Robotica (2002) volume 20, pp. 499–508. © 2002 Cambridge University Press
DOI: 10.1017/S0263574702004307 Printed in the United Kingdom

DMiner-I: A software tool of data mining and its applications

Jie Yang, Chenzhou Ye and Nianyi Chen

Institute of Image Processing & Pattern Recognition, Shanghai Jiaotong University, Shanghai 200030 (China)

(Received in Final Form: May 11, 2002)

SUMMARY

A software tool for data mining (DMiner-I) is introduced, which integrates pattern recognition (PCA, Fisher, clustering, HyperEnvelop, regression), artificial intelligence (knowledge representation, decision trees), statistical learning (rough set, support vector machine), and computational intelligence (neural network, genetic algorithm, fuzzy systems). It consists of nine function models: pattern recognition, decision trees, association rule, fuzzy rule, neural network, genetic algorithm, HyperEnvelop, support vector machine and visualization. The principle, algorithms and knowledge representation of some function models of data mining are described. Nonmonotony in data mining is dealt with by concept hierarchy and layered mining. The software tool of data mining is realized by *Visual C++* under Windows 2000. The software tool of data mining has been satisfactorily applied in the prediction of regularities of the formation of ternary intermetallic compounds in alloy systems, and diagnosis of brain glioma.

KEYWORDS: Data mining; Knowledge representation; Decision trees; Brain glioma diagnosis.

1. INTRODUCTION

Data mining[1] is an important branch of up-to-date intelligent system theories. It combines several advanced techniques such as artificial intelligence, computational intelligence (artificial neural network, genetic algorithm), pattern recognition, database (data warehouse, OLAP) and statistics together to discover valuable and hidden knowledge from databases. We have built a software tool for data mining (DMiner-I, Figure 1). It consists of nine function models: Pattern recognition, decision trees, association rule, fuzzy rule, neural network, genetic algorithm, HyperEnvelop, support vector machine and visualization.

- **Function model of pattern recognition**: Five approaches of pattern recognition (PCA, Fisher, LMAP, KNN, PLS) are used for data evaluation and selection, feature selection, clustering and modeling for classification.
- **Function model of decision tree:**[2,3] Algorithms of decision trees (ID3, OC1) are used for a model of classification.
- **Function model of association rule**: The rough set[4,5] is used for extracting association rules, and for feature selection.

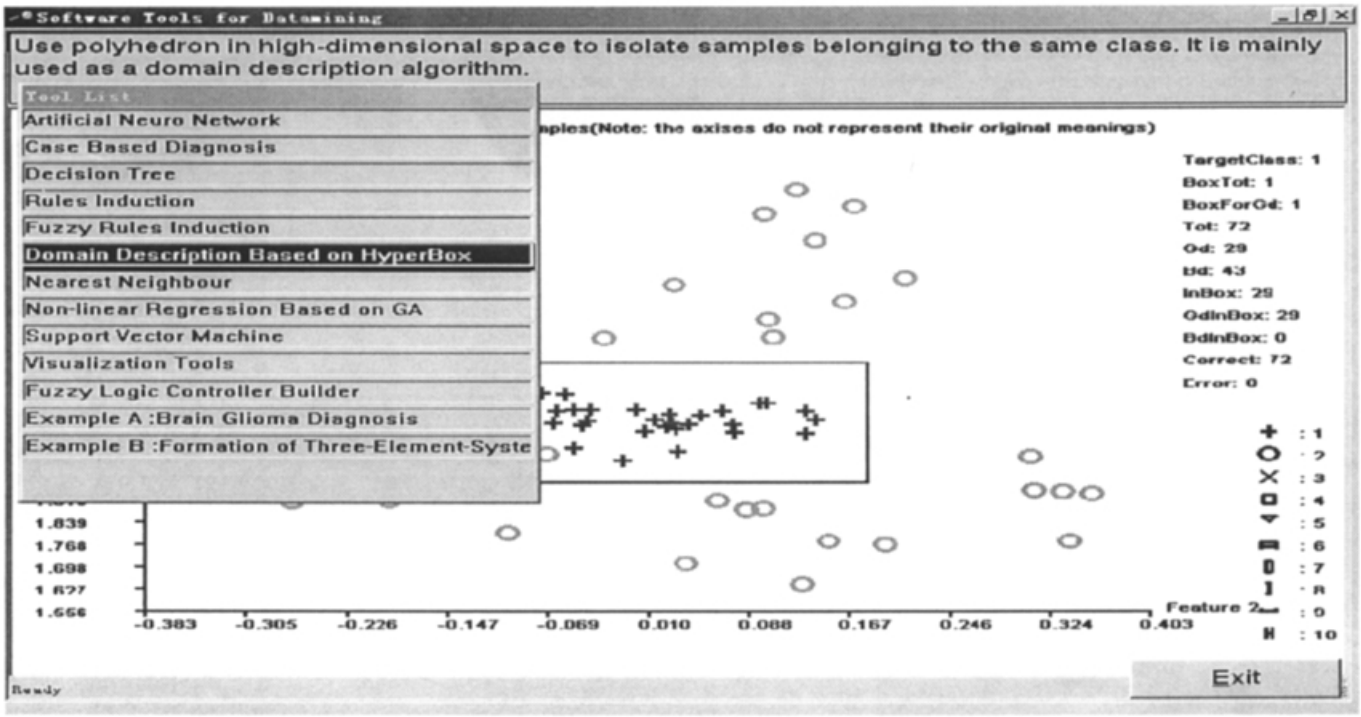

Fig. 1. The interface of the software tool of data mining DMiner-I.

- **Function model of fuzzy rule**: Learning fuzzy rules can be seen as finding the best classifications of fuzzy memberships of input-output variables; multi-layer perceptron network and Min-Max fuzzy neural network are used for automatically learning fuzzy rules.[6] Learning fuzzy rules can also be seen as the combination optimization of input-output fuzzy memberships; genetic algorithms are used for automatically learning fuzzy rules.[6]
- **Function model of neural network**: A multi-layer perceptron network is used as a universal approximator of a nonlinear function, or used for modeling of a classifier. After the user has defined the number of nodes in input layers and in output layers, the function model can design the structure of the multi-layer perceptron network (the number of nodes in hidden layer, initial weights, learning constant etc.) and train the multi-layer perceptron network automatically.
- **Function model of genetic algorithm**: A genetic algorithm is used for optimization. A GA-based algorithm for searching optimal parameters in an n-dimension space is given, which encodes movement direction and distance and searches from coarse to precise. The algorithm can realize global optimization and improve the search efficiency.
- **Function model of HyperEnvelop**: A tool for both classification and data investigation. It employs a series of hyper boxes to isolate samples of the target class from those of other classes.
- **Function model of support vector machine**:[7] An algorithm of a support vector machine is used for modeling of a classifier when the training set of the classifier is not big enough.
- **Function model of visualization**: A component helps analyzers acquire an overview about the data, but also offer an intuitive way to display the mining results.

2. PRINCIPLES OF SOME FUNCTION MODELS

2.1. Function model of decision trees

The generation of decision trees is a top-down process that divides and conquers, which is uncertain because different decision trees are generated if different criteria of division are used. The purpose of the generation of decision trees is to extract rules and to make decisions for future cases. The rate of correctness of decision, and the simplicity and the understandability of rules extracted by a decision tree are influenced by the structure of the decision tree. Under the same rate of correctness of decision, simpler trees are preferable. So the definition of a criterion of division is a key problem in the generation of decision trees, which determines the structure of the generated decision trees.

Existing heuristic criterions (information entropy, Twoing, Gini index, max minority, sum minority) are not always optimal for the generation of decision trees. DMiner-I integrates some heuristic criteria to make use of complementarities of each heuristic criterion, to improve the robustness, correctness and structure of generated decision trees. The following is the algorithm of the generation of decision trees based on the integration n heuristic criteria:

(i) For each node, a relatively optimal division $S_i(1 \le i \le n)$ is searched by n heuristic criteria separately.

(ii) For each division S_i, the degrees of uncertainty $R_{i,j}(1 \le i, j \le n)$ of S_i under jth heuristic criterion are calculated.

(iii) For each $j(1 \le j \le n)$, $R_{i,j}(1 \le i, j \le n)$ are sorted in the sequence from small to large. $Index(i,j)$ represents the sequence-number of $R_{i,j}$, which describes the degree of the optimum of S_i under the jth heuristic criterion. The smaller $Index(i,j)$, the more optimal S_i under jth heuristic criterion.

(iv) For each $i(1 \le i \le n)$, $Sum(i) = \sum_{j=1}^{n} Index(i,j)$ is calculated, which describes the degree of the synthetic optimum under n heuristic criteria.

(v) The division S_i which has the smallest value of $Sum(i)$ is selected as the optimal division under the node of the decision tree.

According to the result (Table I) of the decision trees generated from the database ZZ72,Tttr,Al,Sol collected from some enterprises in China, the integrated heuristic criteria (Gini index, information entropy, max minority) are introduced to improve the structure and classification accuracy of the decision trees induced. All decision trees extracted by the OC1 algorithm based on the integrated heuristic criterion are optimal because they have higher correctness rates and their structures are simpler.

2.2. Function model of rough set

The rough set theory is an important tool to deal with vagueness and uncertainty and is very significant in the field of knowledge discovery and data mining. How to derive the reducts of attribute sets is one of the most important problems of the rough set theory, by which the number of attributes can be reduced and the complexity of the mining task can be decreased without loss of information and sacrifice of the accuracy of mined knowledge; the reducts can also be used for the discovery of rules.

Existing heuristic algorithms for the derivation of reducts are based on the heuristic search algorithm A*. Some algorithms use the importance of attributes as a heuristic function. Some algorithms use the product of the weights of attributes and the frequency of appearance of attributes in the Discernibility Matrix. These algorithms have two drawbacks:

(i) Ease of falling into local optimization: In the derivation of reducts, though all attributes are important, but the reducts composed by these attributes may not be optimal.

(ii) Deriving only one near optimal reduct: For an information system, some reducts which have the same length may be derived. Different reducts compress data from different viewpoints. For different tasks of data mining, different reducts may be involved.

2.2.1. Derivation of reducts based on genetic algorithms. Genetic algorithms (GA) are an effective technique for

https://doi.org/10.1017/S0263574702004307 Published online by Cambridge University Press

Table I. The comparisons of results of decision trees extracted based on different heuristic criteria and the integrated heuristic criterion.

Correctness rate, Structure	Information entropy	Twoing	Gini index	Max minority	Sum minority	Integrated criterion
ZZ72	80.10% 4 layers 8 leaves	80.10% 3 layers 5 leaves	80.10% 4 layers 8 leaves	78.77% 3 layers 5 leaves	80.10% 4 layers 8 leaves	87.00% 3 layers 4 leaves
Tttr	87.80% 6 layers 10 leaves	95.12% 7 layers 13 leaves	97.56% 5 layers 13 leaves	92.68% 2 layers 4 leaves	95.12% 4 layers 15 leaves	97.56% 4 layers 8 leaves
A1	80.77% 7 layers 12 leaves	76.92% 7 layers 15 leaves	69.38% 6 layers 11 1eaves	76.92% 3 layers 5 leaves	80.77% 5 layers 18 leaves	84.62% 6 layers 12 leaves
Sol	90% 7 layers 8 leaves	85.71% 5 layers 11 1eaves	95.24% 8 layers 9 1eaves	80.95% 4 layers 12 leaves	85.71% 3 layers 5 leaves	89.62% 4 layers 9 1eaves

solving complicated optimization problems, they have been applied for the optimization of combinations. In DMiner-I, key techniques in the derivation of reducts by GA are chromosome coding of attribute subset, evaluation of chromosome (reducts) because normal operators of inheritance in GA (selection, crossover, mutation) are used.

For the chromosome coding of reducts, each attribute subset is represented by a binary string. The length of the string is the number of whole attributes. The value 1 in one bit of the string means that the attribute belongs to the attribute subset; the value 0 in one bit of the string means that the attribute does not belong to the attribute subset. For example, the whole attribute set is $\{A1, A2, A3, A4, A5, A6, A7, A8\}$, the chromosome coding of the attribute subset $\{A1, A3, A7, A8\}$, is represented by the binary string 10100011.

The evaluation of chromosome is based on the length of attribute subsets and the degree of dependence between the subset of conditional attributes and the subset of discriminating attributes.

2.2.2. Derivation of reducts based on the "Best-First Search". In order to avoid local optimization in the derivation of reducts, the "Best-First Search" algorithm is used. The difference between the "Best-First Search" algorithm and the algorithm A^* is that during the extending a node, only one best sub-node is searched in the algorithm A^*, but n best sub-nodes ($n>1$) are searched simultaneously in the "Best-First Search" algorithm.

From the comparisons among the experimental results of the derivation of reducts (Table II, the above two algorithms used in DMiner-I are very effective, and can search more than one optimal reducts. In Table II $N(x)$ represents the number of the reducts in length x.

2.3. Function model of HyperEnvelop

HyperEnvelop is a special algorithm proposed in this paper as a tool for both classification and data investigation. It successively generates a series of nested hyper boxes to construct the classifier. The first hyper box (Box 1 in Figure 2), tries to contain all samples of the target class (white dots in Figure 2) without including any sample of the non-target classes (black dots in Figure 2). If it succeeds, HyperEnvelop will be terminated; otherwise (there are samples of non-target classes in the first hyper box), HyperEnvelop will create the second hyper box (Box 2 in Figure 2) that try to

Table II. Comparisons among the experimental results of algorithms.

Database	Number of conditional attributes	Number of samples	Results of the algorithm in [9]	Results of the algorithm based on GA	Results of the algorithm "Best-First Search"
Australian	14	690	N (3)=1	N (3)=5 N (4)=10	N (3)=3 N (4)=4
zoo	16	101	N (3)=1	N (3)=3 N(4)=5	N (3)=2 N(4)=3
Iris	4	150	N(3)=1	N(3)=4	N(3)=4
Lenses	5	16	N (3)=1	N (3)=1	N (3)=1
Monk1	6	432	N(3)=1	N(3)=1	N(3)=1

https://doi.org/10.1017/S0263574702004307 Published online by Cambridge University Press

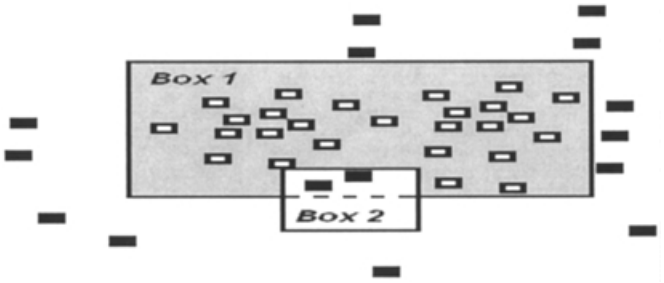

Fig. 2. HyperEnvelop generated to isolate two classes of samples.

"dig out" all the heterogeneous samples without separating any target class sample from the first hyper box. If it succeeds, the process will be terminated; otherwise (the second box contains samples of the target class), the third hyper box will be created to get back from the second hyper box samples of the target class. . . . The process will not be terminated until the samples included by the final hyper box are of the same class or inseparable by any additional hyper box. In this way, HyperEnvelop isolates the region occupied by all the samples of the target class. During classification, a class label of an unknown sample will be decided whether it is inside or outside this region. Unlike traditional classifiers such as MLP, SVM, Nearest Neighbor, decision trees etc., HyperEnvelop offers a closed boundary for a target class, which makes it applicable in a situation (such as human face recognition and signal classification) where the number of possible classes is infinite but only a small number of classes need to be recognized.

In order to cut down the number of unwanted samples existing in a hyper box, we use different methods such as coordinates transformation and genetic algorithm to adjust each face of the box. These methods, in many situations, can obviously decrease the number of hyper boxes generated by HyperEnvelop. As shown in Figure 3, with the help of coordinates transformation and genetic algorithm only one hyper box is needed to isolate samples of the target class. When only the coordinates transformation is adopted, two hyper boxes are generated. However, taking neither of them results in five hyper boxes.

Since the box model is intuitive even when it is in a high dimensional space, we can obtain a basic idea about the distribution of target samples from the construction process of hyper box(es). In order to facilitate this, we offered a 2D projection map to visualize the results of HyperEnvelop. Figure 3 displays a four-feature data set containing samples of two classes. If class 1 ("+") is the target class, HyperEnvelop generates only one hyper box. However, if class 2 ("0") is the target class, two hyper boxes are created, and the first one contains almost all the samples of class 1 as well as all samples of class 2. With HyperEnvelop we have investigated the nine-feature Wisconsin breast cancer dataset (available at the UCI repository of machine learning databases) and found that either benign cases or malignant cases can be included by a single hyper box with no more than 10% samples of the other type inside it.

In addition, we developed an algorithm to grow a decision tree based on the hyper boxes generated by HyperEnvelop. The decision tree employs an oblique hyper plane at each of its non-leaf node, which is similar to OC1 – a well-known decision tree algorithm.[8] However, in our algorithm, oblique hyper planes are generated at the first beginning by HyperEnvelop, while OC1 first grows a univariate decision tree and then adjusts each node with a GA 1ike algorithm to get an oblique hyper plane. Table III displays a performance comparison of different decision tree algorithms on Wisconsin breast cancer dataset mentioned above; records containing missing values are omitted beforehand.

2.4. Function model of support vector machine

Classification is an important issue in data mining. Usually, a classifier is set up in an empirical risk minimization (ERM) way. Taking the multi-layer perceptron network (MLP) as an example, it is not unusual that a well-trained neuron network does not perform in testing so well as it does in training (or in other words, the generalization performance of the network is not so good as it appears to be). These phenomena can be partially explained by the statistical learning theory – a theory dedicated to topics such as the expected error and the sample complexity of various kinds of learning machines.

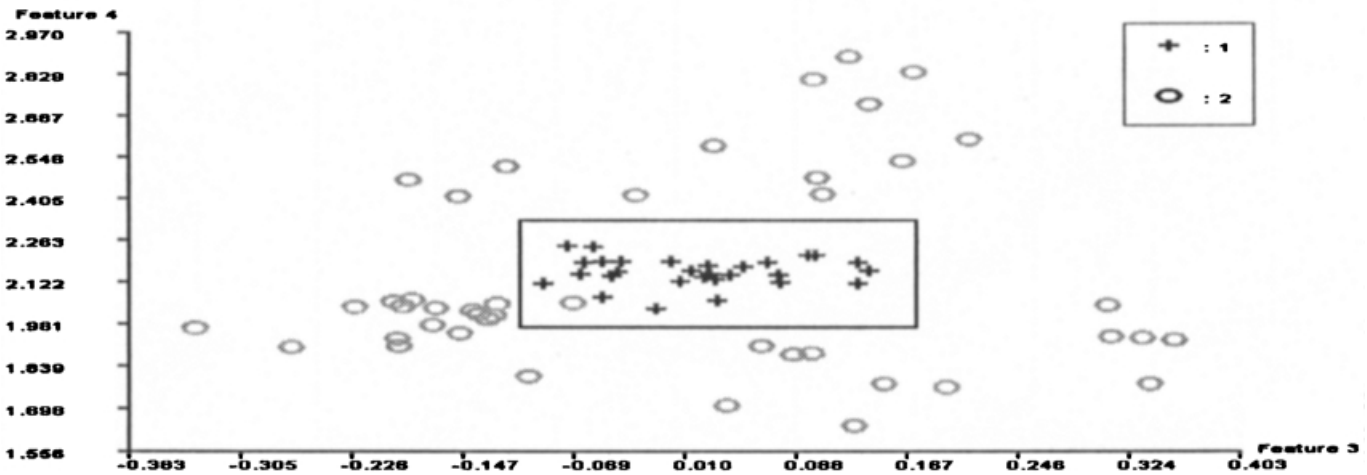

Fig. 3. A hyper box for the samples of class 1 ("+").

https://doi.org/10.1017/S0263574702004307 Published online by Cambridge University Press

Table III. Average number of non-leaf nodes, average testing accuracy and their standard deviations obtained by different decision tree algorithms and a Nearest Neighbor algorithm during a ten-fold cross-validation test on the Wisconsin breast cancer dataset.

	Average number of non-leaf nodes (standard deviation)	Average testing accuracy (standard deviation)
ID3	26.1 (1.1)	94.4% (1.9%)
C4.5	30.2 (2.18)	94.9% (1.6%)
OC1	11.5 (2.1)	94.0% (1.4%)
Our algorithm	15.0 (1.6)	94.6% (0.5%)
Nearest Neighbor	615.0 (0.0)(size of reference set)	96.2% (1.3%)

Based on the statistical learning theory, Vapnik proposed a new way to construct a classifier – the Support Vector Machine (SVM). It constructs an "optimal" hyperplane for classification by quadric programming. Here, the hyperplane can be an ordinary one in the space spanned directly by the features of the data, or an unusual one in a higher dimensional space constructed by the original features and a special kernel function; "optimal" means not only that the empirical risk of the classifier is satisfactory, but also that the structural risk of the classifier family from which the classifier comes is as small as possible (minimization of structural risk results in a better generalization performance). Unlike traditional classifiers such as MLP, decision trees etc., which do not offer an easy way to estimate their expected error SVM provides a theoretical upper bound for this purpose: $E(P_{error}) \leq \frac{E[\text{Number of support vectors}]}{\text{Number of training samples} - 1}$, where E is the mathematical expectation. Because of these and several other advantages, SVM attracts the attention of many researchers in the field of pattern recognition and other related fields. In DMiner-I, we have implemented, using Platt's Sequential Minimal Optimization method (SMO), a SVM for classification and offered, by now, four types of kernel functions:

(i) $k(\vec{x}, \vec{y}) = \vec{x}^T \vec{y}$
(ii) $k(\vec{x}, \vec{y}) = [a(\vec{x}^T \vec{y}) + \mathrm{b}]^p$
(iii) $k(\vec{x}, \vec{y}) = \exp(-\| \vec{x} - \vec{y} \|^2/(2\sigma^2))$
(iv) $k(\vec{x}, \vec{y}) = \tan \mathrm{h}(a(\vec{x}^T \vec{y}) + b)$.

In order to deal with classification problems involving more than two classes, SVM can be used to set up classifiers for each class independently and hence to obtain an array of SVM classifiers. During classification, a class label of the classifier whose output value is the highest will be assigned to the unknown sample. The above process can be done automatically by the "SVM Array" in DMiner-I.

Currently, the function model SVM in DMiner-I is applied in different fields, such as prediction of the degree of malignancy in brain glioma, human face recognition, and signal classification for software radio, and to evaluate its performance thoroughly. The characteristics of SVM were clearly displayed when it was applied to predict the degree of malignancy in brain glioma, in which case the number of training samples was 280 and the number of features used by each sample was fourteen. Compared with MLP trained with BP, the average accuracy achieved by SVM with a linear kernel function was higher (shown in Table IV), running time required by it was much less, and the deviation between its training accuracy and testing accuracy was smaller. Results of our experiments also confirmed the upper bound for the expected error of SVM and revealed that this bound is a little bit conservative (a phenomenon also observed by other researchers).[9]

In the case of human face recognition, the fact that the number of features used (about 30) is even larger than that of training samples for each class (10–20 photos for each person) makes "Over Specialization" or "Over Fitting" a prominent problem to be considered. To obtain a recognizer with high generalization capability, we tried SVM with different kernel functions and obtained a highest testing accuracy when an exponential kernel function was adopted. In fact, even with a linear kernel function the performance of SVM was satisfactory.

With common kernel functions, we have not obtained a satisfactory result in the classification of radio signals. In fact, the design of an appropriate kernel function for a specific problem is still a difficulty in SVM at present. We are going to carry out further research on this issue.

2.4.1. SVM for Regression. Regression is another important issue in data mining. Since "Over Fitting" or bad generalization capability is also a puzzling problem here, the properties of SVM are attractive. In our software, we have implemented, using Smola's Sequential Minimal Optimization method (SMO), a SVM for regression and offered, by now, four types of kernel functions.

Table IV. Average accuracy, standard deviation, highest accuracy and lowest accuracy obtained by different algorithms during a ten-fold cross-validation test on 280 brain glioma cases.

(%)	MLP (4 hidden nodes)	ID3	Nearest Neighbor	FMMNN	SVM
Average accuracy	83.93	81.07	82.50	55.36	84.28
Standard deviation	4.86	5.55	8.37	12.70	6.81
The highest accuracy	92.86	89.29	96.43	78.57	96.43
The lowest accuracy	78.57	71.43	67.86	32.14	71.43

https://doi.org/10.1017/S0263574702004307 Published online by Cambridge University Press

2.5. Function model of visualization

Visualization is an important, if not indispensable, component in the process of data mining. It can help analyzers acquire an overview about the data, and offer an intuitive way to display the mining results. In our software, visualization tools are designed mainly to help the user observe the data in various manners, including projection maps on two or three dimensions (Figure 4 and 5, respectively), value waves on two dimensions, and multi series on time (Figure 6) etc. Users can navigate in the 3D projection map or in the 3D wave graph of values. This kind of navigation is helpful for the feature selection or the choice of data mining tools.

In fact, other algorithms in the software can be used to improve the visualization effects. For example, a noisy dataset can be cleaned to some extent by a data filter before it is visualized, and a high-dimensional dataset can be transformed by a PCA algorithm to make the distribution deviation of different classes prominent in several dimensions (see Figure 7).

3. Nonmonotonic data mining based on concept hierarchy and layered mining

Concept hierarchy is used to describe the taxonomic relations among concepts and their sub-concepts. Concept hierarchy is represented by a tree structure connected by

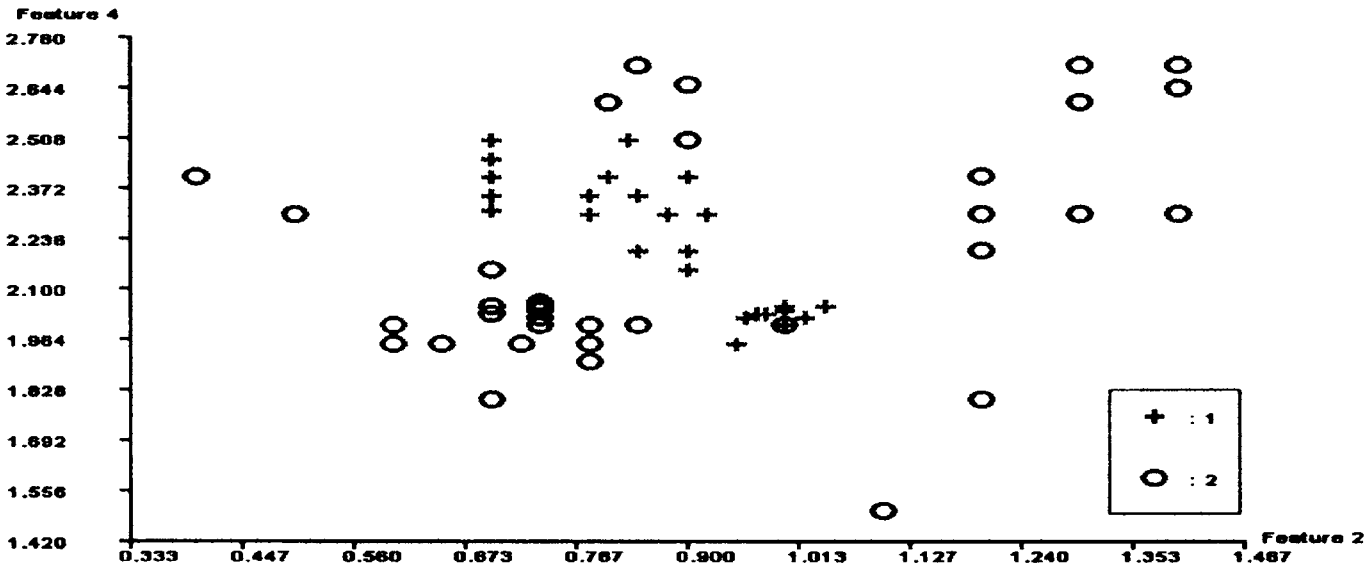

Fig. 4. 2D projection map of a four-feature data set. Feature 2 and feature 4 seem to be important for classification.

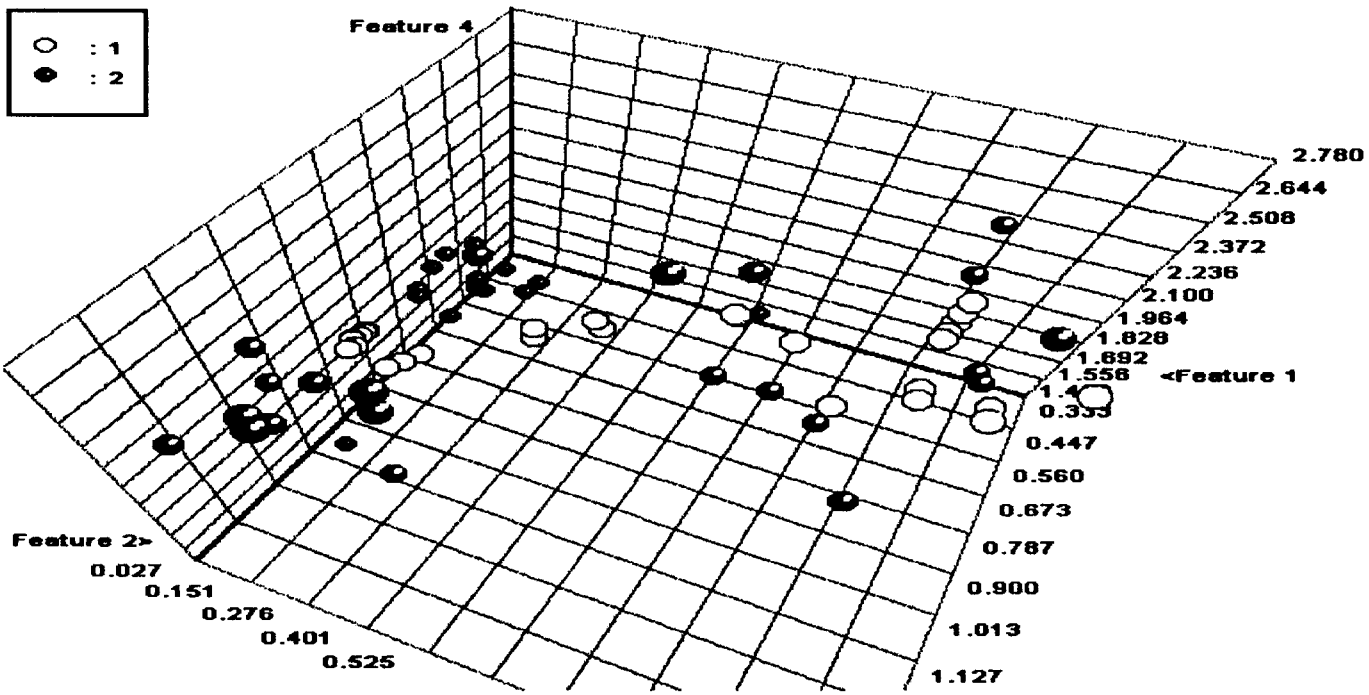

Fig. 5. 3D projection map of the same data set displayed in Figure 4. It gives us a more detailed view about the distribution of the samples. Samples of class 1 are embraced by samples of class 2.

https://doi.org/10.1017/S0263574702004307 Published online by Cambridge University Press

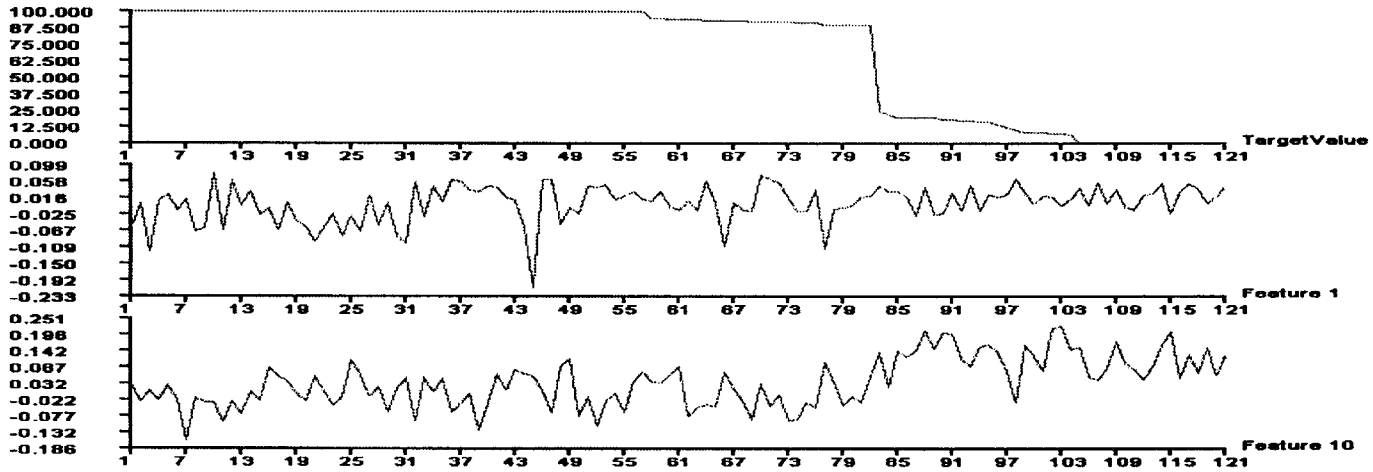

Fig. 6. Multi series on time. The horizontal axis: target value stands for the yield of a real industrial process. When the value of feature 10 goes up, the yield seems to go down.

branch "is-a". If a node (Concept *X*) is connected to its subnode (Concept *Y*) by "is-a", Concept *Y* is a sub-concept of Concept *X*. For example, Concept "*Uni-Student*' is a sub-concept of "*Adult*".

According to concept hierarchy, if rules: $A \rightarrow Y$, $B \rightarrow Y$, $C \rightarrow Y$ are extracted, and Concepts *A*, *B*, *C* are sub-concepts of Concept *D*, then these three rules can be merged into a rule: $D \rightarrow Y$. This merger can make knowledge more understandable and usable. The merger of these three rules means the generalization of knowledge, which may have exceptions. On the contrary, if a rule $A \rightarrow Y$ is extracted, and Concept *B* is a sub-concept of Concept *A*, then another rule $B \rightarrow \neg Y$ may be extracted, because Concept *B* provides more specific information than Concept *A*.

In order to deal with nonmonotony in data mining, when a rule $A \rightarrow Y$ is extracted, all data which are inconsistent with the rule are sorted out to form a new database, then for all sub-concepts $B_1 \ldots, Bn$ of Concept *A* nonmonotonic rules $B_i \rightarrow \neg Y$ $(i=1, \ldots n)$ are tries to be extracted based on concept hierarchy and the new database. Of course, after the nonmonotonic rule $B_i \rightarrow \neg Y$ is extracted, if necessary, nonmonotonic rules may be extracted based on sub-concepts of B_i in concept hierarchy and the revised database.

For example, a rule: *Adult* → *Work* is extracted, all data which are inconsistent with the rule are sorted out to form a new database, all sub-concepts *Uni-Student*, *Teacher*, . . . of Concept *Adult* are searched in the concept hierarchy, a nonmonotonic rule: *Uni-Student* → ¬*Work* is extracted. According to the revised database and the sub-Concept *Night-Uni-Student* of Concept *Uni-Student* a nonmonotonic rule *Night-Uni-Student* →*Work* is extracted.

Nonmonotony in data mining can also dealt with by layered mining.[10] General knowledge is extracted at first,

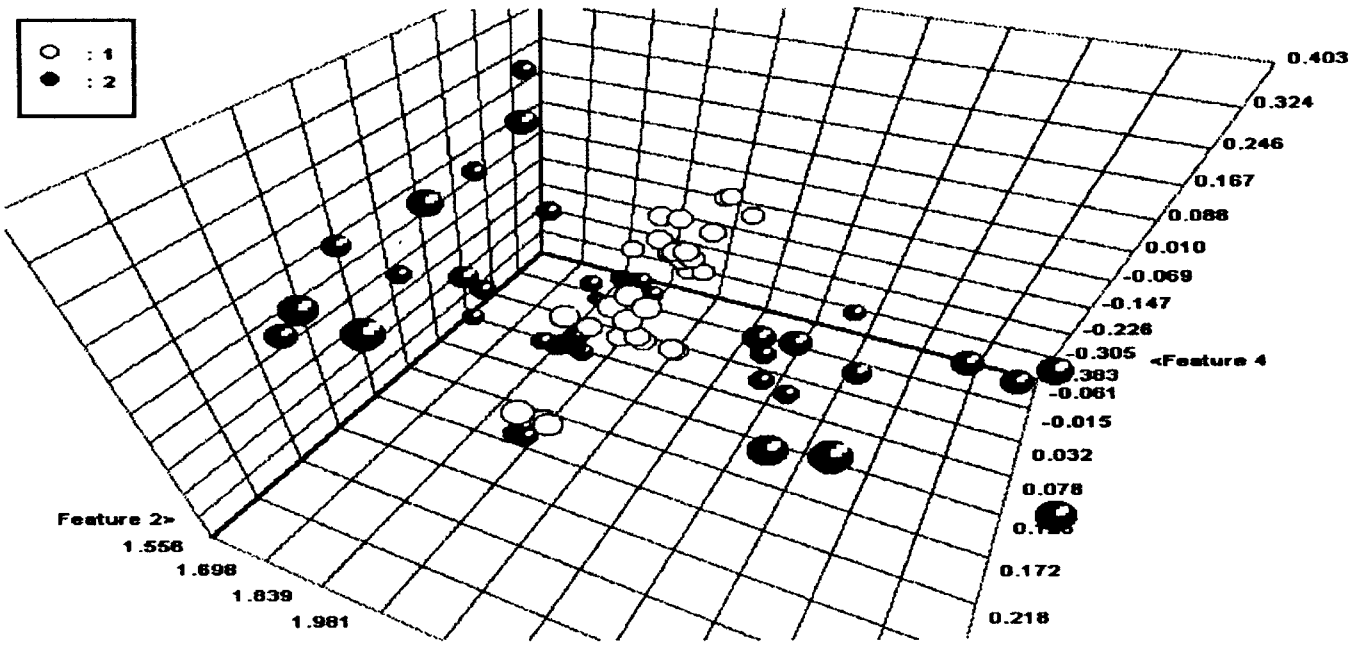

Fig. 7. 3D projection map of the same data Set displayed in Figure 4. The data set is transformed by PCA. Samples of class 1 seem to converge into a cylinder and are meanwhile embraced by samples of class 2. Compared with Figure 5, this figure is more impressive.

https://doi.org/10.1017/S0263574702004307 Published online by Cambridge University Press

then more specific knowledge is extracted. Taking fault diagnosis, for example, diagnostic rules are extracted by a Rough Set from an instance-base. Then all instances that can be satisfied by these default rules are deleted from the instance-base. According to the revised instance-base, neural models based on a multi-layer perceptron network are trained. After neural models are constructed by the instance-base of fault behavior, the instances satisfied by the neural models are deleted from the instance-base. According to the revised instance-base of fault behavior, case-bases about diagnosis of instances in the instance-base can be organized.

4. APPLICATION IN THE PREDICTION OF REGULARITIES OF THE FORMATION OF TERNARY INTERMETALLIC COMPOUNDS

Ternary intermetallic compounds are common alloy phases in many alloys of nonferrous metals. Some ternary intermetallic compounds have been found to be very useful functional materials. Hence the prediction and exploration of new ternary intermetallic compounds are concerned by metallurgists and materials scientists. Traditional methods cannot predict the formation of ternary intermetallic compounds, since the formation of a ternary intermetallic compound depends not only on the chemical affinity between its constituent elements, but also on whether the ternary intermetallic compound or the corresponding binary intermetallic compounds are more stable.

The following independent variables are used as criteria for ternary intermetallic compound formation: Z_1 (number of valence electrons of first element), Z_2 (number of valence electrons of second element), Z_3 (number of valence electrons of third element), R_2/R_1 (atomic radius ratio between the second and first element), R_3/R_1 (atomic radius ratio between the third and first element).

Decision trees are used to predict the formation of ternary intermetallic compounds. For the systems of nontransition elements, the criterion of ternary compound formation is as follows:

$$f(1)=2.154\,Z_1+2.155\,Z_2-3.334\,Z_3+0.356\,(R_2/R_1)-7.222(R_3/R_1)+0.241$$

$$f(2)=0.649\,Z_1-0.587\,Z_2-0.183\,Z_3-0.513\,(R_2/R_1)+0.518(R_3/R_1)-0.504$$

$$f(3)=3.532\times 10^5\,Z_1+10.23\,Z_2-3.532\times 10^5\,Z_3+9.044\,(R_2/R_1)-6.086(R_3/R_1)-7.064\times 10^5$$

$$f(4)=0.012\,Z_1-0.845\,Z_2-2.587\,Z_3+3.347\,(R_2/R_1)+2.364(R_3/R_1)-1.027$$

If $f(1)<0$, $f(2)<0$, then the sample should belong to class "2", i.e., no ternary compound formation.

If $f(1)<0$, $f(2)\geq 0$ and $f(3)\geq 0$, then the sample should belong to class "1", i.e, it should form ternary intermetallic compound.

If $f(1)<0$, $f(2)\geq 0$ and $f(3)<0$, then the sample should belong to class "2".

If $f(1)\geq 0$, $f(4)<0$, then the sample should belong to class "2".

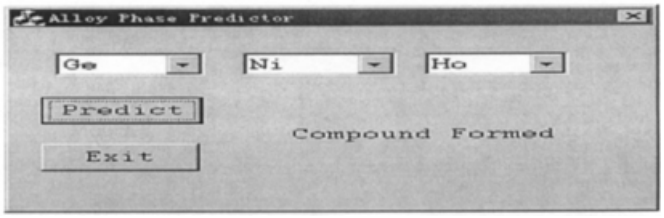

Fig. 8. An interface of "Alloy Phase Predictor"

If $f(1)\geq 0$, $f(4)\geq 0$, then the sample should belong to class "1".

116 ternary alloy systems not included into the training set are used as test samples to test the reliability of the above-listed criterion. It has been found that the rate of correctness of prediction is 94.0%.

An expert system called "Ternary Compound Predictor" has been built based on the criteria extracted by decision trees. It is a module of a software for alloy phase prediction. Figure 8 shows an interface of this software. For example, this expert system predicts that Ge-Pd-Ho, Ge-Ni-Ho, Ge-Pt-Ho, Ge-Ni-Lu and Ge-Ir-Ho systems all form ternary intermetallic compounds and that the AI-Ni-Re system does not form a ternary intermetallic compound. These prediction results have been all confirmed by very recently published experimental results.

5. APPLICATION IN DIAGNOSIS OF BRAIN GLIOMA

The treatment of brain glioma greatly depends on the tumor's degree of malignancy. Nowadays, the main way to assess the grade preoperatively is based on MRI (Magnetic Resonance Imaging) findings and clinical data. However, for most neuroradiologists who have little chance to accumulate enough cases in this field, making a correct judgment is definitely a hard job. To deal with this problem, finding some regular and interpretable patterns to describe the relations between glioma MRI features and the degree of malignancy from abundant cases gathered by a large medical center is desirable. In our application, we proposed a fuzzy rule extraction algorithm based on FMMNN (Fuzzy Min-Max Neural Network) to acquire fuzzy decision rules. The original FMMNN, however, cannot be applied directly to our application due to the following problems: its accuracy is not satisfactory during testing; robustness to data uncertainty is not fully considered. The proposed algorithm is as follows:

Step 1. Expansion of hyperboxes: for every case, according to its type d (high-grade or low-grade) and its numeric presentation $< x_1, x_2, \ldots, x_p >$, detect the nearest and expandable (membership value to it is not less than a predefined threshold θ) hyperbox A_{dk}. If found, expand it; otherwise, initialize a new hyperbox of class d. During the process, if the expansion or initialization of a hyperbox results in either a decrease in E_{fuzzy} or a decrease in classification error (the ratio of correctly classified cases to all) without increase in E_{fuzzy}, the operation will be accepted; otherwise, it will be discarded.

https://doi.org/10.1017/S0263574702004307 Published online by Cambridge University Press

Step 2. Overlap checking: check if there is an overlap between hyperboxes of different classes.

Step 3. Contraction of hyperboxes: contract overlapped hyperboxes one by one. During the process, if the contraction of a hyperbox results in either a decrease in E_{fuzzy} or a decrease in classification error without increase in E_{fuzzy}, the operation will be accepted, otherwise it will be discarded.

Step 4. Additional expansion of hyperboxes: if $< x_1, x_2, \ldots, x_p >$ of a case assigns equal membership values to hyperboxs of different classes, the hyperbox supporting the true type of the case will be expanded a little. Tiny expansions will only occur at the necessary edges that can help to classify the case correctly. During the process, if the expansion of a hyperbox results in either a decrease in E_{fuzzy} or a decrease in classification error, the operation will be accepted; otherwise, it will be discarded. Here, the E_{fuzzy} restriction is looser, because the membership functions (equation 1) is fixed and cannot be modified dynamically.

Step 5. If any hyperbox has been updated during above steps, then go to step 1, else terminate. Unlike FMMNN, iterations are permitted here.

After setting up the classifier, we can obtain the fuzzy rules by translating hyperboxes into linguistic forms. A total of 280 cases of brain glioma are collected, among them 169 are of low-grade gliomas and 111 are of high-grade ones. The resulting fuzzy rules are as follows:

Rule_A1: Age in (1 ~ 53) AND Edema in (Absent, Light) AND Blood Supply in (Normal, A Bit More than Normal) THEN Low-grade glioma.

Rule_A2: Age in (34 ~ 59) AND Mass Effect in (Middle, Heavy) AND Post-Contrast Enhancement in (Heterogeneous) AND Blood Supply in (Affluent) AND Hemorrhage in (Absent, Acute) THEN High-grade glioma.

They achieve an accuracy of 84.64% and can correctly classify 89.94% low-grade cases and 76.58% high-grade ones. According to the above rules, six features such as: "Age", "Edema", "Mass Effect", "Post-Contrast Enhancement", "Blood Supply", and "Hemorrhage" seem to be important. Comparisons on average accuracy, standard deviation, the highest accuracy, and the lowest accuracy are shown in Table V).

The average accuracy of the proposed algorithm is not so good as, but is very competitive to, that of MLP, and is higher and more stable than that of Nearest Neighbor, ID3, and FMMNN. When comes to understandability, the proposed algorithm is evidently superior to the other four. It outputs on the average 2 fuzzy rules, while ID3 uses at least 35 non-leaf nodes and more than 42 leaf ones.

6. APPLICATIONS IN ROBOTICS

Robotic manipulators are highly nonlinear coupled dynamic systems. A robotic manipulator has a complicated mathematical model. It is difficult to design a control system (e.g. mobile robots for reactive navigation) based on the complicated multi-variable nonlinear coupled dynamic model. Intelligent controllers using fuzzy logic do not need a real mathematical model. An adaptive fuzzy control realizes a direct mapping between perceptual situations and control commands in robotic applications. Automatic learning fuzzy rules are a key technique in the realization of fuzzy control. Function models of fuzzy rule (multi-layer perceptron network, Min-Max fuzzy neural network, genetic algorithm) in Dminer-I can be used for automatic learning fuzzy rules. The goal of learning fuzzy rules is that the error between the real outputs of the fuzzy controller to be designed and their expected outputs are reduced as much as possible.

A multi-layer perceptron network can be used for automatic learning fuzy rules because, as an approximation tool, it can realize the data fitting according to the inputs and their expected outputs of the training samples. From the structure of five-layer fuzzy neural network,[6] it can be seen that learning fuzzy rules is to map relations among the combinations of fuzzy predicates of input variables and their corresponding fuzzy predicates of output variables (that is, the weights w_{ij} between fuzzy inference layer and defuzzification layer). Automatically, learning fuzzy rules can be realized by supervised learning in a Min-Max fuzzy neural network. Automatic learning fuzzy rules can be seen as the problem of combination optimization of input-output states. Genetic algorithms can be used for automatic learning of fuzzy rules. Automatic learning of fuzzy rules by a multi-layer feedforward network is simple and fast, but has the problem of a local optimum. Automatic learning fuzzy rules by genetic algorithms can search for a global optimum, but it is complicated and time-consuming. Dminer-I can combine the two approaches to complement each other. Besides the function of automatic learning of fuzzy rules, Dminer-I has also other useful functions: edit and generation of fuzzy membership function and evaluation of fuzzy rules in fuzzy control.

Fault detection and diagnosis play an important role in the operation of autonomous and intelligent robotic systems. Knowledge acquisition and modeling is a key technique in fault detection and diagnosis. Dminer-I can be used for knowledge acquisition and modeling in fault detection and diagnosis. A database or data warehouse of

Table V. Comparisons of accuracy obtained by different algorithms on the 280 glioma cases.

	MLP	ID3	Nearest-Neighbor	FMMNN	Proposed
Average accuracy (%)	84.64	81.07	82.50	55.36	83.21
Standard deviation (%)	4.24	5.55	8.37	12.70	5.31
The highest accuracy (%)	92.86	89.29	96.43	78.57	89.29
The lowest accuracy (%)	73.57	71.43	67 86	32.14	75

https://doi.org/10.1017/S0263574702004307 Published online by Cambridge University Press

data related to fault detection and diagnosis need be constructed for data mining. The function model of association rules in Dminer-I can extract associative rules for fault detection and diagnosis from the database or data warehouse. The function model of a decision tree can extract decision trees for fault detection and diagnosis from the database or data warehouse.

The function model of neural network can construct and train a neural network-based classifier for fault detection and diagnosis from the database or data warehouse. Dminer-I can also combine these function models by layered mining of hierarchical diagnostic models (from general to specific).

Optimization is an important technique in the research of robotics. Genetic algorithms (GA) are an effective technique for solving complicated problem of optimization in robotics, which has been used for collision-free path planning, binocular stereo images matching, task assignment, optimization of the parameters of PID controllers, and configuration for motion planning of a multi-agent-robotic system. The function model of a genetic algorithm in Dminer-I can be used for optimization in robotics. Problems of optimization are represented by chromosomes and solved by operators of inheritance (selection, crossover, mutation) and evaluation of chromosomes For the optimization of real number parameters in a high-dimension space, traditional approaches cannot deal with it efficiently. A GA-based algorithm for searching optimal parameters in a high-dimension space is given in Dminer-I, which encodes movement direction and distance and searches from coarse to precise. The algorithm can realize global optimization and improve search effciency.

Acknowledgements

This research is supported by National Science Foundation of China and National 863 High Tech. Plan of China.

References

1. S. S. Anand, "Designing a Kernel for data mining", *IEEE Expert* **12**(2), 65–74 (1997).
2. J. R. Quinlan, "Decision Trees and Decision making", *IEEE Transaction On Systems, Man, and Cybernetics* **20**(2), 339–346 (1990).
3. A. Chidanand and S. Weiss, "Data mining with decision trees and decision rules", *Future Generation Computer Systems* **13**, 197–210 (1997).
4. Z. Pawlak, *Rough Set, Theoretical Aspects of Reasoning About Data* (Dordrecht, The Netherlands: Kluwer, 1991).
5. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen and A. I. Verkamo, "Fast Discovery of Association Rules", *Advances in Knowledge Discovery and Data Mining* (U. M. Fayyad, ed.) (Morgan Kaufmann, San Mateo, CA). pp. 307–328.
6. J. Yang, Y. Guo and X Huang, "A Software Development System for Fuzzy Control", *Robotica* **18**(4), 375-380 (2000).
7. V. Vapnik, *The Nature of Statistical Learning Theory* (the second edition) (Springer-Verlag, New York, 1998).
8. S. K. Murthy, S. Kasif and S. Salzberg, "System for Induction of Oblique Decision Trees", *Journal of Artificial Intelligence Research* **2**, 1–32 (1994).
9. D. Tax and R. Duin, "Support vector domain description", *Pattern Recognition Letters* **20**, 1191–1199 (1999).
10. J. Yang, C. Ye and X. Zhang, "An Expert system shell for fault diagnosis", *Robotica* **19**(5), 669 674 (2001).