

Short Paper

Generating Weighted Fuzzy Rules from Training Instances Using Genetic Algorithms to Handle the Iris Data Classification Problem*

SHYI-MING CHEN AND HAO-LIN LIN⁺

Department of Computer Science and Information Engineering

⁺*Department of Electronic Engineering*

National Taiwan University of science and Technology

Taipei, 106 Taiwan

E-mail: smchen@et.ntust.edu.tw

In recent years, many researchers have focused on applying the fuzzy set theory to generate fuzzy rules from training instances to deal with the Iris data classification problem. In this paper, we propose a new method to automatically generate weighted fuzzy rules from training instances by using genetic algorithms to handle the Iris data classification problem, where the attributes appearing in the antecedent parts of the generated fuzzy rules have different weights. The proposed method can achieve a higher average classification accuracy rate and generate fewer fuzzy rules than the existing methods.

Keywords: fuzzy rules, genetic algorithms, Iris data, weighted fuzzy rules, average classification accuracy rate

1. INTRODUCTION

In recent years, many researchers have focused on applying the fuzzy set theory [36] to generate fuzzy rules from training instances to handle fuzzy classification problems. There are two basic ways to obtain a set of fuzzy rules for a fuzzy classification system. One of them is to get knowledge from experts and then transform it into fuzzy rules, but it is time consuming. The other way is to use a machine learning method to automatically generate fuzzy rules from training instances [2, 4-13, 17-19, 21, 24, 25, 31-33, 35].

In this paper, we present a new method to generate weighted fuzzy rules from training instances using genetic algorithms to deal with the Iris data classification problem. The major difference between the proposed method and our previous works presented in [5, 9, 11, 13] and [32] is that we use genetic algorithms to get the optimal weights of the attributes appearing in the antecedent portions of the generated fuzzy rules. The proposed

Received January 14, 2004; revised June 9 & August 23, 2004; accepted August 31, 2004.

Communicated by Chin-Teng Lin.

* This work was supported in part by the National Science Council, R.O.C., under grant No. NSC 91-2213-E-011-037.

method can achieve a higher average classification accuracy rate and generate fewer fuzzy rules than the existing methods.

The rest of this paper is organized as follows. In section 2, we briefly review some basic concepts of genetic algorithms from [16, 26] and [27]. In section 3, we present a new method to generate weighted fuzzy rules based on genetic algorithms to deal with the Iris data classification problem. In section 4, we give experimental results obtained using the proposed method. We also make a comparison between the proposed method with the existing methods. Conclusions are drawn in section 5.

2. BASIC CONCEPTS OF GENETIC ALGORITHMS

In [17], Holland proposed genetic algorithms (GA) based on the concept of Darwin's theory of evolution, which describes the law of competition and natural selection. After many years of research, genetic algorithms have been proven to be effective for solving problems by using the evolution process. In a genetic algorithm, a solution that consists of a lot of parameters for solving problems is called a chromosome. Each parameter in a chromosome is called a gene. A set of chromosomes in the same generation is called a population. A function calculating the fitness value of each chromosome in a population is called a fitness function. A chromosome that has a higher fitness value is more adaptable in the environment in which it is living. Every chromosome in a population has a chance to evolve. A genetic algorithm continuously executes a cycle that involves three operations (i.e., crossover, mutation, and reproduction) to evolve a better population in the next generation. The process of evolution involves "the survival of the fittest" meaning that a weaker chromosome will die and a stronger chromosome will evolve. A genetic algorithm does not stop the process of evolution until the goal is achieved. Thus, we can obtain an optimal solution of a problem. In the following, we will briefly review three major operations of a genetic algorithm from [16, 26] and [27].

- (A) **Crossover:** Firstly, the system randomly selects two chromosomes from a population. A parameter CR (Crossover Rate) decides the crossover probability of two selected chromosomes. The system randomly generates a value, compares it with the value of CR, and decides whether the two selected chromosomes perform the crossover operations. If the randomly generated value is smaller than the value of CR, then the system exchanges some genes of the two selected chromosomes. In the following, we will review some general crossover methods from [16] and [26]:
- (1) **Single-Point Crossover:** The system randomly generates a crossover point in two selected chromosomes. Then, the system exchanges the genes of the selected two chromosomes after the crossover point. We illustrate this method as shown in Fig. 1.
 - (2) **Two-Point Crossover:** The system randomly generates two crossover points in two selected chromosomes. Then, the system exchanges the genes between the two randomly generated crossover points of the two selected chromosomes. We illustrate this method as shown in Fig. 2.
 - (3) **Uniform Crossover:** This crossover operation generates a binary string that has the same length as the chromosome. The content of the binary string is randomly generated by the system. Each generated bit of the binary string corresponds to the same

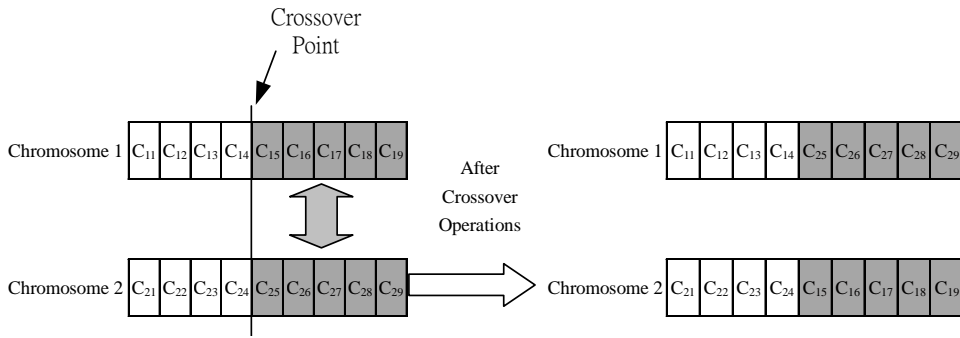


Fig. 1. The single-point crossover operation.

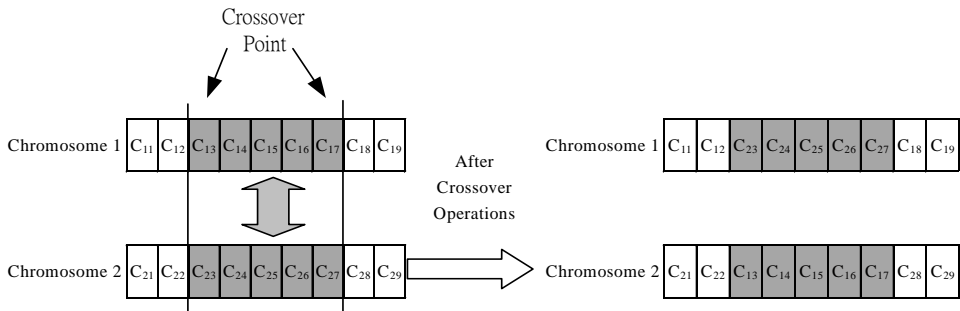


Fig. 2. The two-point crossover operation.

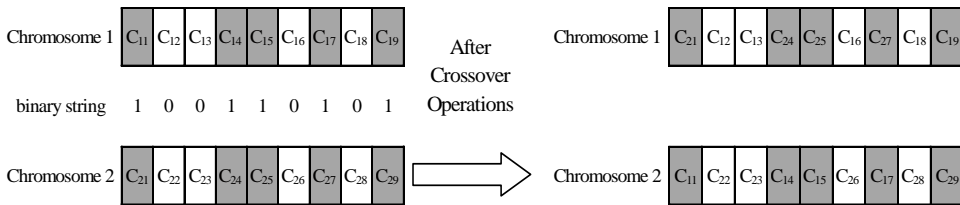


Fig. 3. The uniform crossover operation.

position in the chromosome. When the bit of the binary string is 1, we reserve the value of the same position in the genes of the two chromosomes. Otherwise, the system exchanges the genes of the two chromosomes. We illustrate this method in Fig. 3.

- (B) **Mutation:** The system randomly selects a chromosome from a population by using a parameter MR (Mutation Rate) to decide the mutation probability of the selected chromosome, where $MR \in [0, 1]$. The system randomly generates a value and compares it with the value of MR to decide whether the selected chromosome performs the mutation operation. If the value is smaller than or equal to the value of MR, then

the system mutates a gene of the selected chromosome. In the following, we will review some mutation methods from [16]:

- (1) **Binary Chromosome:** The system randomly selects a chromosome of a population and its mutation point. Then, it performs the “complement” operation on the selected gene of the selected chromosome. The mutation operation performed on a binary chromosome is illustrated in Fig. 4.
- (2) **Real-Number Chromosome:** The system randomly selects a chromosome of a population and its mutation point. It then randomly generates a mutation value, ε , where $\varepsilon \in [0, 1]$, to replace the selected gene of the selected chromosome. The mutation operation performed on a real-number chromosome is illustrated in Fig. 5.

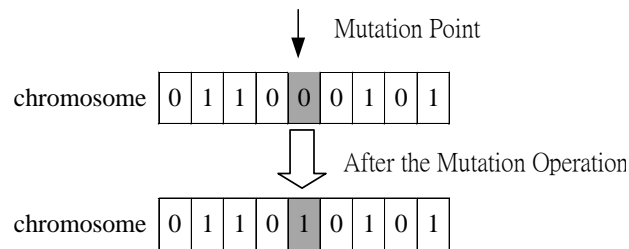


Fig. 4. The mutation operation performed on a binary chromosome.

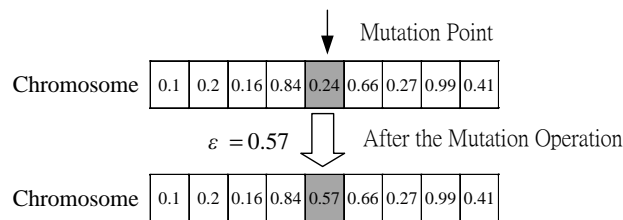


Fig. 5. The mutation operation performed on a real-number chromosome.

- (C) **Reproduction:** The genetic algorithm performs the reproduction operation to achieve the goal of “the survival of the fittest” according to the fitness value of each chromosome. In this paper, we use the “roulette wheel selection” method [14]. Let g_i denote the fitness value of the i th chromosome in a population, and let g_t denote the sum of the fitness values of all the chromosomes, where $g_t = \sum_{i=1}^n g_i$. The selection probability P_{g_i} of the i th chromosome is denoted by $P_{g_i} = g_i/g_t$. The system randomly generates a real value in $[0, g_t]$. If the fitness value of a chromosome is larger than the generated value, then the system has a better chance of selecting this chromosome.

In [12], Chen *et al.* presented a method to generate weighted fuzzy rules from relational database systems to estimate null values using genetic algorithms. In [20], Ho *et al.*

designed a fuzzy classifier using an intelligent genetic algorithm. In [22], Ishibuchi *et al.* presented a method for minimizing the fuzzy rule base and maximizing its performance by using a multiobjective genetic algorithm. In [28], Nawa *et al.* presented a study on the discovery of relevant fuzzy rules using a pseudobacterial genetic algorithm. In [29], Onisawa *et al.* presented a method for fuzzy rule acquisition using interactive algorithms.

3. A NEW METHOD TO GENERATE WEIGHTED FUZZY RULES BASED ON GENETIC ALGORITHMS

In 1965, Zadeh proposed the theory of fuzzy sets [36]. Let A and B be fuzzy sets of the universe of discourse U , and let μ_A and μ_B be the membership functions of the fuzzy sets A and B , respectively, where $\mu_A : U \rightarrow [0, 1]$ and $\mu_B : U \rightarrow [0, 1]$. The fuzzy subsethood $S(A, B)$ [23] measures the degree to which A is a subset of B :

$$S(A, B) = \frac{M(A \cap B)}{M(A)} = \frac{\sum_{u \in U} \min(\mu_A(u), \mu_B(u))}{\sum_{u \in U} \mu_A(u)}, \quad (1)$$

where $S(A, B) \in [0, 1]$.

In the following, we will present a new method to generate weighted fuzzy rules from training instances using genetic algorithms to deal with the Iris data [15] classification problem. Iris data has three species of flowers, i.e., Setosa, Versicolor, and Virginica, where each species has 50 instances. There are four attributes in Iris data, i.e., Sepal Length (SL), Sepal Width (SW), Petal Length (PL) and Petal Width (PW). Initially, we must decide on the environment and generate fuzzy rules. Then, we can apply the genetic algorithm to find the weights of the attributes in the generated fuzzy rules. There are 30 chromosomes in each population. A chromosome consisting of a set of genes is a real-number string, where the content of each gene is between 0 and 1, representing the weight of an attribute. In this paper, we use the single-point crossover method and the real number mutation method for evolution, where the crossover rate and the mutation rate are 0.9 and 0.1, respectively. The fitness function uses the classification accuracy rate to calculate the fitness value of each chromosome to generate weighted fuzzy rules from a training data set.

Firstly, we must translate the numerical training data into fuzzified data. In this paper, we adopt Castro's three-linguistic-terms membership functions [4] for each attribute of the Iris data as shown in Figs. 6-9.

In the following, we will use the Iris data to illustrate the proposed method for generating weighted fuzzy rules from training instances based on the genetic algorithm. The algorithm for generating weighted fuzzy rules from training instances is now presented as follows:

Step 1: Fuzzify the crisp attribute values of the training instances to obtain fuzzified values by using the membership functions shown in Figs. 6-9.

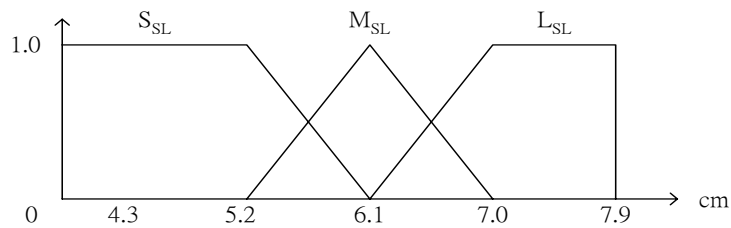


Fig. 6. Membership functions of the linguistic terms of the attribute "SL".

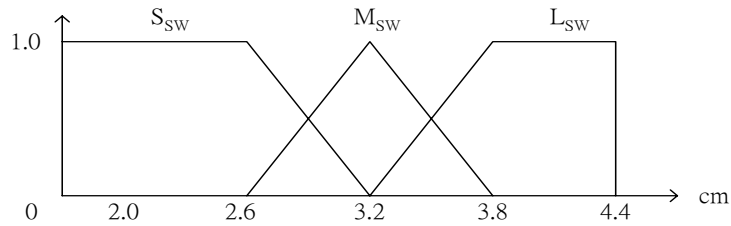


Fig. 7. Membership functions of the linguistic terms of the attribute "SW".

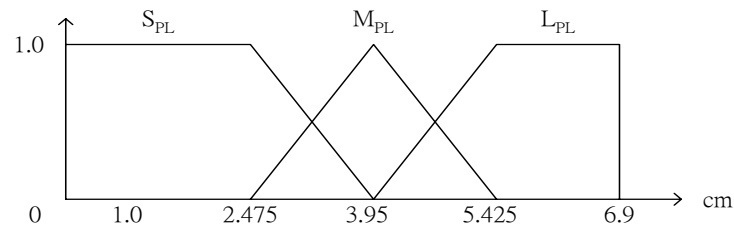


Fig. 8. Membership functions of the linguistic terms of the attribute "PL".

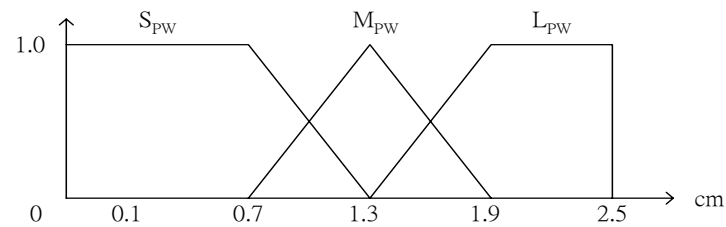


Fig. 9. Membership functions of the linguistic terms of the attribute "PW".

Step 2: Apply γ -level-cut operations to the fuzzified values of each attribute, where $\gamma \in [0, 1]$.

Step 3: Based on the results of step 2, analyze the relationships between the species of flowers and the attributes, respectively, to get the frequency distribution tables for the attributes.

Step 4: Translate the frequency distribution tables of the attributes derived in step 3 into probability distribution tables, respectively.

- Step 5:** For each species of flower (i.e., *Setosa*, *Versicolor*, and *Virginica*), according to the probability distribution table for each attribute and formula (1), calculate the fuzzy subsethood values of each output and each term of each attribute.
- Step 6:** Based on [7], generate fuzzy rules based on the fuzzy subsethood values derived in step 5. For example, after we get the fuzzy subsethood values of each species of flower (i.e., *Setosa*, *Versicolor*, and *Virginica*) and each linguistic term of each attribute, we can select the terms of attributes for each species of flower to form a fuzzy rule. We select the linguistic terms of the attributes with the highest fuzzy subsethood values with respect to the species of flower to form the antecedent part of the rule for each species of flower, and we use the species of flower to form the consequence part of the fuzzy rule. We use a level threshold value, α , to decide whether we want to select the linguistic terms of the attributes or not, where the level threshold value α is given by the user and is between 0 and 1. We also consider the fuzzy subsethood values of the complement of the linguistic terms of the attributes for each species of flower. We choose the linguistic terms of the attributes whose fuzzy subsethood values are not less than the level threshold value, α , where $\alpha \in [0, 1]$. If there are two linguistic terms with fuzzy subsethood values that are not less than the level threshold value, α , then the one with the largest fuzzy subsethood value will be chosen. When a linguistic term whose fuzzy subsethood value with respect to the species of flower is equal to the fuzzy subsethood value of the complement of the linguistic term with respect to the species of flower, we choose the original linguistic term to generate fuzzy rules.

Then, we must decide the weight of each attribute appearing in the antecedent parts of the generated fuzzy rules by applying the genetic algorithm. The genetic algorithm can tune the weights of the attributes appearing in the antecedent parts of the generated fuzzy rules to get the highest classification accuracy rate to deal with the Iris data classification problem. Firstly, we must determine the environment of the genetic algorithm. There are 30 chromosomes in each population. From the generated fuzzy rules, we can see that the number of attributes appearing in the antecedent parts of the generated fuzzy rules is 9. Thus, each chromosome consists of 9 genes representing the weights of the attributes appearing in the antecedent parts of the generated fuzzy rules. That is, a chromosome is a string consisting of 9 real numbers between 0 and 1, indicating the weights of the attributes. In this paper, we use the single-point crossover operation and the real number mutation operation, where the crossover rate and the mutation rate are 0.9 and 0.1, respectively. The fitness function uses the classification accuracy rate of the generated weighted fuzzy rules corresponding to a chromosome to calculate the fitness value of the chromosome.

- Step 7:** Randomly generate 30 chromosomes of the initial population consisting of the weights of the attributes appearing in the antecedent parts of the generated fuzzy rules.
- Step 8:** Assign the content (i.e., the weight) of each gene of each chromosome of the initial population to the corresponding attributes appearing in the antecedent parts of the generated fuzzy rules. Evaluate the fitness value of each chromo-

some in the initial population, where the fitness value of the chromosome is defined as the classification accuracy rate of the corresponding generated weighted fuzzy rules of the chromosome.

Step 9: If the number of evolutions reaches m , where m is a positive integer, then go to step 14. Otherwise, go to step 10.

Step 10: Perform the reproduction operation. This process selects the chromosome whose expected fitness value [14] is not less than 1. If the number of the selected chromosomes is less than 30, then the system uses the roulette wheel method to get the rest of the chromosomes of the population. We can calculate the expected fitness value of each chromosome. The formulas for calculating the expected fitness value $E(chromosome_i)$ of the i th chromosome $Chromosome_i$ are as follows:

$$AVG(\text{Fitness Value}) = \frac{\sum_{i=1}^{30} F_{chromosome_i}}{30}, \quad (2)$$

$$E(chromosome_i) = \frac{F_{chromosome_i}}{AVG(\text{Fitness Value})}, \quad (3)$$

where $F_{chromosome_i}$ denotes the fitness value of $Chromosome_i$, $E(chromosome_i)$ denotes the expected fitness value of $Chromosome_i$, and $1 \leq i \leq 30$. After applying formulas (2) and (3), we can get the expected fitness values of the chromosomes in the initial population. After calculating the expected fitness value of each chromosome in the initial population, the system chooses the chromosomes whose expected fitness values are not less than 1.

Step 11: Perform the single-point crossover operation on the population. The system randomly generates a number and compares it with the crossover rate to decide whether two randomly selected chosen-chromosomes will undergo to the crossover operation. In this step, the crossover operation does not stop until all of the chosen-chromosomes have been selected. If the number randomly generated by the system is larger than the crossover rate, then the system does not need to perform the crossover operation.

Step 12: Perform the real-number mutation operation. The system randomly generates a real number between zero and one and compares it with the mutation rate to decide whether the randomly selected derived-chromosome needs to undergo the mutation operation. When all of the derived-chromosomes are selected, the mutation procedure stops. If the real number randomly generated by the system is larger than the mutation rate, then the system does not need to perform the mutation operation.

Step 13: Evaluate each chromosome of the resultant population using the fitness function by applying the corresponding generated weighted fuzzy rules to classify the training instances to get the classification accuracy rate. Then, the system selects 24 derived-chromosomes from the resultant population that get higher fitness values and selects 6 chromosomes from the population of the former gen-

eration that have higher fitness values to form the population of the next generation. Then, the system goes to step 9 to perform the next cycle of evolution.

Step 14: Choose the best derived-chromosome that has the largest fitness value to obtain the weights of the attributes appearing in the antecedent parts of the generated weighted fuzzy rules to deal with the Iris data classification problem.

4. EXPERIMENTAL RESULTS

We implemented the proposed method on a Pentium 4 PC using Visual Basic Version 6.0. We considered the following three kinds of situations with the Iris data: (1) 150 instances for training and the same 150 instances for testing; (2) 75 instances for training and the other 75 instances for testing; and (3) 120 instances for training and the other 30 instances for testing. The experimental results for the above three cases are as follows:

Case 1: 150 instances for training and the same 150 instances for testing. In this case, the whole Iris data set acts as a training data set and a testing data set, respectively. Using the proposed method, we obtained 3 weighted fuzzy rules and a classification accuracy rate after each run by applying the generated weighted fuzzy rules to the testing data set, where the weighted fuzzy rules could be generated by different numbers of evolutions.

Case 2: 75 instances for training and the other 75 instances for testing. In this case, the system randomly chooses 75 instances of the Iris data as the training data set and lets the other 75 instances of the Iris data be the testing data set. Using the proposed method, we obtained 3 weighted fuzzy rules and obtain a classification accuracy rate after each run by applying the generated weighted fuzzy rules to the testing data set, where the weighted fuzzy rules could be generated by different numbers of evolutions.

Case 3: 120 instances for training and the other 30 instances for testing. In this case, the system randomly chooses 120 instances of the Iris data be the training data set and lets the other 30 instances of the Iris data be the testing data set. Using the proposed method, we obtained 3 weighted fuzzy rules and a classification accuracy rate after each run by applying the generated weighted fuzzy rules to the testing data set, where the weighted fuzzy rules could be generated by different numbers of evolutions.

In Table 1, we compare the experimental results obtained using the proposed method with those obtained using the existing methods. The proposed method can achieve a higher average classification accuracy rate and generate fewer fuzzy rules than the existing methods.

5. CONCLUSIONS

In this paper, we have presented a new method to automatically generate weighted fuzzy rules from training instances using genetic algorithms to deal with the Iris data

Table 1. A comparison of the experimental results obtained using the proposed method with those obtained using the existing methods.

Methods		Average Classification Accuracy Rate	Average Number of Generated Fuzzy Rules
Training Data Set: 120 Instances; Testing Data Set: 30 Instances	Castro-Castro-Schez-Zurita's Method [4] (after executing 10 Runs)	96.60 %	11
	Chen-and-Chen's Method [13] (after executing 200 Runs)	96.70 %	8.849
	Chen-and-Yu's Method [11] (after executing 200 Runs)	96.65 %	N/A*
	The Proposed Method (after executing 200 Runs)	96.87 %	3
Training Data Set: 75 Instances; Testing Data Set: 75 Instances	Hong-and-Lee's Method [18] (after executing 200 Runs)	95.57 %	6.21
	Hong-and-Chen's Method [19] (after executing 200 Runs)	95.57 %	6.21
	Chang-and-Chen's Method [5] (after executing 200 Runs)	96.07 %	3
	Wu-and-Chen's Method [33] (after executing 200 Runs)	96.21%	3
	Chen-and-Yu's Method [11] (after executing 200 Runs)	96.24 %	N/A*
	Tsai-and-Chen's Method [28] (after executing 200 Runs)	95.833 %	7.205
	Incremental Learning Fuzzy Neural Network Method [34]	96.268 %	N/A*
	Fuzzy Classifier with the Hyperbox Regions Method [1]	Minimum: 92 % Maximum: 97.3 %	5 to 17
	Fuzzy ARTMAP Neural Network Method [3]	93.467 %	N/A*
The Proposed Method (after executing 200 Runs)	96.33 %	3	

* Note: N/A means the average number of generated fuzzy rules is not mentioned in the paper.

classification problem. The proposed method generates fuzzy rules and tunes the weights of the attributes appearing in the antecedent parts of the generated weighted fuzzy rules based on the genetic algorithm. The proposed method can achieve a higher average classification accuracy rate and generate fewer fuzzy rules than the existing methods. In the future, we will extend the proposed method to deal with more complex classification problems. Furthermore, we will develop a method for choosing the optimal values of the parameters γ and α to get higher average classification accuracy rates to deal with classification problems.

REFERENCES

1. S. Abe and M. S. Lan, "A method for fuzzy rules extraction directly from numerical data and its application to pattern classification to pattern classification," *IEEE Transactions on Fuzzy Systems*, Vol. 3, 1995, pp. 18-28.
2. D. G. Burkhardt and P. P. Bonissone, "Automated fuzzy knowledge base generation and tuning," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, 1992, pp. 179-188.
3. G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, "Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Transactions on Neural Networks*, Vol. 3, 1992, pp. 698-713.
4. J. L. Castro, J. J. Castro-Schez, and J. M. Zurita, "Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems," *Fuzzy Sets and Systems*, Vol. 101, 1999, pp. 331-342.
5. C. H. Chang and S. M. Chen, "Constructing membership functions and generating weighted fuzzy rules from training data," in *Proceedings of 9th National Conference on Fuzzy Theory and Its Applications*, 2001, pp. 708-713.
6. C. H. Chang and S. M. Chen, "A new method to generate fuzzy rules from numerical data based on the exclusion of attribute terms," in *Proceedings of the International Computer Symposium: Workshop on Artificial Intelligence*, 2000, pp. 57-64.
7. S. M. Chen, S. H. Lee, and C. H. Lee, "A new method for generating fuzzy rules from numerical data for handling classification problems," *International Journal of Applied Artificial Intelligence*, Vol. 15, 2000, pp. 645-664.
8. S. M. Chen, C. H. Kao, and C. H. Yu, "Generating fuzzy rules from training data containing noise for handling classification problems," *International Journal of Cybernetics and Systems*, Vol. 33, 2002, pp. 723-748.
9. S. M. Chen and C. H. Chang, "A new method to construct membership functions and generate weighted fuzzy rules from training instances," *International Journal of Cybernetics and Systems*, Vol. 36, 2005, pp. 397-414.
10. S. M. Chen and S. W. Lee, "A new method to generate fuzzy rules from relational database systems for estimating null values," *International Journal of Cybernetics and Systems*, Vol. 34, 2003, pp. 33-57.
11. S. M. Chen and C. H. Yu, "A new method for handling fuzzy classification problems using clustering techniques," *International Journal of Applied Science and Engineering*, Vol. 2, 2004, pp. 90-104.
12. S. M. Chen and C. M. Huang, "Generating weighted fuzzy rules from relational database systems for estimating null values using genetic algorithms," *IEEE Transactions on Fuzzy Systems*, Vol. 11, 2003, pp. 495-506.
13. Y. C. Chen and S. M. Chen, "A new method to generate fuzzy rules for fuzzy classification systems," in *Proceedings of the National Conference on Fuzzy Theory and Its Applications*, 2000.
14. L. Davis, *Handbook of Genetic Algorithm*, Van Nostrand Reinhold, New York, 1991.
15. R. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, Vol. 7, 1936, pp. 179-188.

16. M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design*, John Wiley & Sons, New York, 1997.
17. J. H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA, 1975.
18. T. P. Hong and C. Y. Lee, "Induction of fuzzy rules and membership functions from training examples," *Fuzzy Sets and Systems*, Vol. 84, 1996, pp. 33-47.
19. T. P. Hong and J. B. Chen, "Finding relevant attributes and membership functions," *Fuzzy Sets and Systems*, Vol. 103, 1999, pp. 389-404.
20. S. Y. Ho, T. K. Chen, and S. J. Ho, "Designing an efficient fuzzy classifier using an intelligent genetic algorithm," in *Proceedings of the Computer Software and Applications Conference*, 2000, pp. 293-298.
21. H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selecting fuzzy if-then rules for classification problems using genetic algorithms," *IEEE Transactions on Fuzzy Systems*, Vol. 3, 1995, pp. 260-270.
22. H. Ishibuchi and T. Murata, "Minimizing the fuzzy rule base and maximizing its performance by a multiobjective genetic algorithm," in *Proceedings of 6th International Conference on Fuzzy Systems*, Vol. 1, 1997, pp. 259-264.
23. B. Kosko, "Fuzzy entropy and conditioning," *Information Sciences*, Vol. 40, 1986, pp. 165-174.
24. H. L. Lin and S. M. Chen, "Generating weighted fuzzy rules from training data for handling fuzzy classification problems," in *Proceedings of the International Computer Symposium: Workshop on Artificial Intelligence*, 2000, pp. 11-18.
25. H. L. Lin and S. M. Chen, "A new method for generating weighted fuzzy rules from training instances using genetic algorithms," in *Proceedings of 6th Conference on Artificial Intelligence and Applications*, 2001, pp. 628-633.
26. K. F. Man, K. S. Tang, and S. Kwong, "Genetic algorithms: concepts and applications," *IEEE Transactions on Industrial Electronics*, Vol. 43, 1996, pp. 519-534.
27. Z. Michalewicz, *Genetic Algorithms + Data Structure = Evolution Programs*, Springer, Berlin, 1992.
28. N. E. Nawa, T. Furuhashi, T. Hashiyama, and Y. Uchikawa, "A study on the discovery of relevant fuzzy rules using pseudobacterial genetic algorithm," *IEEE Transactions on Industrial Electronics*, Vol. 46, 1999, pp. 1080-1089.
29. T. Onisawa and Y. Fujihare, "Fuzzy rules acquisition using interactive genetic algorithms," in *Proceedings of 41st Annual Conference on the Society of Instrument and Control Engineers (SICE)*, Vol. 5, 2002, pp. 2887-2892.
30. M. Setnes and H. Roubos, "GA-fuzzy modeling and classification: complexity and performance," *IEEE Transactions on Fuzzy Systems*, Vol. 8, 2000, pp. 509-522.
31. T. Sudkamp and R. J. Hammell II, "Interpolating, complete, and learning fuzzy rules," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 24, 1994, pp. 332-342.
32. F. M. Tsai and S. M. Chen, "A new method for constructing membership functions and generating fuzzy rules for fuzzy classification systems," in *Proceedings of 10th National Conference on Fuzzy Theory and Its Applications*, 2002.
33. T. P. Wu and S. M. Chen, "A new method for constructing membership functions and fuzzy rules from training examples," *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, Vol. 29, 1999, pp. 25-40.

34. G. G. Yen and P. Meesad, "An effective neuro-fuzzy paradigm for machinery condition health monitoring," *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, Vol. 31, 2001, pp. 523-536.
35. Y. Yuan and M. J. Shaw, "Induction of fuzzy decision trees," *Fuzzy Sets and Systems*, Vol. 69, 1995, pp. 125-139.
36. L. A. Zadeh, "Fuzzy sets," *Information and Control*, Vol. 8, 1965, pp. 338-353.

Shyi-Ming Chen (陳錫明) received the B.S. degree in Electronic Engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan, in June 1982, and the M.S. and Ph.D. degrees in Electrical Engineering from the National Taiwan University, Taipei, Taiwan, in June 1986 and June 1991, respectively. From August 1987 to July 1989 and from August 1990 to July 1991, he was with the Department of Electronic Engineering, Fu Jen University, Taipei, Taiwan. From August 1991 to July 1996, he was an Associate Professor in the Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan. From August 1987 to July 1989 and from August 1990 to July 1991, he was with the Department of Electronic Engineering, Fu-Jen University, Taipei, Taiwan. From August 1991 to July 1996, he was an Associate Professor in the Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan. From August 1996 to July 1998, he was a Professor in the Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan. From August 1998 to July 2001, he was a Professor in the Department of Electronic Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan. Since August 2001, he has been a Professor in the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan. He was a Visiting Scholar in the Department of Electrical Engineering and Computer Science, University of California, Berkeley, in 1999. He was a Visiting Scholar in the Institute of Information Science, Academia Sinica, Republic of China, in 2003. He has published more than 220 papers in referred journals, conference proceedings and book chapters. His research interests include fuzzy systems, information retrieval, knowledge-based systems, artificial intelligence, neural networks, data mining, and genetic algorithms.

Dr. Chen has received several honors and awards, including the 1994 Outstanding Paper Award of the Journal of Information and Education, the 1995 Outstanding Paper Award of the Computer Society of the Republic of China, the 1995 and 1996 Acer Dragon Thesis Awards for Outstanding M.S. Thesis Supervision, the 1995 Xerox Foundation Award for Outstanding M.S. Thesis Supervision, the 1996 Chinese Institute of Electrical Engineering Award for Outstanding M.S. Thesis Supervision, the 1997 National Science Council Award, Republic of China, for Outstanding Undergraduate Student's Project Supervision, the 1997 Outstanding Youth Electrical Engineer Award of the Chinese Institute of Electrical Engineering, Republic of China, the Best Paper Award of the 1999 National Computer Symposium, Republic of China, the 1999 Outstanding Paper Award of the Computer Society of the Republic of China, the 2001 Institute of Information and Computing Machinery Thesis Award for Outstanding M.S. Thesis Supervision, the 2001 Outstanding Talented Person Award, Republic of China, for the contributions in Information Technology, the 2002 Institute of information and Computing

Machinery Thesis Award for Outstanding M.S. Thesis Supervision, the Outstanding Electrical Engineering Professor Award granted by the Chinese Institute of Electrical Engineering (CIEE), Republic of China, the 2002 Chinese Fuzzy Systems Association Best Thesis Award for Outstanding M.S. Thesis Supervision, the 2003 Outstanding Paper Award of the Technological and Vocational Education Society, Republic of China, the 2003 Acer Dragon Thesis Award for Outstanding Ph.D. Dissertation Supervision, the 2005 "Operations Research Society of Taiwan" Award for Outstanding M.S. Thesis Supervision, the 2005 Acer Dragon Thesis Award for Outstanding Ph.D. Dissertation Supervision, and the 2005 Chinese Fuzzy Systems Association Award for Outstanding Ph.D. Dissertation Supervision.

Dr. Chen is currently the President of the Taiwanese Association for Artificial Intelligence (TAAI). He is a Senior Member of the IEEE, a member of the ACM, the International Fuzzy Systems Association (IFSA), and the Phi Tau Phi Scholastic Honor Society. He was an administrative committee member of the Chinese Fuzzy Systems Association (CFSA) from 1998 to 2004. He is an Editor of the International Journal of New Mathematics and Natural Computation (published by World Scientific). He is also an Associate Editor of the IEEE Transactions on Systems, Man, and Cybernetics - Part C and an Associate Editor of the International Journal of Fuzzy Systems. He is an Editorial Board Member of the International Journal of Information and Communication Technology (Published by InderScience Publishers) and an Editorial Board Member of the WSEAS Transactions on Systems. He was an Editor of the Journal of the Chinese Grey System Association from 1998 to 2003. He is listed in *International Who's Who of Professionals*, *Marquis Who's Who in the World*, and *Marquis Who's Who in Science and Engineering*.

Hao-Lin Lin (林皓琳) received the B.S. degree from the Department of Information Engineering, I-Shou University, Kaohsiung, Taiwan, R.O.C., in June 1999. He received the M.S. degree in Computer Science and Information Engineering at National Taiwan University of Science and Technology, Taipei, Taiwan, in June 2001. His current research interests include fuzzy systems, data mining, and artificial intelligence.