1999 IEEE International Fuzzy Systems Conference Proceedings
August 22-25, 1999, Seoul, Korea

Efficient Fuzzy Rule Generation Based on Fuzzy Decision Tree for Data Mining

Myung Won Kim, Joong Geun Lee, and Changwoo Min
School of Computing, Soongsil University,
1-1, Sangdo-5-Dong, Dongjak-Ku, Seoul, Korea
Tel. +82-2-820-0923 Fax. +82-2-822-3622
E-mail. mkim@computing.soongsil.ac.kr

Abstract

In this paper, we propose an efficient fuzzy rule generation algorithm based on fuzzy decision tree for data mining. We combine the comprehensibility of rules generated based on decision tree such as ID3 and C4.5 and the expressive power of fuzzy sets. Particularly, fuzzy rules allow us to effectively classify patterns of non-axis-parallel decision boundaries, which are difficult to do using attribute-based classification methods.

In our algorithm we first determine an appropriate set of membership functions for each attribute of data using histogram analysis. Given a set of membership functions then we construct a fuzzy decision tree in a similar way to that of ID3 and C4.5. We also apply the genetic algorithm to tune the initial set of membership functions. We have experimented our algorithm with several benchmark data sets. The experiment results show that our method is more efficient in performance and comprehensibility of rules compared with other methods including C4.5.

1. Introduction

With an extended use of computers, we can easily generate and collect data. The human genome database project has collected gigabytes of data on the human genetic code. The NASA Earth Observing System has generated 50 gigabytes of data on the Earth per hour. Because of widespread use of bar codes for most commercial products, it is easy to collect various data on commercial trades. Economic and industrial fields have generated a flood of data. Moreover, advances in database technologies including database management system and data warehousing, have allowed us to store and search large databases fast[1].

The value of these collected raw data is the possibility to extract higher level information which is useful for decision making, exploration, and better understanding of the phenomena described by the data. Traditionally, the human has analyzed the data. However as their volume and dimensionality increase, it becomes hard to analyze the data manually. Therefore there is a need to automatically acquire knowledge from large databases. The subject of Knowledge Discovery in Databases(KDD) is development of automatic and intelligent methods and tools to find useful knowledge from databases.

Data mining is the most important phase in KDD, which finds useful and comprehensible knowledge from the transformed data[1]. Therefore to succeed in KDD, an efficient data mining algorithm for acquiring useful and comprehensible knowledge is needed.

In data mining important factors are efficiency and comprehensibility. The discovered knowledge should well describe the characteristics of the data and they should be easy to understand in order to facilitate better understanding of the data. In this paper, we propose an efficient fuzzy rule generation algorithm called the FDT algorithm for data mining, which integrates the comprehensibility of a decision tree and the expressive power of fuzzy sets. The rules generated by the algorithm have simple conditional parts, they are small in number, and their accuracy is sufficiently high. These characteristics make the generated fuzzy rules more comprehensible.

The paper is organized as follows: in the next section we briefly overview various machine learning algorithms, which can be used for data mining. In section 3, we describe our algorithm: the fuzzy inference method used in our approach, and fuzzy membership function generation based on histogram analysis and using genetic algorithm, followed by fuzzy decision tree construction. In section 4, we describe experiments of our algorithm with a variety of the benchmark data sets. The conclusion section completes our paper.

2. Related Works

Many machine learning algorithms can be applied to data mining. Particularly, inductive learning creates a model for describing data. Decision tree construction and fuzzy rule generation are popular inductive learning methods[2-13]. Many neural network learning algorithms including the Error Back-Propagation(EBP) are a kind of inductive learning. Because the purpose of data mining is to facilitate human's understanding about the data, data mining algorithms should generate not only accurate but also comprehensible knowledge. In this section we survey popular inductive learning algorithms as related to our approach, which can be classified into two categories: decision tree construction and fuzzy rule generation.

2.1 Decision Tree Construction

A decision tree is a useful method for data mining. A decision tree consists of nodes and arcs: non-terminal nodes represent attributes, arcs represent attribute values, and terminal nodes correspond to pattern classes. Each path from the root node to a terminal node can be interpreted as a rule: its antecedent part is a conjunction of conditions in the form of attribute-value pair each of which corresponds to a node and its outgoing arc on the path and its conclusion part is a pattern class corresponding to the terminal node. ID3 and C4.5 proposed by Quinlan are most widely used and they use entropy based on the information theory to construct a compact decision tree[7]. The resulting knowledge, in the form of production rule, has been praised for its comprehensibility. However, the decision tree partitions the whole pattern space into subspaces in an axis-parallel way. If the distribution of data forms non-axis-parallel decision boundaries, it often generates an excessive number of rules to approximate them by axis-parallel hyper-rectangles[1, 7].

2.2 Fuzzy Rule Generation

Fuzzy theory offers a way to overcome such limitations to symbolic decision tree. Fuzzy reasoning is widely applied to complex problems such as pattern recognition and adaptive control. We can represent fuzzy rules linguistically and they can have non-axis-parallel decision boundaries. So fuzzy rules can be comprehensible and accurate.

Recursive fuzzy partitioning algorithms and neuro-fuzzy modeling are widely used for fuzzy rule generation. [9] can be categorized into a recursive fuzzy partitioning method. It divides a fuzzy subspace which contains misclassified training data into its finer fuzzy subspaces. [10] proposes another recursive fuzzy partitioning based on activation regions and inhibition regions to generate fewer and more accurate fuzzy rules. First the activation region is defined as the minimal hyper-rectangle which contains data for one class. The inhibition region is an overlapped region of activation regions. [10] generates fuzzy rules by defining recursively the activation regions in the inhibition regions. Recursive fuzzy partitioning algorithms are simple. But they tend to generate too many rules, and thus it is difficult to understand the fuzzy rules so generated. [11] proposes a fuzzy feed forward network in which input units are associated with fuzzy membership functions while hidden units and output units are associated with fuzzy operators as their activation functions. It uses the gradient descent method for training the fuzzy neural network. [12] proposes a fuzzy rule generation algorithm based on fuzzy associative memory. It generates fuzzy membership functions based on histogram analysis and then it generates fuzzy rules by learning fuzzy relations between input and output data using the fuzzy Hebbian learning method. The fuzzy neural networks proposed by [11, 12] generate relatively fewer fuzzy rules compared to other methods, however, those rules are represented distributedly in connection weights, and thus it is difficult to understand them.

Attempts have been made to integrate the comprehensibility of decision tree and the expressive power of fuzzy sets[8, 13]. However, they have some difficulties such as determining appropriate membership functions for fuzzy sets involved[13] and determining appropriate cut points to partition the range of continuous values of attributes[8]. To solve such problems, Janikow has used rather a complex genetic algorithm[2, 3]. In this paper we propose a FDT algorithm which can generate comprehensible and accurate fuzzy rules. We combine histogram analysis and genetic algorithm for efficiently generating membership functions.

3. The FDT Algorithm

3.1 Fuzzy Inference

We use a simple form of fuzzy rules and inference for better human's comprehensibility. Each fuzzy rule is composed of an antecedent part, which is a conjunction of conditions and a consequent part. Each fuzzy rule also is associated with a *CF*(Certainty Factor) to represent the degree of belief that the consequent is drawn from the antecedent satisfied. Equation (1) is a typical form of fuzzy rules used in our approach.

$$Rule_i : if\ a_{i1}\ is\ U_{i1}\ and\ a_{i2}\ is\ U_{i2} \cdots and\ a_{in}\ is\ U_{in}\ then\ 'class'\ is\ k(CF_i) \qquad (1)$$

In equation (1) a_{ik} represents an attribute and U_{ik} denotes a linguistic term representing a fuzzy set associated with attribute a_{ik}. Application of such a rule to a pattern $X = (x_1, x_2, ..., x_n)$ yields the confidence of class k with which X is classified into class k. In this paper among a variety of fuzzy inference methods we adopt the standard method as follows:

1) *min* is used to combine the degrees of satisfaction of individual conditions of an antecedent;
2) *product* is used to propagate the degree of satisfaction of an antecedent to a consequent;
3) *max* is applied for aggregation of the results of rule applications.

For a given data $X = (x_1, x_2, ..., x_n)$, according to our method the confidence of class k is obtained as

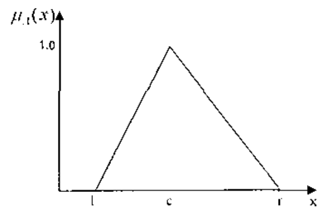
$$Conf_k(X) = \max_{i \in R(k)}(\min_j(\mu_{U_{ij}}(x_j)) \cdot CF_i \qquad (2)$$

In equation (2) $R(k)$ is the set of all rules that classify patterns into class k (their consequent parts are '*class* is *k*'. The class of the maximum $Conf_k(X)$ is the final classification of X.

3.2 Membership Functions

Membership functions are very important in fuzzy rules. Membership functions affect not only the performance of fuzzy rule based systems but also the comprehensibility of rules. In this paper we adopt triangular fuzzy numbers for membership functions and we generate membership functions based on histogram analysis. Our justification to doing this is that more frequently observed attribute values(or value ranges) of data should be more significant than less frequently observed ones in classification. We also use genetic algorithm to automatically generate an appropriate set of membership functions for fuzzy sets as attribute values.

A triangular fuzzy membership function can be represented by a triple of numbers (l, c, r), where l, c, and r represent the left, the center, and the right points of the triangular membership function, respectively(Figure 1).

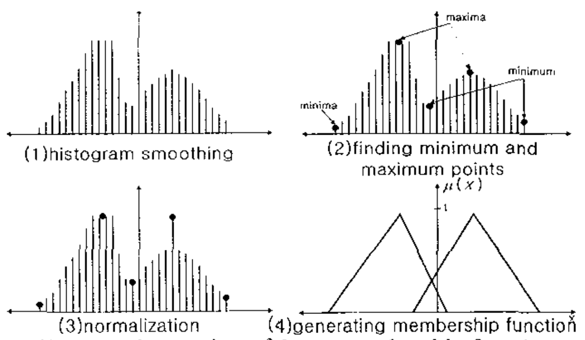


$$\mu_A(x) = \begin{cases} \frac{x-l}{c-l} & : \text{if } l \le x \le c \\ \frac{r-x}{r-c} & : \text{if } c \le x \le r \\ 0 & : \text{otherwise} \end{cases}$$

Figure 1. Triangular fuzzy membership function

3.2.1 Histogram Analysis

To generate fuzzy membership functions, we first calculate the histogram from the training data for each attribute of each class and then we smooth it using the moving average method to remove shallow local minima and maxima points of the histogram. We normalized the smoothed histogram to adjust the heights of the local maxima to the global maximum. Next we generate a fuzzy membership function corresponding to each maximum point of the histogram so that the center point of the triangular fuzzy membership function corresponds to the maximum point and the end points are determined by connecting the maximum point to its nearest local minimum points in both sides. In pattern classification problems it is important to identify regions of critical attribute values determined by minima and maxima points and this method ensures that the critical regions of the histogram are properly mapped into fuzzy membership functions. Figure 2 illustrates our fuzzy membership function generation based on histogram analysis.



maxima
minimum
minima
(1)histogram smoothing
(2)finding minimum and maximum points
$\mu(x)$
1
(3)normalization
(4)generating membership function

Figure 2. Generation of fuzzy membership functions based on histogram analysis.

3.2.2 Genetic Algorithm

Genetic algorithm is an efficient search method simulating natural evolution, which is characterized by the survival of the fittest. In the algorithm potential solutions are a coded by chromosomes and an initial population of chromosomes are generated. The initial population evolves by repeatedly generating a new population applying genetic operations to individual chromosomes of the old population.

The genetic operations are applied to individual chromosomes based on their fitness to the environment in such a way that more fitting chromosomes survive more likely. In our genetic algorithm, a chromosome is of the form (Φ_1, Φ_2, . . . , Φ_h) where i represents the set of membership functions associated with attribute i, which is, in turn, of the form $\{\mu^i_1, \mu^i_2, \dots, \mu^i_k\}$where μ^i_j represents a membership function for attribute i.

Genetic Operations

We have genetic operations such as crossover, mutation, addition and deletion as described in the following.

1) *crossover* : it generates new chromosomes by exchanging the whole sets of membership functions of a randomly selected attribute of the parent chromosomes;
2) *mutation*: randomly selected membership functions in a chromosome are mutated by adding random numbers to the left, the center, and the right points of an individual membership function;
3) *addition*: for a randomly selected attribute, a randomly generated membership function is added to the set of membership functions associated with the attribute;
4) *deletion*: for a selected attribute a randomly selected membership function is deleted.

Addition and deletion operations allow us to generate the appropriate number of membership functions for a given attribute. In addition, in each generation similar membership functions are merged into one.

Fitness Score

If membership functions are determined for each of attributes, we generate an fuzzy decision tree according to the algorithm described in section 3.3. We use the performance of the fuzzy decision tree in fitness evaluation of a chromosome. Suppose a fuzzy decision tree $\tau(i)$ is generated from the membership functions represented by chromosome i. The fitness score of chromosome i is given by $Fit(i)=\alpha\ P(\tau(i))+\beta\ C(\tau(i))$ where $P(\tau(i))$ represents the performance of decision tree $\tau(i)$ and $C(\tau(i))$ represents the complexity of $\tau(i)$, measured in terms of the number of nodes of $\tau(i)$. We also apply the roulette wheel method for selecting candidate chromosomes for genetic operations. For more efficient search we first generate a set of fuzzy membership functions by histogram analysis of training data and introduce it to the initial population for genetic algorithm.

3.3 Fuzzy Decision Tree Construction

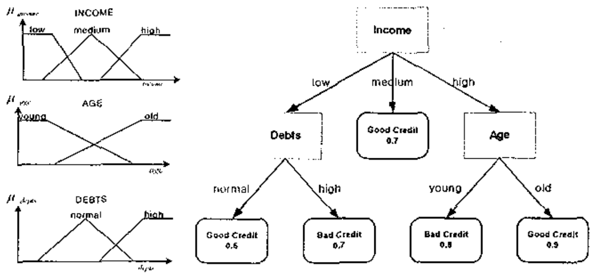


Figure 3. Fuzzy decision tree for credit classification

A fuzzy decision tree is composed of nonterminal nodes, terminal nodes, and arcs. A nonterminal node represents an attribute to partition the pattern space. A terminal node is associated with a class *id* and a *CF*. An arc is associated with a fuzzy set of the attribute corresponding to the parent node.

Figure 3 illustrates a typical form of fuzzy decision trees. In a fuzzy decision tree, each path from the root node to a terminal node corresponds to a fuzzy rule and it also corresponds to a partitioned fuzzy subspace of the whole pattern space. In figure 3, round rectangles(nonterminal nodes) correspond to attributes, angled rectangles (terminal nodes) represent classes, and arcs represent fuzzy sets.

$$v_{il} = \begin{cases} \min_{f \in Z_i}(\mu_f(x_{fl})) : Z_i \neq \phi \\ 1 \qquad\qquad\qquad : Z_i = \phi \end{cases} \tag{3}$$

Equation (3) calculates the membership degree of training data X_l on a fuzzy subspace partitioned by the fuzzy sets from the root node to node *i*. In equation (3), $Z_i = \{f_{i1}, f_{i2}, \ldots, f_{ik}\}$, a class of fuzzy sets each of which is associated with an arc on path P_i from the root node to node *i* and x_{fl} is an attribute value for membership function *f* of training data X_l. In our algorithm, v_{il} represents the membership degree that training data X_l belongs to the fuzzy subspace corresponding to path P_i and it is used to calculate the entropy that measures the degree that the fuzzy subspace contains a single class of patterns.

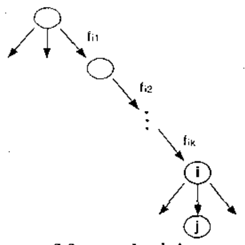


Figure 4. Process of fuzzy decision tree construction.

The entropy of attribute *F* for node *i*, E^i_F is defined as in Equation (4), where *C* is the set of classes, *D* is the set of training data.

$$E^i_F = \sum_j (p_{ij} I^j) \tag{4}$$

$$I^j = -\sum_{k \in C}(p^j_k \log_2 p^j_k) \tag{5}$$

$$p^j_k = \frac{\sum_{m \in k} v_{jm}}{\sum_{l \in D} v_{jl}} \tag{6}$$

$$p_{ij} = \frac{\sum_{l \in D} v_{jl}}{\sum_{l \in D} v_{il}} \tag{7}$$

In the above equations, *j* is a child node of node *i*, corresponding to a fuzzy membership function for an attribute *F*. Equation (5) calculates the information gain for node *j*, equation (6) represents the probability that node *i* represents class *k*, and equation (7) represents the possibility that for a given attribute *F* a training data fits to node *j* compared to other sibling nodes.

Our fuzzy decision tree construction algorithm is as follows.

<step 1>

If one of the following conditions is satisfied, then make node i a terminal node.

(1) $\frac{1}{|D|}\sum_{l \in D} v_{il} \leq \theta_s$

(2) $\frac{\sum_{m \in k^*} v_{im}}{\sum_{l \in D} v_{il}} \geq \theta_d$ where $k^* = \max_{k \in C}(\sum_{m \in k} v_{im})$

(3) no more attributes are available.

In condition (2) class k^* represents the dominant class of the corresponding fuzzy subspace. In this case it is the terminal node whose class is k^* and the associated *CF* is determined by

$$CF = \frac{\sum_{m \in k^*} v_{im}}{\sum_{l \in D} v_{il}}$$

<step 2>

Otherwise,

(1) Let $E^i_{F^*} = \min_F(E^i_F)$.

(2) For each fuzzy membership function of attribute F^*, make a child node from the node associated with F^*

(3) Go to step 1 and apply the algorithm to all newly generated child nodes, recursively.

In the above the threshold parameters θ_s and θ_d determine when we terminate the pattern space partitioning. Condition (1) prohibits generating fuzzy rules for sufficiently sparse fuzzy subspaces while condition (2) prohibits generating fuzzy rules for the minor classes having no sufficient number of patterns in a subspace. In general the fuzzy partitioning method divides the whole pattern space into L^{N_F} subspaces, where N_F is the number of attributes and *L* is the

Table 1. Comparison of classifications rate and the number of generated fuzzy rule for various algorithms

The size of training data	Simple fuzzy grid[9]	Distributed fuzzy if-then rules[9]	CF criterion[9]	NM criterion[9]	RM criterion[9]	Jang[12]	FDT algorithm
21	91.3%(455)	92.4%(8328)	91.1%(253)	89.8%(71)	89.6%(72)	88.3%(32)	91.3%(3)
30	92.7%(1727)	93.8%(20512)	91.8%(307)	93.0%(83)	93.3%(87)	91.6%(36)	95.3%(3)
60	93.5%(2452)	95.4%(63069)	93.8%(307)	93.9%(105)	94.1%(107)	93.3%(46)	96.0%(3)
90	94.5%(3440)	95.8%(140498)	95.1%(528)	94.8%(150)	94.6%(150)	95.0%(46)	96.0%(3)

average number of membership functions associated with an attribute. Each subspace also can have $|C|$ classes of patterns. In this case we may have an excessive number of fuzzy rules to be generated. The threshold parameters θ_s and θ_d are used to control such overfitting in fuzzy rule generation.

For a fuzzy decision tree constructed each path from the root node to a terminal node of the tree corresponds to a fuzzy rule. After constructing a fuzzy decision tree, we determine an appropriate linguistic expression for each fuzzy membership function.

4. Experiments

We have experimented our algorithm with several standard benchmark data sets. They include the IRIS data, the Wisconsin breast cancer data, the credit screening data, the heart disease data, and the sonar data. All of them are obtained from UCI machine learning databases[14]

4.1 The IRIS data

The IRIS data consist of 150 samples of 3 classes with 4 attributes. To construct a fuzzy decision tree we sample randomly the same number of training data for each class. We use all 150 data for test. To analyze classification performance we vary the size of training data to 21, 30, 60, 90 for training. For all experiments we have θ_s and θ_d fixed to 0.1 and 0.7, respectively. Table 1 shows the classification rate and the number of generated fuzzy rules(in parenthesis) of the FDT algorithm and those described in [9, 12] as varying the size of training data.

We notice that the proposed algorithm generates the smaller number of fuzzy rules and its classification rate is higher compared to others. The generated fuzzy rules are as follow:

- if petal(f4) is small(-0.182,0.298,0.653) then setosa(c1) (1);
- if petal(f4) is middle(0.796.1.4,1.95) then vesicolor(c2) (0.84);
- if petal(f4) is large(1.3.2.01.2.65) then virginica(c3) (0.81).

In the above rules, the fuzzy linguistic terms such as small, middle, and large are given by the human.

A rule generated by [9] has its antecedent part, which is a conjunction of conditions associated with input attributes. The antecedent part of rules are unnecessarily complicated and often far more rules than the number of training data are generated. The fuzzy rules generated by [12] is represented distributedly in connection weights which causes difficulties in understanding. However it is easy to understand the fuzzy rules generated by the FDT algorithm because their antecedent parts are simple and they are small in number.

4.2 The Wisconsin Breast Cancer data

The Wisconsin Breast Cancer data consist of data of 2 classes with 10 attributes. The total number of data is 367 and 14 of them have missing values. In our experiment we use only 353 data by discarding those data with missing values. To construct a fuzzy decision tree we use 106 data for class 1 and 94 data for class 2. We use all the 353 data to test the constructed fuzzy decision tree. In this experiment we have θ_s, and θ_d set to 0.2, and 0.7, respectively. The number of generated fuzzy rules is 5 and the classification rate is 91.8%. Figures 5 and 6 show classification rates and the number of generated fuzzy rules as varying θ_s and θ_d, respectively, with the other parameter fixed. We can obtain the desired performance and an appropriate number of rules by controlling the threshold parameters properly.

In the experiments with the IRIS data and the Wisconsin breast cancer data we use membership functions generated from histogram analysis as described in section 3.2.1.

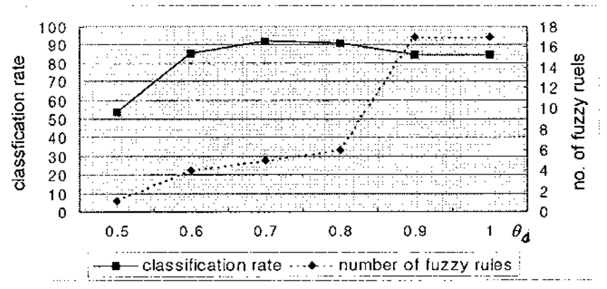


Figure 5. Classification rate and the number of fuzzy rules as varying θ_d with $\theta_s = 0.3$

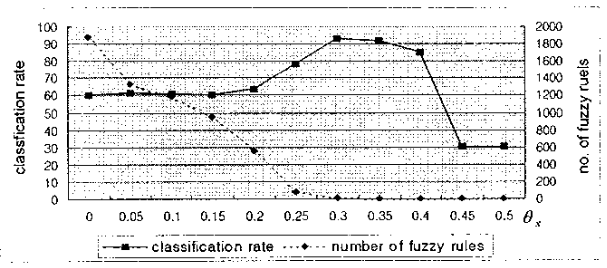


Figure 6. Classification rate and the number of fuzzy rules as varying θ_s with $\theta_d = 0.7$

4.3 More Experiments

In addition, we have experimented our algorithm with several other set of data including the credit screening data, the heart disease data, and the sonar data. In the following we have compared our algorithm with the different versions of C4.5 including Bagged-C4.5[5], Boosted-C4.5[5], and RULES-C4.5[6], which have been proposed as ID3-type algorithms for handling continuous values. In these experiments we use genetic algorithm to generate membership functions. Table 2 compares C4.5[4](release 7 and 8) and the FDT algorithm in error rate and the tree size. Table 3 compares four different versions of C4.5[5, 6] and the FDT algorithm. It is clearly shown that our algorithm is more efficient than any version of C4.5 both in classification rate and the number of rules generated.

Table 2. Comparison of C4.5(release 7 and 8) and the FDT algorithm in error rate and the tree size.

data set	C4.5(Rel.7)[4]		C4.5(Rel.8)[4]		FDT	
	err (%)	Tree size	err (%)	Tree size	err (%)	Tree size
Breast cancer	5.29 ±.09	20.3 ±0.5	5.26 ±.19	25.0 ±0.5	**3.70**	**7**
Credit screening	15.80 ±.30	57.3 ±1.2	14.70 ±.20	33.2 ±1.1	**13.60**	**14**
Heart disease	24.90 ±.40	45.3 ±0.3	23.00 ±.50	39.9 ±0.4	**17.10**	**12**
Iris	4.87 ±.20	9.3 ±0.1	4.80 ±.17	8.5 ±0.0	**4.00**	**4**
Sonar	28.40 ±.60	33.1 ±0.5	25.60 ±.70	28.4 ±0.2	**18.30**	**8**

Table 3. Comparison of different version of C4.5 and the FDT algorithm in error rate and no. of rules

data set	C4.5 [5]	Bagged C4.5[5]	Boosted C4.5[5]	RULES C4.5[6]	FDT
	err (%)	err (%)	err (%)	err(%)/ no. of rules	err(%)/ no. of rules
breast cancer	5.28	4.23	4.09	4.5/ 8.5	**3.70/ 5**
credit screening	14.70	14.13	15.64	15.9/15.0	**13.60/10**
heart disease	22.94	21.52	21.39	23.1/11.1	**17.10/ 8**
Iris	4.80	5.13	6.53	4.7/ 4.1	**4.00/ 3**
sonar	25.62	23.80	19.62	31.1/ 7.0	**18.30/ 7**

5. Conclusions

In this paper, we propose an efficient fuzzy rule generation algorithm based on fuzzy decision tree for data mining. Our method provides the efficiency and the comprehensibility of the generated fuzzy rules, which are important to data mining. Particularly, fuzzy rules allow us to effectively classify patterns of non-axis-parallel decision boundaries using membership functions properly, which is difficult to do using attribute-based classification methods. We also apply the genetic algorithm to evolve the initial set of membership functions generated from histogram analysis. The experiment results show that our method is more efficient in performance and comprehensibility of rules compared with other methods including C4.5.

We are now applying our algorithm for generating fuzzy rules describing the blast furnace operation in steel making. Our preliminary experiments have shown that the algorithm is powerful for many real world applications. Future work is to develop a fuzzy decision tree pruning algorithm to generate a more compact fuzzy decision tree. We also plan to work for improving the algorithm to be able to handle a large volume of data efficiently, which is often an important requirement to data mining.

References

[1] Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., From Data mining to Knowledge Discovery: An Overview, in *Advances in Knowledge Discovery and Data Mining*, Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., pp. 1-34, MIT Press, 1996,

[2] C. Z. Janikow, Fuzzy Decision Tree : issues and methods, *IEEE Transactions on Systems, Man and Cybernetics – PART B : Cybernetics*, vol. 28 no. 1, pp.1-14, 1998

[3] C. Z. Janikow, A Genetic Algorithm Method for Optimizing the Fuzzy Component of a Fuzzy Decision Tree, In *GA for Pattern Recognition*, S. Pal & P. Wang(eds.), CRC Press, pp.253-282, 1996

[4] Quinlan, J. R., Improved use of continuous attributes in C4.5, Journal of Artificial Intelligence Research, vol. 4, pp.77-90, 1996

[5] Quinlan, J. R., Bagging, boosting, and C4.5, Proceedings 13th American Association for Artificial Intelligence National Conference on Artificial Intelligence, pp.725-730, AAAI Press, Menlo Park, CA, 1996

[6] Quinlan, J. R., MDL and categorical theories (continued), Proceedings Twelfth International Conference on Machine Learning, pp.464-470, Morgan Kaufmann, Montreal, 1995

[7] Quinlan, J. R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, 1993

[8] J. Zeidler, M. Schlosser, Conitnuous-Valued Attributes in Fuzzy Decision Trees, *Proc. Of the 6-th Int. Conf. On Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pp.395-400, 1996

[9] Ishibuchi, H., Nozaki, K., Tanaka, H., Effective fuzzy partition of pattern space for classification problems, *Fuzzy Sets and Systems*, vol. 59, pp.295-304, 1993

[10] Abe, S., Lan, M.-S., A Classifier Using Fuzzy Rules Extracted Directly from Numerical Data, *Proceedings of 2nd IEEE Conference on fuzzy Systems*, pp.1191-1198, 1993

[11] Rhee, F. C., Krishnapuram, R., Fuzzy rule generation methods for high-level computer vision, *Fuzzy Sets and Systems*, vol. 60, pp.245-258, North-Holland, 1993

[12] Jang, D-S., Choi, H-I., Automatic Generation of Fuzzy Rules with Fuzzy Associative Memory, In *Proc. of the ISCA 5th International Conference*, pp.182-186, 1996

[13] Umanno, M., Okamoto, H., Hatono, I. et al, Fuzzy Decision Trees by Fuzzy ID3 Algorithm and Its Application to Diagnosis Systems, In *Proc. of 3th IEEE International Conference on Fuzzy Systems*, pp.2113-2118, 1994

[14] http://www.ics.uci.edu/~mlearn/MLRepositoy .html