

A METHOD OF GENERATING RULES FOR A KERNEL FUZZY CLASSIFIER

AI-MIN YANG^{1,2}, XIN-GUANG LI¹, LING-MIN JIANG¹, YONG-MEI ZHOU¹, QIAN-QIAN LI³

¹School of Informatics, Guangdong University of Foreign Studies, Guangzhou 510420, China

²School of Computer, National University of Defense Technology, Changsha 410073, China

³School of Computer and Communication, Hunan University of Technology, Zhuzhou Hunan, 412008, China

E-MAIL: amyang18@163.com

Abstract:

A method of generating rules for a kernel fuzzy classifier is introduced. For this method, firstly, the initial sample space is mapped into a high dimensional feature space by selecting the appropriate kernel function. Then in the feature space, the proposed dynamic clustering algorithm dynamically separates the training samples into different clusters and finds out the support vectors of each cluster. For each cluster, a fuzzy rule is defined with ellipsoidal regions. Finally, the rules are adjusted by Genetic Algorithms. This classifier with such fuzzy rules is evaluated by two typical data sets. For this classifier, the learning time is short, the classification accuracy is better and the speed of classification is quick.

Keywords:

Fuzzy classification rule; Kernel function; Dynamic clustering; Genetic algorithms

1. Introduction

Fuzzy classification is an important application of fuzzy set theory. Fuzzy classification rules [1-6] are widely considered a well-suited representation of classification knowledge, with readability and interpretation. Fuzzy classification has been widely applied in many fields, such as, image processing, words recognition, voice recognition, text classification, remote sensing etc. But there are some problems in generating and optimizing fuzzy classification rules, and in how to select membership function, that is, it is difficult to gain fuzzy rules with good performance. So, there are many researchers on studying these problems and developing variety of systems, which were based on neural network [1], genetic algorithm [2-3,6], clustering [4-5] and so on. But fuzzy classification systems based on neural network and genetic algorithm have long training time and too many rules, when the dimensions of input samples or the number of class is many. In the system based on clustering, it is difficult to optimize the rules. Especially, when the distribution of samples is as Figure 1(a), the general clustering method has poor classification accuracy.

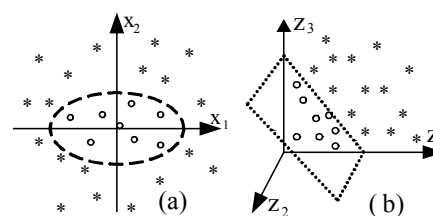


Figure 1. A distribution of samples

In order to train classifiers fast and obtain well generalization ability as neural network classifier, we propose a method of generating and optimizing fuzzy rules and construct a new fuzzy classifier with these rules, called kernel fuzzy classifier. For this method, firstly, the initial sample space is mapped into a high dimensional feature space by selecting appropriate kernel function, as Figure 1, $(x_1, x_2) \rightarrow (z_1, z_2, z_3)$, where, $z_1 = x_1^2$, $z_2 = 2^{1/2}x_1x_2$, $z_3 = x_2^2$, and the dimension of sample space becomes from two to three. Then in the feature space, the proposed dynamic clustering algorithm dynamically separates the training samples into different clusters and finds out the support vectors of each cluster. For each cluster, a fuzzy rule is defined with ellipsoidal regions. Finally, the rules are adjusted by Genetic Algorithms.

This paper organizes as follows. In section 2, the basic form of kernel fuzzy classification rules is introduced. Section 3 presents the method of computing support vectors. Section 4 describes the proposed dynamic clustering algorithm. In section 5, the method of adjusting rule is introduced. Section 6 is performance evaluation for proposed classifier. The last is conclusion and future work.

2. Basic form of kernel fuzzy classification rules

Let R^q denote the sample space (q is dimension of sample space), $\Phi: R^q \rightarrow F$, $x \rightarrow \Phi(x)$, F is a higher dimension space relative to R^q , called feature space. Φ is a map function. x is the sample in initial space, $\Phi(x)$ is the

samples in feature space and F . Let $K(\cdot)$ denote kernel function[11]. Its definition is as Eq.(1).

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (1)$$

Where, “ \cdot ” is the inner product. In this paper, the kernel function of as Eq.(2) is used.

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (\gamma > 0) \quad (2)$$

Fuzzy rule based on ellipsoidal regions can be described as follows. Let x ($x \in R^q$) and $\Phi(x)$ ($\Phi(x) \in F$) be an input sample in the initial space and in feature space respectively. The class number of the classified patterns is n . C_i ($i=1, \dots, n$) denotes the i -th class. Each class can be clustered with the increase of training samples, C_{ij} ($j=1, \dots$) denotes the j -th cluster of the i -th class. When the samples number of a cluster increases to a given value, a new fuzzy classification rule is generated and the expression is as (3).

$$R_{ij} : \text{IF } \Phi(x) \text{ is around } C_{ij} \text{ THEN } x \in C_i \quad (3)$$

$$\text{with } CF = \alpha_{ij}$$

Where, R_{ij} is the label of rule and generated from the j -th cluster of i -th classification, CF represents the degree of $\Phi(x)$ belonging to this rule, $\alpha_{ij} \in [0, 1]$ is described as Eq(8).

Let $\Phi(cc_{ij})$ ($cc_{ij} \in R^q$) denote the center of the j -th cluster of i -th classification. $mf_{ij}(\Phi(x))$ denotes the membership function for the input sample. $u_{ij}(\Phi(x))$ is the degree of membership function. The following equations are defined.

$$mf_{ij}(\Phi(x)) = \exp(-h_{ij}^2(\Phi(x))) \quad (4)$$

$$h_{ij}^2(\Phi(x)) = \frac{d_{ij}^2(\Phi(x))}{\delta_{ij}} \quad (5)$$

$$d_{ij}^2(\Phi(x)) = \|\Phi(x) - \Phi(cc_{ij})\|^2 \quad (6)$$

$$= K(x, x) - 2K(x, cc_{ij}) + K(cc_{ij}, cc_{ij})$$

$$u_{ij}(\Phi(x)) = mf_{ij}(\Phi(x)) \quad (7)$$

$$\alpha_{ij} = u_{ij}(\Phi(x)) \quad (8)$$

Where, the membership function $mf_{ij}(\Phi(x))$ is a typical exponential function like Gauss function. $h_{ij}(\Phi(x))$ is called adjusting distance. $d_{ij}(\Phi(x))$ is the Euclidean distance between $\Phi(x)$ and the center of C_{ij} cluster. $\Phi(cc_{ij})$ is determined with our proposed dynamic cluster algorithm in setion 4. δ_{ij} is a adjusting parameter. Eq(6) is calculated according to Eq(14) (in selection 3) after created rules.

3. The method of finding support vectors

Suppose C_{ij} is a cluster with n_{ij} samples ($s=1, 2, \dots, n_{ij}$). The selected mapping function Φ and kernel function satisfy Eq.(1) and Eq.(2) (in selection 2). There are the following questions to be solved. (1) how to compute the minimal radius R_{ij} of C_{ij} . (2) how to compute the distance $d_{ij,L}^2$ between $\Phi(x_L)$ ($L \neq s$) and the center of C_{ij} .

If sample $\Phi(x_s)$ belongs to cluster C_{ij} , inequation (9) holds.

$$\|\Phi(x_s) - \Phi(cc_{ij})\|^2 \leq R_{ij}^2 \quad (9)$$

Where, $\|\cdot\|$ is the standard Euclidean distance, $\Phi(cc_{ij})$ is the center of C_{ij} . We calculate the minimal value of R_{ij}^2 in Eq(9) by using Lagrangian equation (as Eq(10)).

$$L = R_{ij}^2 - \sum_{s=1}^{n_{ij}} (\beta_s (R_{ij}^2 - \|\Phi(x_s) - \Phi(cc_{ij})\|^2)) \quad (10)$$

Where, β_s ($\beta_s \geq 0$) is the lagrangian multiplier. To get the minimal value of L , R_{ij} and $\Phi(cc_{ij})$ have to satisfy the constraint condition Eq.(11) and Eq.(12).

$$\sum_{s=1}^{n_{ij}} \beta_s = 1 \quad (11)$$

$$\Phi(cc_{ij}) = \sum_{s=1}^{n_{ij}} \beta_s \Phi(x_s) \quad (12)$$

The variables of R_{ij} and $\Phi(cc_{ij})$ in Eq(10) can be canceled using the above equations. Eq.(10) is translated to the Wolfe dual[8] problem as Eq.(13).

$$\begin{aligned} W &= \sum_{s=1}^{n_{ij}} \Phi(x_s)^2 \beta_s - \sum_{s=1, s'=1}^{n_{ij}} \beta_s \beta_{s'} \Phi(x_s) \cdot \Phi(x_{s'}) \\ &= \sum_{s=1}^{n_{ij}} K(x_s, x_s) \beta_s - \sum_{s=1, s'=1}^{n_{ij}} \beta_s \beta_{s'} K(x_s, x_{s'}) \end{aligned} \quad (13)$$

Where, W is the function about β_s , its max value can be got using quadratic programming as well as β_s of each sample $\Phi(x_s)$, and these values make Eq(10) getting minimal value. The set of samples $\Phi(x_s)$ ($s=1, 2, \dots, n_{ij}$) is separated to two types according to the value of β_s . One with $\beta_s=0$ is in the inner of the sphere with radius R_{ij} (some are outside of the sphere), while the other with $\beta_s>0$ is on the border of the sphere, called the support vectors [7-8]. n_{ij_sv} is the number of support vectors of cluster C_{ij} , $n_{ij_sv} \leq n_{ij}$. So, in feature space, the distance between any sample $\Phi(x_L)$ and the center of C_{ij} can be calculated using Eq.(14), and R_{ij} of C_{ij} can be calculated according to Eq.(15).

$$d_{ij,L}^2(x_L) = \|\Phi(x_L) - \Phi(cc_{ij})\|^2$$

$$= K(x_L, x_L) - 2 \sum_{s=1}^{n_{ij,sv}} \beta_s K(x_s, x_L) \quad (14)$$

$$+ \sum_{s,s'}^{n_{ij,sv}} \beta_s \beta_{s'} K(x_s, x_{s'})$$

$$R_{ij}^2(x_k) = K(x_k, x_k) - 2 \sum_{s=1}^{n_{ij,sv}} \beta_s K(x_s, x_k)$$

$$+ \sum_{s,s'}^{n_{ij,sv}} \beta_s \beta_{s'} K(x_s, x_{s'}) \quad (15)$$

Where, x_k is one support vector of cluster C_{ij} . So, the distance between sample and the center of cluster and the radius of the cluster are calculated according to Eq.(14) and Eq.(15).

4. Dynamic clustering algorithm

We put forward a dynamic clustering algorithm to fast generate fuzzy classification rules. In feature space F , this algorithm can separate the training samples into different clusters by scanning the training samples once. Let C_{ij}^k represent C_{ij} which has been modified for k times ($k=0,1,2,\dots$). $\Phi(cc_{ij}^k)$ is the center of the cluster C_{ij} which has been modified for k times. Ru_{ij}^k is Ru_{ij} which has been modified for k times. $d_{ij,L}$ is the distance between input sample x_L and the center of C_{ij} . dth_i is the radius threshold for each cluster of C_i .

Dynamic clustering algorithm:

Input: training samples (x_L, C_i) ($L=1,2,\dots,m$, $i=1,2,\dots,n$), the selected kernel function which satisfies Eq.(1) and Eq.(2), dth_i ($i=1,2,\dots,n$).

Output: the radius of each cluster, support vectors.

Algorithm:

Step 1: For each input training sample, judge the class, which this sample belongs to (do clustering for the samples in the same class).

Step 2: If a sample is the first training sample of C_i ($i=1,\dots,n$), C_{i1} is created and this sample becomes the cluster center of C_{i1} , and the following symbols are given, C_{i1}^0 , $\Phi(cc_{i1}^0)$, Ru_{i1}^0 , and $Ru_{i1}^0=0$, else, go to step 3.

Step 3: If training sample $\Phi(x_L)$ belongs the C_i while it is not the first one, calculate the distances $d_{ij,L} = \|\Phi(x_L) - \Phi(cc_{ij})\|$ (calculating according to Eq.(14)), which is between $\Phi(x_L)$ and created cluster centers of C_{ij} ($j=1,\dots,cn_i$). Here, cn_i is the number of created clusters of C_i .

Step 4: Compare the $d_{ij,L}$ which is calculated in Step 3

with the corresponding cluster radius Ru_{ij} . If there is the cluster C_{ig} ($1 \leq g \leq cn_i$) which satisfies $\|\Phi(x_L) - \Phi(cc_{ig})\| = \min(\|\Phi(x_L) - \Phi(cc_{ij})\|)$, and $d_{ij,L} \leq Ru_{ij}$ ($j=1,2,\dots,cn_i$), $\Phi(x_L)$ is assigned into C_{ig} , then go to Step 1, else, go to Step 5.

Step 5: Find the cluster C_{ia} which is nearest to $\Phi(x_L)$ among the created clusters of C_i . Firstly, calculate $S_{ij,L}$, $S_{ij,L} = d_{ij,L} + Ru_{ij}$ ($j=1,\dots,cn_i$), then choose C_{ia} according to $S_{ij,L}$ and Eq.(16).

$$S_{ia,L} = d_{ia,L} + Ru_{ia} = \min\{S_{ij,L}\} \quad (16)$$

$$(j = 1, 2, \dots, cn_i)$$

Step 6: If $S_{ia,L} > 2 \times dth_i$, it indicates that $\Phi(x_L)$ doesn't belong to any created clusters, then a new cluster should be created, the creating method and process are similar to those of Step 2, then go on Step 1, else, go to Step 7.

Step 7: If $S_{ia,L} \leq 2 \times dth_i$, the center (cc_{ia}) and radius (Ru_{ia}) of C_{ia} should be updated by the method in section 3, and the support vectors refreshed also. If all the input training samples have been input, turn to Step 8, else, turn to Step 1.

Step 8: For every cluster, output the radius and support vectors.

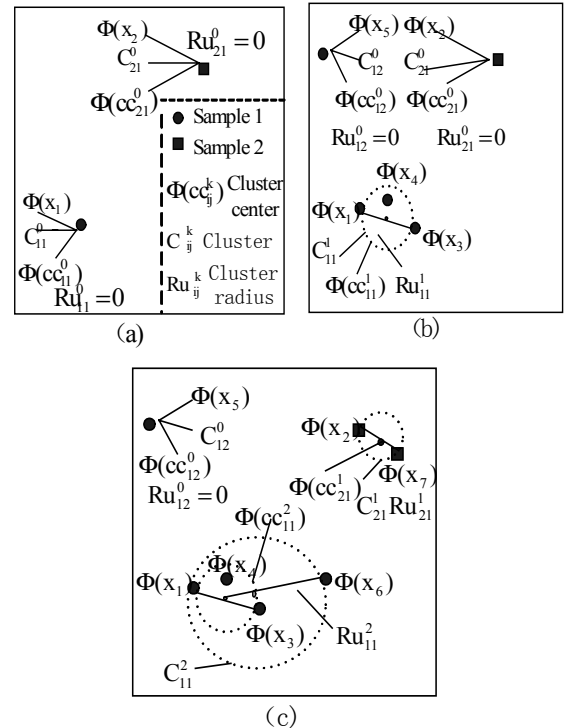


Figure 2. Illustration of dynamic clustering

Figure 2 shows the process of the dynamic clustering algorithm. In Figure 2(a), the training samples $\Phi(x_1)$ and $\Phi(x_2)$ belong to two different classes, and create clusters C_{11} and C_{21} respectively. In Figure 2(b), sample $\Phi(x_3)$

satisfies the condition in Step 7, so the center and radius should be modified, cc_{11}^1 and Ru_{11}^1 are generated, sample $\Phi(x_4)$ satisfies the condition in Step 4, the cluster center and radius need not to be modified, sample $\Phi(x_5)$ satisfies the condition in Step 6, and cluster C_{12} is created. In Figure 2(c), $\Phi(x_6)$ and $\Phi(x_7)$ are similar to the sample $\Phi(x_3)$.

Fuzzy classification rules are generated during the process of dynamic clustering, each cluster generates a rule as expression (3) when the number of training samples for this cluster is over set value (default is $2 \times q$). Rule R_{ij} is determined by following parameters: the support vectors, Lagrange Multiplier β_s , and the adjusting parameter δ_{ij} . After the creation of rules, rules should be adjusted because the overlapping of rule regions is unconsidered when rules were generated. This is discussed in section 5.

The samples can be classified after the rules adjusted. CF can be calculated corresponding to each input sample $\Phi(x)$. If the value α_{kj} of the rule kj is the most, x can be classified to the k -th class.

5. Method of adjusting rule by genetic algorithms

Because the overlapping of rule regions is unconsidered during the generation of rules, these rules should be adjusted. In this selection, we propose a method of adjusting rule by Genetic Algorithms. A rule is mainly determined by the support vectors, β_s , and δ_{ij} . If all of these parameters are adjusted, it will consume much time [6]. Here, we only adjust the value of δ_{ij} by Genetic Algorithm to get the maximal recognition rate [9]. According to Eq.(6), when the value of δ_{ij} increases, the degree of membership function will increase, vice versa.

5.1. Gene coding

Genetic information of fuzzy rule R_{ij} is δ_{ij} and the firing information. A chromosome (see Figure 3(a)) consists of δ_{ij} and firing information. Each training sample is normalized to $[0, 1]$, and $0 < \delta_{ij} \leq 2$. δ_{ij} is expressed by a 8-bit binary string and firing information is expressed by a bit. Figure 3(b) shows the composition of an individual. Figure 4 shows the composition of a population. I_p ($p=1, 2, \dots, P$) denotes an individual, P denotes the size of this population.

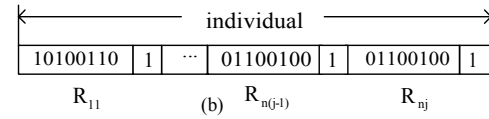
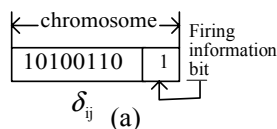


Figure 3. Component of chromosome and its individual

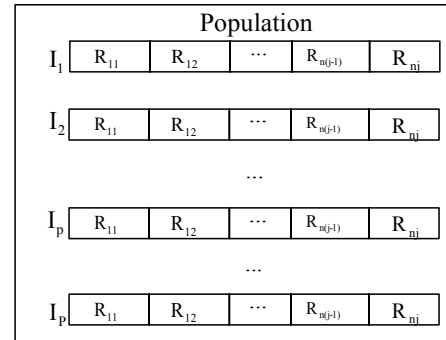


Figure 4. Component of a population

5.2. Fitness function

For each individual, the fitness function is Eq.(19). The recognition rate of individual I_p for training samples can be calculated according to Eq.(17). In Eq.(17), Num_{all} is the total number of training samples and Num_{cop} is the number of training samples which have been classified correctly. Eq.(18) is the average classification accuracy of rule R_{ij} . In (18), $\Phi(x_L)$ is the training samples which have been classified correctly, N_{corr} is the number of training samples which have been classified corrected by rule R_{ij} and N_{ncorr} is the number of training samples which have been classified incorrectly by rule R_{ij} . Eq.(19) is the fitness function of individual I_p . In Eq.(19), n is the number of class and N_{R_all} is the total number of created rules.

$$Accuracy = \frac{Num_{cop}}{Num_{all}} \quad (17)$$

$$ACF_{R_{ij}} = \frac{\sum_{L=1}^{N_{corr}} u_{ij}(\Phi(x_L))}{N_{corr} + N_{ncorr}} \quad (18)$$

$$Fit_p = Accuracy \times \frac{\sum_{i=1}^n \sum_{j=1}^n ACF_{R_{ij}}}{N_{R_all}} \quad (19)$$

5.3. Determination of firing information

In formula (20), $ACC(R_{ij})$ is the correction rate of classification of rule R_{ij} for training samples, $Trnum_{corr}(R_{ij})$ is the number of training samples which are classified correctly by rule R_{ij} , $Trnum_{all}(R_{ij})$ is the total number of the training samples classified by rule R_{ij} , and k is general in the range from 0.6 to 0.9. In each generation, if formula (20) is satisfied, rule R_{ij} will be fired.

$$ACC(R_{ij}) = \frac{Trnum_{corr}(R_{ij})}{Trnum_{all}(R_{ij})} > k \quad (20)$$

Each rule maintains a count for unfired information. $nfc(R_{ij})$ is the count for R_{ij} . NFC is a threshold of this count. During the evolving procedure of the rules, if the rule is not fired, $nfc(R_{ij})$ will be increased by 1. As soon as $nfc(R_{ij})=NFC$, rule R_{ij} will be deleted, and these training samples for R_{ij} are united with the rule which is the nearest to deleted rule R_{ij} . This strategy can delete some rules and make the number of rules more reasonable.

5.4. Genetic operation

The procedure of genetic operation is as follows.

Step 1: Initial population is generated. Generally, this process is carried out randomly and all firing bits of rules are 1.

Step 2: Evaluate the fitness of each individual I_p in the population as Eq.(19).

Step 3: Create a new population by repeating following steps until a population size is reached P .

Selection Operation: Select two parent chromosomes from a population according to their fitness by using the roulette wheel method.

Crossover Operation: A single point crossover(firing bit is excluded) method is conducted to form a new offspring.

Mutation Operation: With a mutation rate, mutate new offer spring at each locus (position in chromosome, firing bit excluded). Place new offspring in a new population.

Step 4: Genetic algorithm is done with newly generated populations. If the maximum number of generations has been reached, Genetic algorithm is stopped. Otherwise, return to to Step 2.

6. Performance evaluation

Two typical data sets are used to valuate the proposed method. The results are compared with methods of neural network and reference [6].

6.1. The recognition experiment with wine data set

Wine data [10] include 178 datum with 13 input attributes and 3 classes. 90 datum are used as training samples while 88 datum are used as testing samples. Parameters of experiment are as follows. $dth_1=dth_2=dth_3=0.2$. The minimum number of training samples for creating a rule is 6. The maximum number of rules for a class is 8. Initial value of δ_{ij} is 1, $P=30$, $Generation=3000$. Mutation probability is 0.001. Others, $\gamma=0.1$, $k=0.75$, $NFC=150$. Table 1 is the experimental result of the proposed method. Table 2 is comparison of accuracy for proposed method and reference [6] method.

Table 1. The experimental result of the proposed method

Generations	Time(sec.)	Accuracy(%)
10	4	76.78
30	11	81.28
150	42	82.60
480	128	84.25
580	150	85.24
1000	270	85.75
1500	500	88.23
1800	589	89.86
2000	658	92.49

Table 2. Comparison of accuracy for wine data

		PM(%)	RM(%)
Training samples(90)	Average accuracy	92.49	88.7
	Maximum accuracy	94.8	90
Test samples(88)	Average accuracy	89.6	81.8
	Maximum accuracy	92.3	86
The average number of rules		10	3.5
The number of evolution generations		3000	5000

* PM: Proposed method RM: Reference [6] method

6.2. Experiment of handwritten digits data[10]

This data set included 7494 training samples and 3498 test training samples with 16 input attributes and 10 classes. Table 3 is comparison of accuracy and training time for proposed method and neural network method. Parameters of the experiment are as follows. $dth_1=\dots=dth_{10}=0.25$. The minimum number of training samples for creating a rule is 32. The maximum number of rules for each class is

30. Initial value of δ_{ij} is 1, $P=60$, $\text{Generation}=3000$, $\gamma=0.1$. Mutation rate is 0.001, $k=0.78$, $\text{NFC}=300$. The neural network has a 3-layer structure and 10 hidden layer units. Other parameters are as follows: initial weight is distributed in $[-0.1, 0.1]$. Different initial values are used to train for 15 times. epoch is 2000. The learning speed is 0.8. The momentum is 0.

Table 3. Comparison for Handwritten Digits data[10]

Classifier system	CPU time (s)	Rule numbers	Accuracy of training sample	Accuracy of testing sample
PM	720.9	57	94.7%	92.9%
NNM	2350.6	10(units)	93.9%	91.2%

* PM: Proposed method NNM: Neural Network method

These results and comparison show that the accuracy of our proposed classifier are comparable to the maximum accuracy of the multilayered neural network classifier and reference [6], the training time is much shorter.

7. Conclusion and future work

This paper introduces a method of generating rules with kernel fuzzy classifier. Firstly, the initial sample space is mapped into a high dimensional feature space by selecting appropriate kernel function. Then in the feature space, our proposed dynamic clustering algorithm dynamically separates the training samples into different clusters and finds out the support vectors of each cluster. Because dynamic clustering arithmetic is able to separate training data into clusters by one-pass, and Genetic Algorithm only adjusts δ_{ij} for each fuzzy classification rule, the process is very quick. The accuracy of the constructed classifier by our proposed method are comparable to the maximum accuracy of the multilayered neural network classifier and reference [6], and the training time is much shorter.

In the future, we will discuss the self-adaptive determination of cluster radius threshold in the dynamic cluster algorithm, and select other kernel functions and membership functions to do some comparison.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (60373062) and the Natural Science Foundation of Hunan Province of China (05JJ40101).

References

- [1] F.Uebele, S.Abe, and M.-S.Lan, "A Neural Network-Based Fuzzy Classifier", IEEE Trans. Systems, Man, and Cybernetics, Vol. 25, No. 2, pp. 353–361, 1999.
- [2] K.Shirnojima, T.FuKuda, Y.Hasegawa, "RBF-Fuzzy System with GA Based Unsupervised/Supervised Learning Method", Proc. 4th IEEE Int. Conf. Fuzzy Syst., pp.253-258, 1995.
- [3] Hiroyuki INOUE, "Automatic Generation of Fuzzy Rules Using Hyper Elliptic cone Membership Functions by Genetic Algorithms", J. Intelligent and Fuzzy systems, vol.6, no.1, pp.65-81, 1998.
- [4] M.Ryoke, H.Tamura and Y.Nakamori, "Fuzzy Rule Generation by Hyperellipsoidal Clustering", Methodologies for the Conception, Design and Application of Intelligent Syst., pp.86-89, World Scientific, 1996.
- [5] S.Abe, "Dynamic Cluster Generation for a Fuzzy Classifier with Ellipsoidal Regions", IEEE Trans. Systems, Man, and Cybernetics-Part B, Vol. 28, No. 6, pp.869–876, 1998.
- [6] Hng, L., Inoue, H., "Automatic generation of fuzzy classification systems using hyper-cone membership function", Computational Intelligence in Robotics and Automation, 2003. Proceedings. 2003 IEEE International Symposium on, Volume: 2, pp.658 – 663, July, 2003.
- [7] Chun-Fu Lin, Sheng-De Wang, "Fuzzy support vector machines", Neural Networks, IEEE Transactions on, Volume: 13, Issue: 2, pp.464– 471, March, 2002.
- [8] Ben-Hur, A., Horn, D., Siegelmann, H.T., Vapnik, V., "A support vector clustering method", Pattern Recognition, 2000. Proceedings. 15th International Conference on, Volume: 2, 3-7 pp.724 - 727 vol.2, Sept 2000.
- [9] Kaieda, K., Abe, S., "A kernel fuzzy classifier with ellipsoidal regions", Neural Networks, 2003. Proceedings of the International Joint Conference on, Volume: 3, pp.2043 – 2048, July, 2003.
- [10] [UCI Repository of Machine Learning Databases Network Document]. <ftp://ftp.ics.uci.edu/pub/machine-learning-databases/>.