack records often appearing consequently, it is highly suitable for testing gradation functionality.

We compared the detection rates of various attacks for $ORCA$, $LOADED$ and $GrADS$ taking detection rates for $ORCA$ and $LOADED$ as claimed in [1]. These algorithms detect attacks as outliers, while $GrADS$ identifies region type for each of the attacks. For each region, we compute the region purity and report in Table IV.

We found that comparatively bigger classes like smurf, ipsweep, portsweep etc. were reported as patterns and small attacks like nmap, buffer_overflow etc. were graded as noise. Thus $GrADS$ was able to detect all types of attacks whereas detection rate of $LOADED$ for bigger attacks was less.

W	FTPwrite		Ipsweep			Phf		Pod		Rootkit		Teardrop			WarezM*	
	DP	NR	DP	PR	NR	DP	NR	DP	NR	DP	NR	DP	PR	NR	DP	NR
1	6	3	197	106	81	1	1	20	20	0	0	99	0	99	0	0
2	2	1	563	512	130	2	2	20	20	0	0	100	0	99	20	16
3	0	0	102	512	98	0	0	62	60	7	3	198	100	72	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	99	0	99	0	0
7	0	0	256	0	252	0	0	41	40	0	0	383	0	383	0	0
8	0	0	0	0	0	0	0	99	0	99	1	1	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	129	0	129	1	1	22	20	2	2	100	0	100	0	0

TABLE V
GRADATION OF KDDCUP DATA, W: WINDOW NO., DP: DATA POINTS IN W, PR: POINTS IN PATTERN REGION, NR: POINTS IN NOISY REGION, WAREZM* : WAREZMASTER; PLS NOTE THAT FOR SMALLER CLASSES, PR IS NOT SHOWN AS COLUMN AS NO POINT OF THESE CLASSES WERE REPORTED IN PATTERN REGION

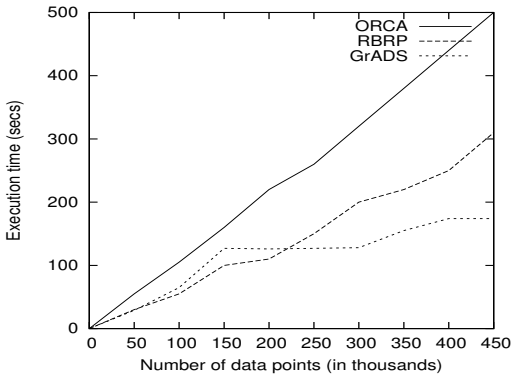


Fig. 7. Scalability test on KDDcup Data

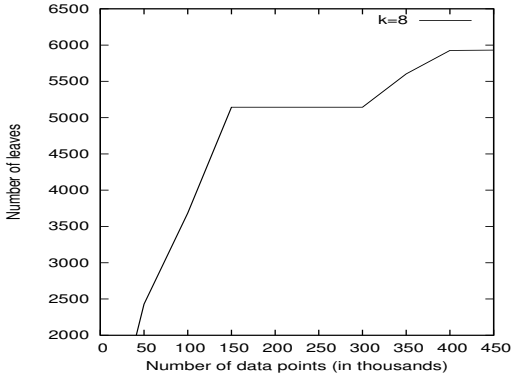


Fig. 8. Data size vs Leaves on KDDcup Data

Attack Type	$ORCA$	$LOADED$	$GrADS$
Buffer Overflow	63%	94%	87%
Back	5%	100%	5%
FTP write	88%	28%	50%
Guess Password	21%	100%	90%
Imap	13%	100%	92%
IP Sweep	3%	42%	89%
Land	66%	100%	100%
Load Module	100%	100%	55%
Multihop	57%	94%	45%
Neptune	1%	92%	80%
Nmap	8%	94%	98%
Perl	100%	100%	100%
Phf	25%	0%	100%
Pod	12%	45%	98%
Portsweep	13%	100%	81%
Rootkit	70%	33%	60%
Satan	9%	75%	93%
Smurf	1%	24%	93%
Spy	100%	100%	100%
Teardrop	1%	50%	97%
WarezClient	3%	48%	27%
WarezMaster	15%	0%	80%

TABLE IV
DETECTION RATE ON KDDCUP DATASET

In Table V, we selectively show attacks which have detection rate $\leq$ 50% in $LOADED$. The table shows the detailed description of these attacks in different windows, and their handling by $GrADS$. For example, Phf attack has only 4 records in dataset. Only one record appears in first window, which is detected as noisy region (not as a succinct outlier). Next window has two records and last window has the fourth record and each time it is graded as noisy region because of the distance criterion.

Better performance of $LOADED$ in some attacks e.g. Back, Loadmodule and Multihop can be attributed to the effective usage of categorical attributes for outlier detection whereas $GrADS$ ignores these attributes.

The effect of pruning is clearly visible in attacks $Teardrop$ and $Ipsweep$, where heavy clusters detected in one window show up in next window with lesser points even when no point for that cluster is received in next window. However small regions detected as noise (for other attacks), are pruned and removed from the grid.

E. Effect of Weight Threshold on Region Purity

We compared the purity of discovered clusters and noisy regions in two dimensional dataset (Figure 4) for two different computation of weight threshold. We used $\rho_1 = abs(n_\mu - 3 * n_\sigma)$ and $\rho_2 = (n_\mu + 3 * n_\sigma)$ as two computations and got the results shown in Table VI.

When ρ_1 is used for computing the threshold the results are close to expectation. region number 5 is reported as a cluster/pattern instead of noise because of the smaller value of the threshold. In the latter case the second cluster i.e. region number 2 splits into 3 different regions because higher weight threshold characterizes a pattern. The first of the three regions is a cluster, while the other two are noisy regions in its neighborhood. Notice that one of these noisy regions

subsumes another noisy region i.e. region number 3 thereby decreasing the purity. Thus threshold ρ_1 must be used when small sparsely populated clusters are desired whereas ρ_2 must be used when large dense clusters are needed.

RN	DP	ρ_1		ρ_2	
		DP (Purity)	RT	DP (Purity)	RT
1	350	354(99%)	P	354(99%)	P
2	1100	1103(99.7%)	P	762(100%) 174(100%) 213(77%)	P N N
3	53	46(100%)	N	-	-
4	12	12(100%)	N	12(100%)	N
5	21	21(100%)	P	21 (100%)	N
6	1	1 (100%)	O	1 (100%)	O

TABLE VI
EFFECT OF WEIGHT THRESHOLD ON PATTERN PURITY; RN: REGION NUMBER; DP: DATA POINTS; RT: REGION TYPE;

V. CONCLUSION

A framework for gradation of anomalies for data stream with bounded data space was proposed. The approach first segregates the patterns from anomalies, and subsequently analyzes and grades the anomalies. The proposed $GrADS$ algorithm is data driven and comprises of an online and an offline component. It reports three types of outliers and two types of noisy regions based on parameters computed from recent data. The signature of noisy regions reported by the algorithm add to user's understanding of the data. The experiments demonstrate the functionality and scalability of the proposed framework.

REFERENCES

[1] A. Ghoting, M. E. Otey and S. Parthasarathy: LOADED: Link-based Outlier and Anomaly Detection in Evolving Data Sets, ICDM. 2004.
[2] A. Ghoting, S. Parthasarathy and M. E. Otey: Fast Mining of Distance-Based Outliers in High Dimensional Datasets, SDM. 2006.
[3] Charu C. Aggarwal, Jiawei Han and Philip S Yu: A Framework for Clustering Evolving Data Streams, VLDB . (2003) 81–92.
[4] Chetan Gupta and Robert Grossman: Multiscale Analysis of Data: Clusters, Outliers and Noise - Preliminary Results, NASA Data Mining Workshop. 2006.
[5] Dragolijub Pokrajac, Aleksander Lazarevic and Longin J. Lateak: Incremental Local Outlier Detection for Data Streams, CIDM. 2007.
[6] Feng Cao, Martin Ester, W. Qian and A. Zhou : Density-Based Clustering over an Evolving Data Stream with Noise, SIAM. (2006) 326–337.
[7] J. Han and M. Kamber: Data Mining: Concepts and Techniques, Morgan-Kauffman. 2001.
[8] J. Zhang and H. Liu: D-GridMST : Clustering Large Distributed Spatial DBs, Classification and clustering for Knowledge discovery. (61–72) 2005.
[9] Kenji Yamanishi and Jun-ichi T.: A Unifying Framework for Detecting Outliers and Change Points from Non-Stationary Time Series Data, SIGKDD. 2002.
[10] KDDcup 99 Data: http://kdd.ics.uci.edu//databases/kddcup99/kddcup99.html.
[11] Levent Ertoz, Michael Steinbach and Vipin Kumar: Finding Clusters of Different Sizes, Shapes and Densities in Noisy, High Dimensional Data, SIAM. 2003.
[12] M. Ester, H. P. Kriegel, J. Sander and X. Xu.: A density-based algorithm for discovering clusters in large spatial databases with noise, KDD. (1996).
[13] M. H. Dunham: Data Mining: Introductory and Advanced Topics, Prentice Hall. 2003.
[14] M. M. Breunig, Hans-Peter Kriegel, Raymond T. Ng. and Jorg Sander: LOF: Identifying Density-based Local Outliers, ACM SIGMOD. Dallas, TX. 2000.
[15] Muhammad H. Arshad and Philip K. Chan: Identifying Outliers via Clustering for Anomaly Detection, TR CS-2003. Florida Institute of Technology Melbourne. 2003.
[16] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos and Prabhakar Raghavan: Automatic Subspace Clustering of High Dimensional data for Data Mining application, ACM SIGMOD. 1998.
[17] Robert V. Hogg and Allen T. Craig, Introduction to Mathematical Statistics: Pearson Education, Asia. 2002.
[18] S. Bay and M. Schwabacher: Mining Distance-based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule, ACM SIGKDD. 2003.
[19] S. Muthukrishnan, Rahul Shah and J. Scott Vitter: Mining Deviants in Time Series Data Streams, SSDBM. Greece. 2004.
[20] S. Papadimitriou, H. Kitawaga, P. B. Gibbon, and C. Faloutous: LOCI: Fast Outlier Detection using the Local Correlation Integral, ICDE. 2003.
[21] Suddhendu Biswas: Topics in Statistical Methodology, Wiley Eastern Limited. 1999.
[22] U. M. Fayyad, G. Piatetsky-Shaperio and et al.: Advances in Knowledge Discovery in Databases, AAAI/MIT Press. 1996.
[23] U. M. Fayyad, Piaatetsky-Shapiro G.and Smyth P.: Knowledge Discovery and Data Mining: Towards a Unifying Framework, KDD. Portland, OR. (1996) 82–88.
[24] Vasudha Bhatnagar and Sharanjit Kaur: Exclusive and Complete Clustering of Streams, DEXA. Germany. 2007.
[25] Yansheng Lu, Yufen Sun, Guiping Xu and Gang Liu: A Grid-Based Clustering Algorithm for High-dimensional Data Streams, ADMA. 2005.
[26] Y. Zhang, A. Luo and Y.Zhao: Outlier Detection in Astronomical Data, National Astronomical Observatories. 2004.

Intelligent E-Learning Systems

Peter Kazik, Igor Sivy, and Frantisek Jakab

***Abstract*— E-learning systems are information systems used for educational purposes. It is powerful way for teaching and training people in many areas of human activity. Besides many advantages, these systems also have disadvantages in process of education. Main disadvantage is generalization of learners, which means that all learners get the same studying materials and e-learning system do not have tools for satisfy learners individual study needs, like providing of other studying materials or assisting learner with a psychical aid. Intelligent e-learning system might be possible solution for this problem. Intelligent e-learning systems are rather new technology with basic principles identical with e-learning systems, but they also contain so-called intelligent principles for eliminating learner generalization with individualized access to learners study. This can also make learner study more effective. There are many ways for implementation of this intelligent principles, selecting from a set of intelligent principles is based on character of studying material, format of study course, etc. This work discusses mainly used principle which is adaptability principle, where system stores information about progress of learner study and make appropriate actions.**

***Index Terms*— adaptability, e-learning, intelligence, metadata, optimal path, student model**

I. Introduction

Problem of education is mostly connected with knowledge-based economy, which has been defined as main factor for development in our community, because only properly trained or skilled persons can mainly support this development. However, quality of knowledge mostly depends on quality of education. At present, new information and communication technologies have enabled development of new learning and teaching environments [1]. Intelligent e-learning systems (IELS), which have basic principles identical with standard e-learning systems (provide education), are example for this development. Main reason for developing these systems is that standard e-learning system contains disadvantages, like generalization of learners and inability to provide individualized access to learner study. By this disadvantage, all learners get the same studying material, which might be comprehensible for one learner, but might be incomprehensible for other learner. IELS try to eliminate this problem with intelligent principles (algorithms). There are many approaches for these intelligent principles, selection and application of these principles depends for example on type of studying material or format of courses. This work will discuss mainly used principle - principle of adaptability. By this principle, IELS store information about learner's progress in study into so-called student model and next system action is based on that stored information. IELS also have his own system architecture, which may contain new modules, which are not presented in standard e-learning systems. Important part for is its metadata specification for learning object in data repository. This specification can increase efficiency of system in searching for alternative materials or changing the structure of learning material (if possible).

Manuscript received October 9, 2001. This work was supported in part by the U.S. Department of Commerce under Grant BS123456

P. K., phone: +421 904 312 085; e-mail: kazik.peter@centrum.sk
I. S., from Department of Computer and Informatics, Technical University of Košice, Letná 9, 042 00 Košice, Slovakia, e-mail: sivy@elfa.sk.
F. J., from Department of Computer and Informatics, Technical University of Košice, Letná 9, 042 00 Košice, Slovakia, e-mail: jakab@elfa.sk

II. Analysis

A. E-Learning systems

E-learning systems are information systems mostly used in on-line education. It can be:

- General – can serve education in many areas
- Problem-oriented – have supporting tools for studying of a particular subject, study of other subject can be problematic.
- Combination of general and problem-oriented, example for this system is [4].

Problems in this form of education are eliminated with embedded intelligent principles, which transform standard e-learning system into intelligent e-learning system.

B. Development of IELS

Development of IELS consists from several steps:
Analysis of teaching materials and courses, which can be provided in developed system, is the first step. This is followed by selection of the most appropriate intelligent supporting technology. For example, e-learning systems for teaching programming languages must have built-in lexical and syntactic analyzers for a selected language and components for intelligent analysis of student results.
Project of architecture of new system is certainly the next step. New system can be developed, or an existing system may be updated. Nevertheless, new architecture must contain new modules for selected intelligent support.
Selection of metadata specification is an important step, which is also based on analysis of teaching materials and courses.

Programming and technical development must be obviously based on outputs from previous steps.

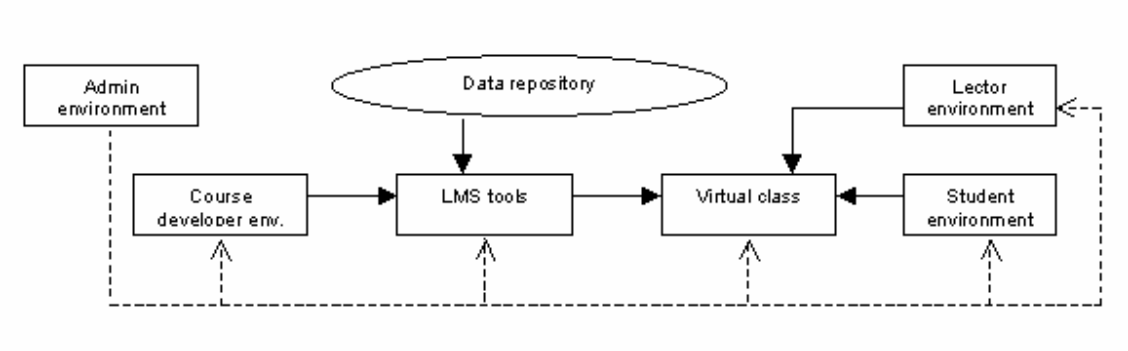


Fig. 1: Architecture of Standard E-learning System

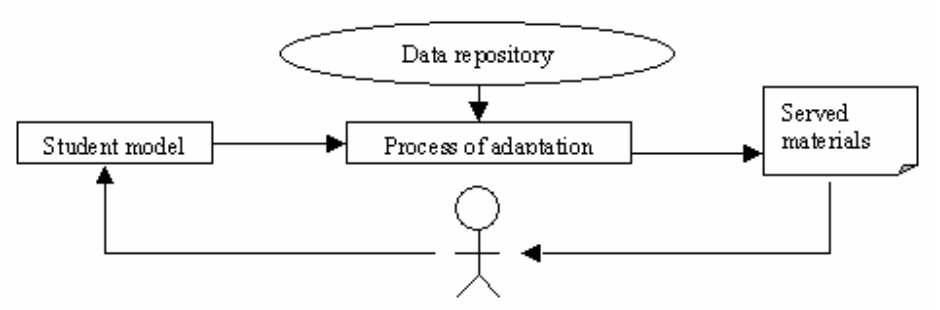


Fig. 2: Principle of Adaptability

C. Architecture of Standard E-learning System

Important prerequisite for designing architecture for IELS is analysis of architecture for standard e-learning system. This architecture is shown on the Figure 1 and consists from these basic modules:

- LMS tools - contains tools for course manipulation and can have tools for creating learning objects.
- Virtual class - can simulate standard classes known from school. Lector can bind this class with some course and manage users for this class.
- Data Repository - disk space containing learning objects.
- User environments – usually web-browsers for on-line e-learning system.

This architecture is also demonstration of the fact, that standard e-learning system does not have modules (components) for supporting individualized access to learners study.

D.Principle of Adaptability

This principle brings support for individualized access to learner study. It is shown on Fig. 2 and principle can be considered as follows: Information about learner study progress is stored in a student model [6]; this information tells system rules for managing teaching materials from data repository and then are these materials served to learner. If the materials invoke a change in learner study progress, this information is again stored in student model.

Mostly used technologies based on principle of adaptability are [3]:

- Intelligent tutoring systems
  - Curriculum sequencing (Optimal Path problem)
  - Problem solving support technologies (Intelligent analysis of student solutions, Interactive problem solving support, the example-based problem solving support)
- Adaptive hypermedia systems
- Student model matching

That was only few technologies used in IELS with adaptability support. Theory of IELS is relatively new and is being constantly developed.

After selection of intelligent principle, it is possible to project architecture of IELS. Example of this architecture can be found in “Solution and results”.

E. Metadata specification

Every teaching object used in system must have its own metadata. Metadata are information about data (learning objects) and they are usually stored in database tables, as it is shown on the Fig. 3. There are many metadata specifications, for example [2]:

- Dublin Core
- Ariadne
- IMS
- IEEE-LOM
- GEM
- Advanced Distributed Learning – SCORM

Some of these specifications can be used as templates for updated metadata specification. For example, [7] is an interesting work, which describes useful so-called ecological

approach for specification IEEE-LOM.

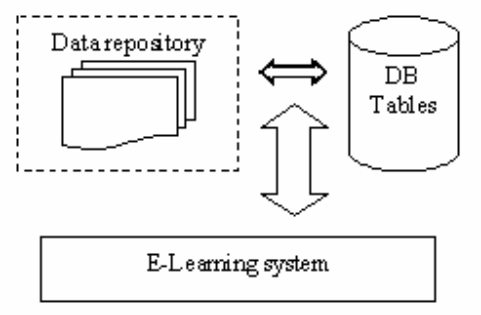


Fig. 3: Use of Metadata

III. SOLUTION AND RESULTS

Principles from this work were tested on existing e-learning system eEduser.

First step was analysis of courses and teaching material, which can be used through this system. Assignment defines, that eEduser is a general, not problem-oriented e-learning system, which can simulate original form of education known from schools. Simulation of this school education can be accurate by using principle of adaptability: technology of Optimal Path.

A. Optimal Path Problem

Skeleton of the problem is based on the rules (binds) between teaching materials. Categorizing of teaching materials in two groups is a start for rule set up:

- Theoretical materials (or objects) – all object that does not evaluate learned knowledge
- Object for knowledge-checking – like tests

Then the rules are:

- *if* some theoretical object was seen/read *then* make other theoretical object/tests accessible
- *if* test for some theoretical object was successfully accomplished *then* make other theoretical object accessible

These rules must be defined by a course developer. If learner fails in a test, system will look up for another teaching material and provide it to learner. Example of this can be seen on the next figure:

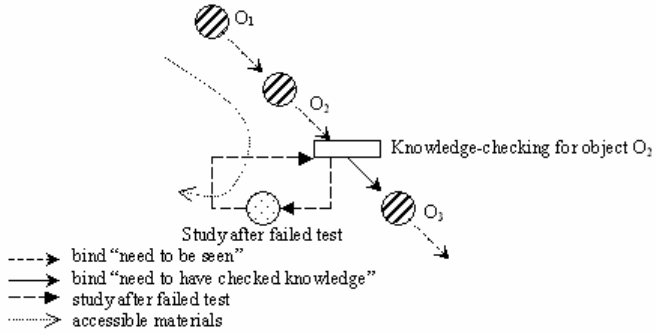


Fig. 4: Example of Optimal Path

B. Architecture of IELS

New system architecture must contain new modules for intelligent principle of optimal path. This architecture is shown on Figure 5 and contains the following new modules:

- *Bind module* – a module, in which course developer defines if-then rules. System needs for storing rules the following information: identification of course, identification of learning objects and type of bind.
- *Student model* contains information about learner progress in study. Models are implemented as data records in database tables. For this version of optimal path student model contains information how many times theoretical object was visited from course, calculated balance, which depends on bounded tests and information about additive teaching materials, which are served to learner, when he fails in a test.
- *Adaptability module* is a part of source code, which is responsible for managing (control of accessibility) of teaching materials provided to student. This managing depends on information from Bind module and Student model.
- *Monitoring module* is a part of a source code, responsible for storing information in student model.

C. Metadata Specification

Metadata specification is next part in development of IELS. Selected specification is Dublin Core [5]. It is a specification used in libraries with so-called conservative access to objects metadata, but sufficient to applied problem. Basic specification involves 15 elements: title, creator, subject, description, publisher, contributor, date, type, format, identifier, source, language, relation, coverage, rights [4]. This basic specification was updated for some new elements. Final specification elements were categorized into 4 groups:

- *Basic information* (DC_Basic) contains identification of data record, Title, Description, Source Thema, Keywords, Languages and Version
- *Rights information* (DC_Rights) contains Price, Creator, Contributor, Publisher, Coverage, basic description of rights, information about starting and ending date (time) of accessibility of data source and information about acceptance of source.
- *Format and Type information* (DC_Format_and_Type) contains basic file name for object (ev. so-called index file), type of source, storing information and some eventually needed information.
- *Relation information* (DC_Relation) are stored information on relation between individual data sources. It can help system in finding best additive teaching materials to satisfy learner needs.
- *Other information* (DC_Others) consist from information about Competencies, Knowledge, needed Skills, Prerequisites, Department and History (uses of object).

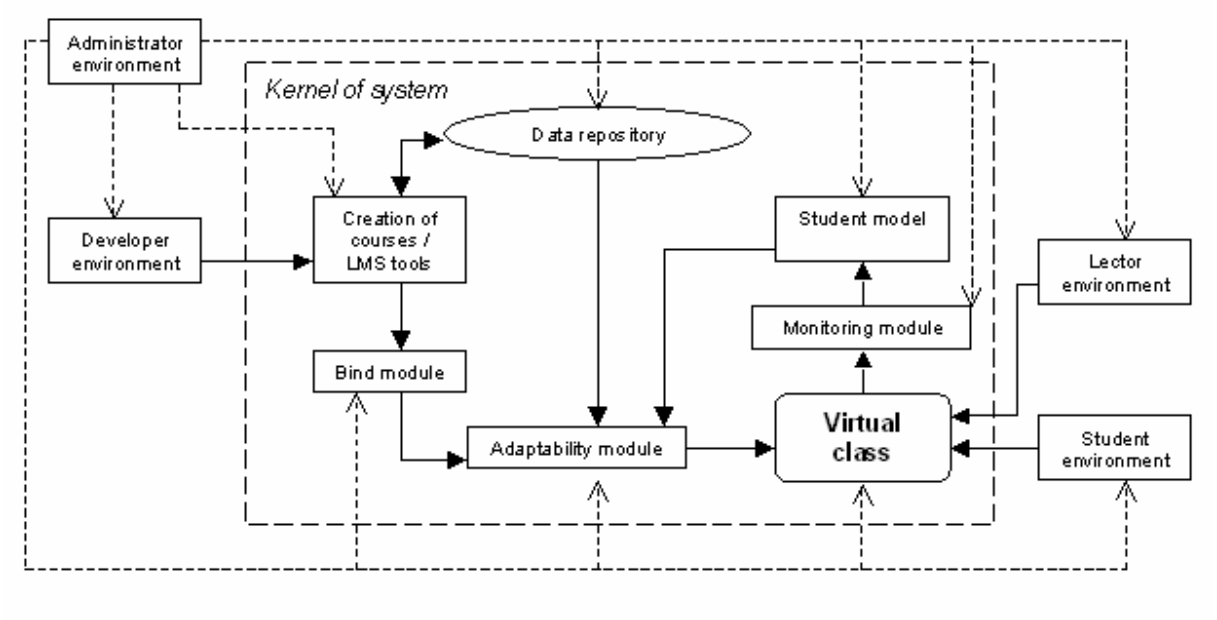


Fig. 5: Architecture of IELS

D.Additive Teaching Materials

Searching for alternative and additive data sources would be last solution provided to particular learner, which would be stored in IELS. If student fails in a test, system would get metadata from theoretical object, for which failed test was bounded. System then uses these elements for finding similar learning object in data repository. This algorithm is shown on the next figure:

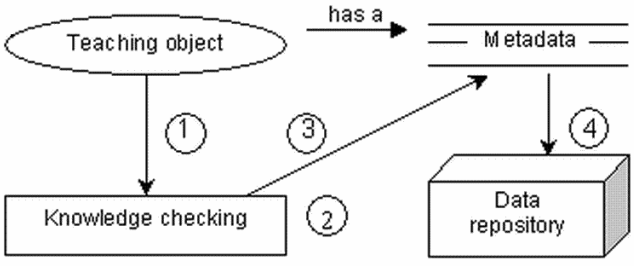


Fig. 6: Searching for Additive Materials

Usable elements are:

- Title – learning object with the same title can probably tell information about same topic, but in understandable way.
- Keywords – classical search algorithm. A certain level of keywords agreement (for example 70%) must be set.
- Source Theme could be alternative to description, but it is shorter.
- Author/Creator – same author can probably create other teaching materials, which cover the same topic. Searching with this element must be combined for example with keyword searching with lover level of agreement, etc.

Information about founded teaching materials would be stored into student model.

IV. CONCLUSION AND FUTURE WORK

This work is an introduction into theory of IELS. We have founded, that a principle of adaptability is mostly used principle for IELS. There is a set of technologies, which can be used in different IELS. Optimal path was used as selected technology and it was tried to implement it into existing e-learning system. This technology requires rebuild of system architecture and addition of some new modules. Changing of metadata specification can be also useful part of development. This work maps beginning of this technology. In future work, this technology can be combined with technology of intelligent analysis of students result. Then, if a student fails in a test, system will find alternative materials for study only for wrongly answered questions. New metadata specifications and rebuild of system architecture might be required.

REFERENCES

[1]. STANKOV, Slavomir - ŽITKO, Branko - GRUBIŠSIĆ, Ani: Ontology as a Foundation for Knowledge Evalution in Intelligent E-learning Systems. 12th International Conference on Artificial Intelligence in Education, Armsterdam, Netherlands, 2005, pp. 81-84

[2]. SADDIK, A. E. E-Learning Standards: The Dawn Has Broken. School of Information Technology and Engineering, University of Ottawa, Canada, 2001.

[3]. BRUSILOVSKY, Peter: Adaptive and Intelligent echnologies for Web-based Education, Special Issue on Intelligent Systems and Teleteaching.

[4]. PEYLO, Christoph – THELEN, Tobias – ROLLINGER, Claus – GUST, Helmar: A web-based intelligent educational system for PROLOG, Osnabrück 2000, Institute for Semantic Information Processing – University of Osnabrück.

[5]. Dublin Core Metadata Element Set, Version 1.1
http://dublincore.org/documents/dces/
[6]. BUREŠ, Miroslav - JELÍNEK, Ivan: Formální popis adaptívniho webového systému, Praha 2004, Odborná skupina Webing, katedra počítačú ČVUT FEL
[7]. BROOKS, Christoper - McCALLA, Gordon - WINTER, Mike: Flexible Learning Object Metadata, 2005, 12th International Conference on Artificial Intelligence in Education, Amsterdam, Netherlands, pp. 1 – 8

Peter Kažik: born in 1982, has successfully graduated on the high school Stredná priemyselná škola elektrotechnická in Košice, in year 2001 and completed his university studies by taking a state examination in the study field of Computers and Informatics on Technical University of Košice in year 2007.

Igor Sivý: born in 1960, received the MSc. degree in Systemotechnic engineering from the St. Petersburg Electrotechnical Institute (Russia) in 1984 and the PhD. degree in 1988. He works as an assistant professor at the Dept. of Computers and Informatics, Technical University of Košice. His research interests include operating systems, e-learning systems and intelligent tutoring systems.

František Jakab: born in 1959, received the MSc. degree in Systemotechnic engineering from the St. Petersburg Electrotechnical Institute (Russia) in 1984 and the PhD. degree in 2005. He works as an assistant professor at the Department. of Computers and Informatics of the Technical University of Košice, Slovakia. He is a head of the Computer Engineering Group and Computer Networks Laboratory (www.cnl.tuke.sk). His research interests include projecting of computer network, modeling, simulation and network management, new form of multimedia-based communication, QoS, telelearning systems, intelligent tutoring systems.

ISAST Transactions on Intelligent Systems, No. 1, Vol. 1, 2008
K. Altenburg et al.: Simian: A Multi-agent Simulation Framework for Decentralized UAV Task Allocation

Simian: A Multi-agent Simulation Framework for Decentralized UAV Task Allocation

Karl Altenburg, Michael Hennebry, Jonathan Pikalek, and Kendall E. Nygard

***Abstract*—**Simian is an extensible, multi-agent simulation environment designed to facilitate the development and evaluation of intelligent task allocation techniques for systems of unmanned aerial vehicles (UAVs). The objective of Simian is to provide researchers with development scaffolding for the empirical investigation of command and control of massively parallel UAV operations. Using Simian, the researchers have simulated a variety of many-UAV reconnaissance and strike missions while exploring the application of emergent intelligence.

***Index Terms*—**Agents, Emergent Intelligence, Multi-agent, UAV.

***Areas of Interest*—**2.4 Autonomous Agents, 2.18 Modeling and Simulation

I. Introduction

SIMIAN is an extensible, multi-agent simulation environment designed to facilitate the development and evaluation of new intelligent task allocation techniques for unmanned aerial vehicles (UAVs). Simian is a single-threaded, discrete-time-stepped simulator core written in Java. Simian can be run in conjunction with or without 3D visualization. The primary design objective is to provide researchers with development scaffolding for the empirical investigation of the application of emergent intelligence methods for UAV command and control.

We emphasize that Simian is not a general purpose simulation environment, rather, it is designed specifically for modeling systems of UAVs. The simulation environment provides domain-specific features for UAV simulations "out-of-the-box" and accommodates researchers who want to treat various aspects of UAV simulations as "black boxes." Collision detection and avoidance is an example of a ubiquitous aspect of UAV research for which implementation is a nontrivial task. But collision detection is generally of interest only in conjunction with other domain-specific features.

The design of Simian is extensible, allowing for future intelligent control components to augment or replace existing components. Thus, Simian is highly customizable, so that researchers can easily develop and integrate their own components. In turn, these new components can augment Simian for the use of future researchers.

Manuscript received January 28, 2008.

K. Altenburg is with the Department of Accounting and Information Systems, North Dakota State University, Fargo, ND 58105 USA (phone: 701-231-6651; fax: 701-231-6545; e-mail: Karl.Altenburg@ndsu.edu).

M. Hennebry is with the Center for Nanoscale Science & Engineering, North Dakota State University, Fargo, ND 58105 USA (e-mail: Michael.Hennebry@ndsu.edu).

J. Pikalek is with the Department of Computer Science and Operations Research, North Dakota State University, Fargo, ND 58105 USA (e-mail: Jonathan.Pikalek@ndsu.edu).

K.E. Nygard is with the Department of Computer Science and Operations Research, North Dakota State University, Fargo, ND 58105 USA (e-mail: Kendall.Nygard@ndsu.edu).

II. History and Motivation

Simian evolved from the simulation needs of several UAV research projects. This section summarizes the strengths and weakness of the Simian predecessors.

Several related simulators were designed to investigate the use of optimization models such as the capacitated transshipment problem (CTP) to determine task allocation and sequencing among UAVs [4][10][12]. While functional, these simulators have the disadvantage of using a shared memory model for inter-agent communication, under the assumption that complete global communication among all UAVs was always available. This assumption is invalidated for realistic ad hoc networks, and motivated us to incorporate a loosely coupled and highly configurable interagent communication model in Simian. Another collection of related simulators was designed to investigate emergent intelligence methods for directing UAVs [1][8][13]. These simulators model each UAV as a separate thread, but do not directly schedule the threads. This leads to difficulties in replicating experiments, as the order of thread execution can affect the outcome of some events. For example, greedy task allocation schemes induce precedence relationships among the tasks that must be enforced. Another drawback is the tight coupling between the simulated UAV actions and their graphical visualizers. For example, in one simulator, collision detection is supported through manipulation of the pixel representation of the UAVs. In the Simian design, independence from the visualizers was enforced.

Other simulators were developed to investigate specific intelligent task allocation schemes. These include multi-agent auction models [11] and neural networks [7]. While qualitative comparisons among the methods implemented in these simulators are possible, quantitative comparisons are nearly impossible [9]. These considerations led to the development of a generic and flexible cross-platform environment based on the Open Agent Architecture (OAA) [3][6] for the message

passing infrastructure. Using OAA allowed development to focus on the simulator rather than interagent message passing. Unfortunately, message passing was a performance bottleneck, primarily due to the slow communication mechanisms. In addition, errors in the C++ bindings hindered supporting multiple development languages.

Although building upon OAA did not satisfy the goal of simulating very large collections of UAVs, its basic architectural design had a number of desirable characteristics that were not dependent on the use of the OAA infrastructure. Inspired by the design choices made in the OAA environment, the architecture of Simian evolved rather quickly.

III. Design Goals

Based on our experiences with previously developed simulations, we adopted the following design goals.

A. Separating Physics and Strategic Logic through Layers of Abstraction:

In Simian, simulating the detailed movements of a UAV move does not require the user to write code that follows the laws of physics. UAV motion is encapsulated in an abstract layer and accessed by other layers through a well specified interface. The layers are loosely coupled and allow for ease of modification or replacement. The four primary layers used in representing UAV operations are the strategist (strategic logic), autopilot (UAV movement control), physical (motion and displacement), and environment (interagent communication and other object interactions).

B. Separating Problem and Strategy Specifications:

The OAA-based simulator was specifically intended to separate problem specification from strategy. Borrowing this idea, input in Simian comes from XML files and any Java classes to which the XML files refer. The top-level XML elements are: 1) situation, 2) policy, 3) strategy, and 4) global. The physical and political situation is described by situation and policy elements. Each UAV's strategy is described by a corresponding element within a strategy element. Strategy elements can be changed without changing the physical and political situation.

C. Consistent Support for Three Dimensional Spatial Representations:

UAVs operate in a three-dimensional world. Accordingly all UAV functions supported by the simulator must use three-dimensional data.

D. Support for the real-world units of measure:

Standard for measurement allows for comparisons with other simulations as well as real UAV operations. To support model validation, Simian enforces consistency among units of measure, that is, combinations of meters, kilograms and seconds. This idea was advanced in modeling in [2]. These conventions permit a standardized method for the analysis and transfer of simulation results.

E. Determinism:

The ability to replicate an experiment is critical in simulation; therefore, determinism is an important design goal. Determinism is accomplished by running the simulation on a single thread. Control algorithms that make use of potentially non-deterministic services such as random number generation must ensure that determinism is maintained (for example, seeding a pseudo-random number generator with user input). In addition, all control decisions are based only on simulated time.

F. Efficiency and Execution Performance:

Simian is potentially used in real-time decision support and thus requires fast execution and interrupt support. Thus, communication message support among agents was refined to maximize execution speed.

G. Separating Visualization and Simulation:

Simulation and visualization can be CPU intensive; therefore it is desirable for the visualizer to run on a separate host. We support simulation and visualization as loosely coupled separate applications that run as separate processes. This design facilitates the development of alternative visualizers without requiring changes to the simulator and allows the simulator and visualizer to be developed and tested independently.

H. Portability:

Portability requires using a language and libraries that are available on multiple platforms and perform consistently on each platform. Java, C and C++ were the contenders for the language. Java was chosen because it is the most popular language among the developers, and because the *dom4j* library for handling XML is written in Java. Control algorithms are specified by their Java classes. Simian uses reflection to load classes named in its XML input.

I. Realistic Communication:

Cooperation is achieved only through communication. Modeling interagent communication, such as radio broadcasts, is essential for this type of model. Also, due to the potential volume of communication and the number of interacting agents, communication must be modeled very efficiently and not be a bottleneck for simulation performance.

J. Post-simulation Scoring:

Policy can be vague or fluid. After a simulation is run, it is typically useful to re-evaluate the results according to a standard that had not existed earlier. By recording the simulation's state throughout an experiment, new analysis can be applied to the record of the experiment to answer new questions and allow better comparisons with other systems.

IV. Architecture

Conceptually, Simian decomposes UAV simulation into four primary layers: 1) strategist, 2) autopilot (optional), 3) physical and 4) environment. Figure 1. depicts this architecture.

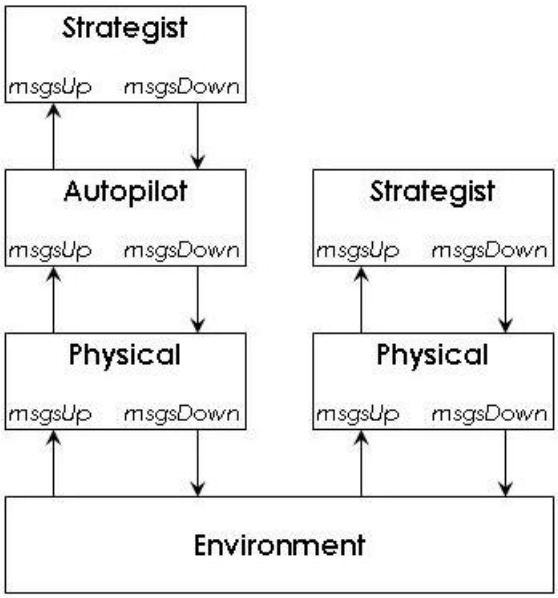


Figure 1. The Simian control layer architecture. Each UAV (or other agent) may contain one or more control layers.

A UAV's physical layer represents its hardware, that is, the UAV itself and its radios, sensors and weapons. Methods for describing physical motion are contained in this layer. A UAV sends messages to its immediate controller object with a method call and fetches commands with another method call. Simian typically handles spatial information in the form of a *Vector3D* object. A *Vector3D* object stores three public doubles: *x*, *y*, and *z*. *Vector3D* also has methods for convenient vector arithmetic.

The environment layer lies below each UAV's physical layer. There is only one for all agents in a simulation. It provides time-stepped simulation and handles interactions between objects. Collision detection, radio messages, and all events involving more than one physical object are handled in the environment layer. To this end, the environment has access to every physical object which might interact with another. The environment layer communicates with a physical layer through method calls on the physical layer and some assignments to internal data.

A UAV's control layers lie above its physical layer. The layer immediately below a control layer communicates with it through method calls on the higher layer. Data only goes up and commands only come down. Most UAVs have two strategist and autopilot control layers. The topmost control UAV layer is called its strategist. The strategist understands its team's policy and directs the UAV accordingly. Commands from a strategist go to the layer below. This might be the physical layer, but is more likely to be another control layer, such as an autopilot. A UAV's optional autopilot layer, if there is one, lies just above its physical layer and provides convenience commands such as: go to position X, adjust speed, adjust velocity, and adjust heading. These initiate sequences of commands understandable by its physical layer. An autopilot offers strategists an alternative to issuing low-level vehicle control commands.

Interlayer communication is driven exclusively from below. No layer has direct access through a reference to a layer below it. A control layer does have access to its UAV's initialization data. A layer sends messages by invoking the *msgsUp* method of the layer immediately above it and receives messages by invoking the *msgsDown* method of that layer. A UAV has new information at least once every clock tick. It promptly sends that information to its controller. A UAV requests commands at least once every clock tick.

A. Reporter:

The reporter's job includes recording score-related information. The reporter need not know the precise scoring standard to be applied, but it does need to know the kind of data to store. The current standard reporter does this job by copying most messages sent to it to the standard output. If told to do so, it also provides data to a visualizer through a TCP port or to a file. Because of the loose coupling, neither the standard reporter nor the standard visualizer needs the other. Each simply needs a partner that communicates in the same language. In Simian's XML data, the reporter's class is specified in a sub-element of a top-level global element.

B. Visualization:

The visualizer provides a visual representation of the simulation. The standard visualizer understands the messages sent by the standard reporter through a TCP port. As stated above, the visualizer is independent of the simulation and reporter. The Visualizer runs as a separate process. The current visualizer was written in C++ and generates its images with *OpenGL*. Figure 2 and Figure 3 show two views from the visualizer.

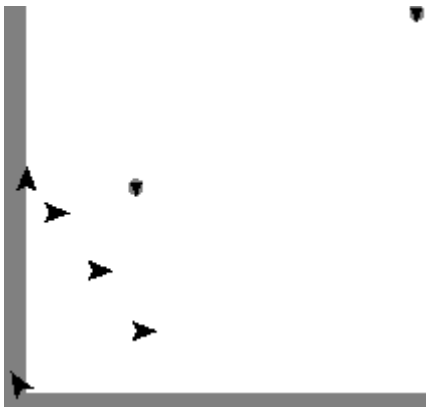


Figure 2. Visualizer displaying UAVs searching for a threat.

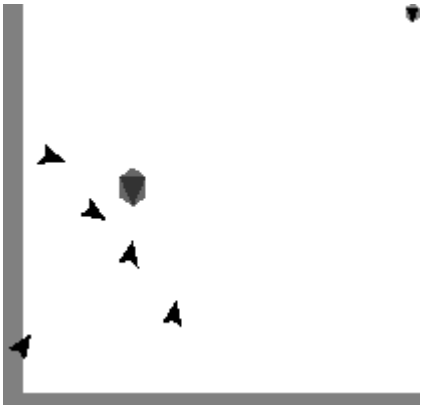


Figure 3. Visualizer displaying UAVs dispatching a threat. A UAV expended itself in a strike and a new UAV is sent in to replace it.

C. Initialization:

Simian uses XML for initialization; converting interagent messages to XML throughout the simulation is no longer an efficiency concern. XML provides a consistent self-describing framework. Without such a framework, subtle errors are easy to introduce with a new input format. For example, suppose one changes the format of an input line so that it uses five inputs where it formerly used six inputs. The meaning of the remaining five will typically change. If the actual input is not changed, an old format line will almost certainly be accepted and quietly accomplish what one does not want. A similar change in an XML format would normally change the names by which the five numbers are referenced. Failure to change the actual input would result in some of the numbers not being found. We believe that a loud complaint is better than quiet error. Below is a fragment of an initialization file in XML for a simple simulation situation:

sit.xml

```
<situation>
 <model name='Predator' class='uav' crashradius='5' >
  <!− crashradius=5 meters for collision detection − >
  <limits
      :
      fwrd_ac='3'
      speed='20'
      fuelCapacity='300'
  />
 </model>
 <team name='USA' >
  <smallthing model='Predator'
      name='EagleEye' >
   <position> 0 0 30</position>
   <velocity> 4.0 1.0 0 </velocity>
  </smallthing>
 </team>
</situation>
```

V. Future Work

A central design goal of Simian is the ability to easily exchange strategy elements for the UAV agents. To support this feature, a library of intelligent control modules is being designed and implemented. Modules that have been used in previous work to be included in the Simian library incorporate: neural networks, Bayesian belief networks, Petri nets, partially observable Markov decision processes (POMDP), auction models, and capacitated transshipment problem formulations.

Projects to exploit emergent intelligence with autonomous team formation are underway. In one scenario, many UAVs are deployed in a search-and-strike mission. As potential threats are discovered, teams are autonomously formed through interagent communication and negotiation. This project extends the emergent intelligence paradigm demonstrated in earlier work. It is envisioned that Simian will be incorporated in the model base of real-time decision support architectures (for example, see the Emergent Adversary Behavior component in [5]).

Currently Simian has no provision for gravity or for terrain. To compensate for the absence of gravity, we adjust the accelerations available to a UAV. There is no help for the absence of terrain. Absent gravity and terrain, the world is neither flat nor round. If we add terrain and gravity and make the world round, some of Simian's physics would changed for a round world. In the first draft of UAV motion, a UAV does not bank. It's vertical axis is assumed to be parallel to the z-axis. In a round world, the z-axis will not always be parallel to gravity. Also, in the standard autopilot, altitude is equal to z.

Simian does not come with the values of any physical constants. The radius of the earth in meters, used to address the effect of the curvature of the Earth on line-of-sight communication, along with other pertinent values, is scheduled to be added.

VI. Conclusion

Simian addresses many of the key needs for research in decentralized many-UAV task allocation. The requirements for a layered, modular architecture, high efficiency, distributed execution, and ease of inter-simulation comparison have been met. Flexibility of the design has been demonstrated by re-creating several scenarios from previous works. We feel that this research tool will benefit not only our research efforts, but also those of others interested in UAV-centric, massively parallel, agent-based simulations.

References

[1] K. Altenburg, J. Schlecht, and K.E. Nygard, "An Agent-based Simulation for Modeling Intelligent Munitions," in *Proceedings of the Second WSEAS Int. Conf. on Simulation, Modeling and Optimization*, Skiathos, Greece, 2002.

[2] G.H. Bradley, and R.D. Clemence, "Model integrated with a typed executable modeling language," in *Proceedings of the Twenty-First Annual Hawaii International Conference on Decision Support and Knowledge Based Systems*, pp. 403 - 410, Kailua-Kona, HI, 1988.

[3] A. Cheyer, and D. Martin. "The Open Agent Architecture," *J. of Autonomous Agents and Multi-Agent Systems*, vol. 4 , no. 1, pp. 143-148, March 2001.

[4] A. Dargar, A. Kamel, G. Christensen and K.E. Nygard, "An Agent Based Framework for UAV Collaboration," in *Proceedings of the ISCA*

11th International Conference on Intelligent Systems, pp. 54-59, Boston, MA, 2002

[5] D.A. Gilmour, J.P. Hanna, W.A. Koziarz, W.E. McKeever, and M.J. Walter. (2005, February). High-Performance Computing for Command and Control Real-Time Decision Support. *AFRL Technology Horizons* [Online].Available: http://www.afrlhorizons.com/Briefs/Feb05/IF0407.html

[6] N. Huff, A. Kamel, and K.E. Nygard, "An Agent Based Framework for Modeling UAVs," in *Proceedings of the 16th International Conference on Computer Applications in Industry and Engineering (CAINE_03)*, Las Vegas, November, 2003.

[7] A. Kamel, M.M. Anwar and K.E. Nygard, "Neuro-Dynamic Programming for Task Allocation to Unmanned Aerial Vehicles.," in *Proceedings of the ISCA 13th International Conference on Intelligent and Adaptive Systems and Software Engineering*, Nice, France, July 1-3, 2004.

[8] C.A. Lua, K. Altenburg, and K.E. Nygard. "Synchronized Multi-Point Attack by Autonomous Reactive Vehicles with Simple Local Communication," in *Proceedings of the IEEE Swarm Intelligence Symposium*, Indianapolis, Indiana, 2003

[9] K.E. Nygard, K. Altenburg, J. Tang, D. Schesvold, and J. Pikalek. "Alternative Control Methodologies for Patrolling Assets with Unmanned Air Vehicles," *Lecture Notes in Economics and Mathematical Systems: Cooperative Systems – Control and Optimization*, vol. 588, pp. 105-116.

[10] K.E. Nygard, P.R. Chandler, and M. Pachter, "Dynamic Network Flow Optimization Models for Air Vehicle Resource Allocation," in *Proceedings of the American Control Conference*, Arlington, 2001.

[11] J. Patterson. *Market-Based Adaptive Task Allocation for Autonomous Agents.* Master's Thesis, Dept. of Computer Science and Operations Research, North Dakota State Univ., Fargo, ND, 2004.

[12] J. Pikalek. *An Optimization Model for Autonomous Agent Task Planning*. Master's Thesis, Dept. of Computer Science and Operations Research, North Dakota State Univ., Fargo, ND, 2005.

[13] J. Schlecht, K. Altenburg, B.M. Ahmed, and K.E. Nygard. "Decentralized Search by Unmanned Air Vehicles using Local Communication," in Joshua R, Y. Mun (eds.), *Proceedings of the International Conference on Artificial Intelligence*, Volume II, Las Vegas, NV. 2003, pp. 757-762.