JOURNAL OF KOREA MULTIMEDIA SOCIETY, VOL. 7, NO. 6, JUNE 2004(pp. 841-850)

Fuzzy Learning Method Using Genetic Algorithms

Sangho Choi†, Kyungdal Cho††, Sajoon Park†††, Malrey Lee††††, Kitae Kim†††††

ABSTRACT

This paper proposes a GA and GDM-based method for removing unnecessary rules and generating relevant rules from the fuzzy rules corresponding to several fuzzy partitions. The aim of proposed method is to find a minimum set of fuzzy rules that can correctly classify all the training patterns. When the fine fuzzy partition is used with conventional methods, the number of fuzzy rules has been enormous and the performance of fuzzy inference system became low. This paper presents the application of GA as a means of finding optimal solutions over fuzzy partitions. In each rule, the antecedent part is made up the membership functions of a fuzzy set, and the consequent part is made up of a real number. The membership functions and the number of fuzzy inference rules are tuned by means of the GA, while the real numbers in the consequent parts of the rules are tuned by means of the gradient descent method. It is shown that the proposed method has improved than the performance of conventional method in formulating and solving a combinatorial optimization problem that has two objectives: to maximize the number of correctly classified patterns and to minimize the number of fuzzy rules.

Keywords: Genetic Algorithm, Gradient Descent Method, Fuzzy Inference Model, Classification Problem

1. INTRODUCTION

Many applications of fuzzy reasoning to construct advanced controllers have been reported. Most of these controllers are constituted of a fuzzy model described in the IF-THEN type rules that are derived from the qualitative knowledge and the experimental know-how of experts or experienced operators. The studies of information processing systems based on fuzzy rules have been mainly applied to control problems[1,2]. The fuzzy rules that are used in most fuzzy control systems are generally derived from human expert's experience including linguistic information. To build a high non-linear system, several experiments, interviews and trials and errors are taken to acquire proper fuzzy rules. That fuzzy rules are provided by using numerical information of I/O data, doing the study to use a learning and having been the study used the neural network and clustering[3,4]. For classification problems, the automated method of generation of fuzzy rules has been proposed by Ishibuchi et al[5,6]. The generation of fuzzy rules from numerical data for pattern in classification problems consists of two phases. Fuzzy partition of a pattern space and the determination of fuzzy rules for each fuzzy subspace. The pattern space is divided by a fuzzy grid into smaller fuzzy subspaces and the fuzzy rules are generated from each the fuzzy subspace. The performance of fuzzy

※ Corresponding Author: Kyungdal Cho, Address: (155-756) Artificial Intelligence Laboratory, Chung-Ang Univ. 221, HeukSeok-Dong, Dongjak-Gu, Seoul, Korea. TEL : +82-2-820-5304, FAX : +82-2-817-8729, E-mail : kdcho88@daum.net
Receipt date : Feb. 12, 2004, Approval date : June 18, 2004
† SK Communications, the IT and Internet portal Company
(E-mail: chrischoi@nate.com)
†† Ph.D. degree, Dept. of Computer Science and Engineering, Chung-Ang Univ.
(E-mail: kdcho88@daum.net)
††† Ph.D. degree, Dept. of Computer Science and Engineering, Chung-Ang Univ.
(E-mail: phdjoon@ailab.cse.cau.ac.kr)
†††† Professor, Div. of Electronics and Information Engineering, Chonbuk National Univ.
(E-mail: mrlee@chonbuk.ac.kr)
††††† Professor, Dept. of Computer Science and Engineering, Chung-Ang Univ.
(E-mail: ktkim@ailab.cse.cau.ac.kr)

classification systems depends on the choice of the fuzzy partitions. If the partition is too coarse, the performance may be low (many patterns may be unclassified). On the other hand if the fuzzy partition is too fine, many of the necessary fuzzy rules may not be generated because of the lack of training patterns in the corresponding fuzzy subspace. So the choice of the fuzzy partitions is very important. For example, consider the two-class classification problem shown in Fig. 2, that closed circles and open circles denote the patterns in class 1 and class 2, respectively. Because of the difficulty of finding proper simple fuzzy grid, it is complicated that appropriate fuzzy partitions based on a simple fuzzy grid are chosen. Then distributed fuzzy rules were proposed in[4,5,8-10], that all the fuzzy rules corresponding to several fuzzy partitions were simultaneously employed in fuzzy classification system. The main drawback of these approaches is that the number of fuzzy rules is enormous. If unnecessary distributed fuzzy rules are removed and only relevant fuzzy rules are selected, performance of the well-selected rule set may be high in spite of less number of fuzzy rules. This paper proposes the GA[7,14] and GDM-based [13] method for removing the unnecessary rules and selecting the relevant rules from fuzzy rules corresponding to several fuzzy partitions. The aim of proposed method is to find a minimum set of fuzzy rules that should classify correctly all the training patterns. This is achieved by formulating and solving a combinatorial optimization problem that has two objectives: to maximize the number of correctly classified patterns and to minimize the number of fuzzy rules. A fine fuzzy partition will be used that good fuzzy value classification system consists of. And to solve the problem of fine fuzzy partitions, a fuzzy classification system will employ the concept of rules that can use a large number of fuzzy partitions simultaneously. This method solves the problem of degrading the system efficiency on account of using too many fuzzy rules in inference. In the proposed method, the fuzzy rules corresponding to various fuzzy partitions are simultaneously utilized in the fuzzy inference process. Fuzzy rules are generated from an area with the highest inference errors among those areas that are divided by two membership functions of neighboring antecedent part. The fuzzy of inferences are structured by a set of simple fuzzy rules. In each rule, the antecedent part is made up the membership functions of a fuzzy set, and the consequent part is made up of a real number. The membership functions and the number of fuzzy inference rules are tuned by means of the GA, while the real numbers in the consequent parts of the rules are tuned by gradient descent method. GA solve problems by using principles inspired by natural selection: they maintain a population of knowledge structures that represent candidate solutions, and then let that population evolve over time through competition and controlled variation [8-10].

The rest of this paper is organized as follows. In chapter 2, fuzzy reasoning, chapter 3 described the selecting of fuzzy rules. In chapter 4, simulation is described. Finally, conclusion is presented in chapter 5.

2. FUZZY REASONING

This section explains the method used to control the real numbers in the consequent parts of the rules by using simple inference and the gradient descent method. When the inputs are expressed as $x_1, x_2, \ldots, x_m$, and the output is expressed as y, an inference rule used in simplified fuzzy reasoning can be expressed as follows[12]. Equation (1) is shown below.

$$\text{Rule } i\text{: IF } x_1 \text{ is } A_{i1} \ldots \text{ and } x_m \text{ is } A_{im} \text{ THEN } y \text{ is } w_i \quad (i=1,\ldots,n) \qquad (1)$$

where i is rule number, $A_{i1}, \ldots, A_{im}$ are the membership functions of the antecedent part, and w_i is a real number in the consequent part. The membership function $A_{ij}(x_j)$ of the antecedent part is represented by an isosceles triangle as shown in Fig. 1.

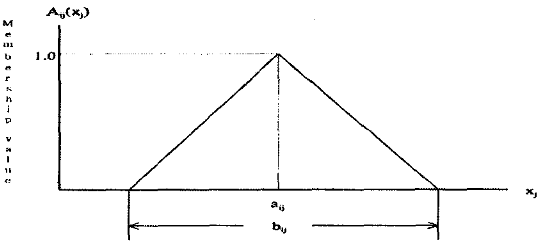


Fig. 1. Membership function of antecedent part

The parameters determining the triangle are the center value a_{ij} and the width b_{ij}. The output y of the fuzzy reasoning can be derived from the equations (2,3 and 4) shown below.

$$A_{ij}(x_j) = 1 - \frac{2 \cdot |x_j - a_{ij}|}{b_{ij}} \quad (j=1,\ldots,m) \in [0,1] \tag{2}$$

$$\mu_i = \min\{A_{i1}(x_1), A_{i2}(x_2), \ldots, A_{im}(x_m)\} \tag{3}$$

$$y = \frac{\sum_{i=1}^{n} \mu_i \cdot w_i}{\sum_{i=1}^{n} \mu_i} \tag{4}$$

where μ_i is the membership value of the antecedent part. This is used as an entity (a member of the population) in the genetic algorithm to search the global solution. A real number in the consequent part, w_i, is optimized by the gradient descent method to provide a local fine tuning mechanism for the GA.

The gradient descent method is designed to seek for the vector Z that minimizes an objective function E(Z), where Z is a p-dimensional vector $Z = (Z_1, Z_2, \ldots, Z_P)$ of the tuning parameters[11]. In this method, the vector that decreases the value of an objective function E(Z) is expressed as $(-\partial E/\partial Z_1, -\partial E/\partial Z_2, \ldots, -\partial E/\partial Z_p)$, and the learning rule is expressed by the following equation (5).

$$Z_i(t+1) = Z_i(t) - K \cdot \frac{\partial E(Z)}{\partial Z_i} \quad (i=1,\ldots,p) \tag{5}$$

where t is a number of iterations of learning and K is a constant. By altering Z according to this learning rule, the value of the objective function E(Z) converges to a local minimum[13].

In the present method, the inference rules are tuned so as to minimize the objective function E(6) which is defined as follows.

$$E = \frac{1}{2}(y - y^p)^2 \tag{6}$$

where y^p is the desirable output data (as acquired from specialists). The objective function E represents the inference error between the desirable output y^p and the output of fuzzy reasoning, y.

The objective function E consists of the tuning parameter w_i. From Eq. 5, the learning rules of simplified fuzzy reasoning are expressed by Eq.7.

$$W_i(t+1) = w_i - k_w \cdot \frac{\partial E}{\partial w_i} \tag{7}$$

Eq.7 shows respective $(t+1)^{th}$ values of tuning parameter where k_w is constant. The learning rules Eq.7 changes adaptively the tuning parameters for a direction to minimize the objective function E. Thus, by the learning rules of Eq.7 the tuning parameter of inference rules is optimized to minimize the inference error between the desirable output y^p and the output of fuzzy reasoning y.

By applying I/O data to the fuzzy rule repeatedly, we can minimize the objective function and acquire a global fittest solution without falling in to the local minimum value. Conventional self-tuning methods need many experiments, trials and error process in order to search for optimal rules. This paper uses the GA and the GDM based method to acquire the optimal rules.

3. SELECTING OF FUZZY RULES

This section explains a method of optimizing the membership function shape and the number of fuzzy inference rules using GA. The real numbers of the consequent parts of the fuzzy rules are obtained through the use of the gradient descent method. GA is the optimization technique based loosely on the principles of natural selection. GA starts with a set of encoded parameter strings and an evaluation function of the performance corresponding to each string. Then, through the operations of reproduction, crossover and mutation, the GA attempts to introduce increasingly fit strings into the set. Mutation is added after crossover to expand the region of points that can possibly enter as members of the population. The GA chooses the string with the maximum fitness function $E(s_r)$. Each string is represented as a binary number. A set of string S, called the population, can be represented as (8,9)

$$s_r = L_{r1}, L_{r2}, L_{rg} \ (g = 1, \ldots, G) \tag{8}$$

$$S = \{ s_1, s_2, \cdots, s_R \} \tag{9}$$

3.1 Generation of fuzzy rules

This paper considers the two-class classification problem shown in Fig. 2. The figure shows example of generated problem instances for the eq. $f(x) = 1/4\sin(2\pi x_1) + x_2 - 0.5$. The pattern space $[0,1] * [0,1]$ is divided into two classes according to the value of the following function $f(x) = 1/4\sin(2\pi x_1) + x_2 - 0.5$. If a pattern has $f(x) \geqq 0$, then x belongs to G1, otherwise x belongs to G2. Closed circles and open circles in Fig. 2 represent patterns of G1 and G2 respectively. As learning data, we divided the group numbers of M (G1, G2, ..., GM) and supposed the pattern number of M($x_p=(x_{p1}, x_{p2}, \ldots, x_{pm})$, p=1, 2, ..., m). Then each dimension of 2-dimensional pattern space consists of a fuzzy set of the number of k.

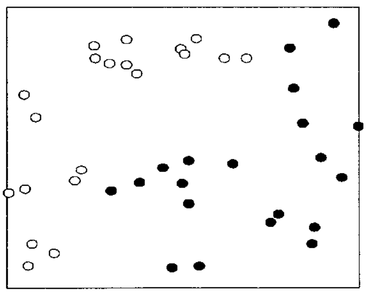


Fig. 2. A classification problem

[Generation method for fuzzy rules]

[Step 1] Determination of generation of regions for the fuzzy rules.

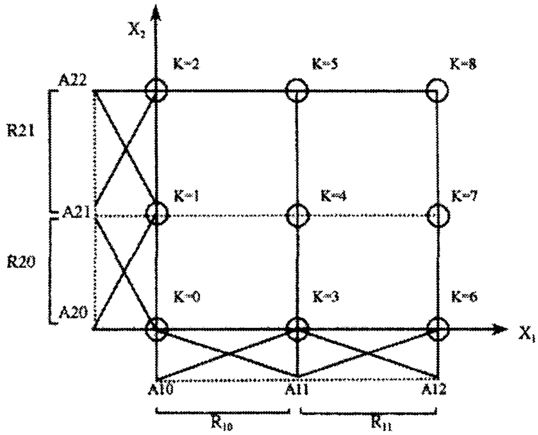


Fig. 3. Examples of fuzzy rule generating regions

A rule is generated from the regions with the biggest inference error, which is calculated in an each area partitioned by membership functions in neighboring two-antecedent part. Fig. 3 shows an example for rule generation regions. The three-membership function in Fig. 3 is a fuzzy set with the input variables x_1 and x_2. So the nine rules in total specified by the notation 0 are set. An inference error has four areas; that is to say, it is calculated from R_{10}, R_{11} for x_1, R_{20} and R_{21} for x_2. The region for rule generation is also selected for each input variable.

[Step 2] Generation of the Membership function of the antecedent part

Inference errors have to be calculated in the four regions, i.e., R_{10} and R_{11} for x_1, and R_{20} and R_{21} for x_2, and selected the region having highest error for rule generation. The region for rule generation might be selected for each input variables. A membership function is generated dividing the region determined by (step 1) into two equal regions by its center value a_{ij}. And then, the membership function number is renumbered in an order of its smallest number. Fig. 4 shows an example of the generation of the membership function in the region R_{10} shown in Fig. 3. In the case, three rules specified by a notation are newly generated.

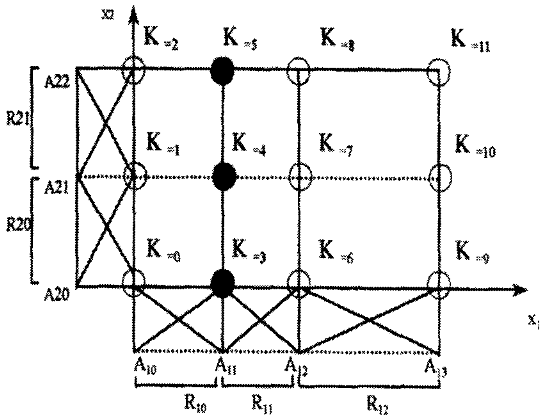


Fig. 4. Example of the generated Rules

[Step 3] Generation of the real numbers of the consequent part

Generation of the real numbers is achieved by means of the gradient descent method of eq. 7. The membership functions of an antecedent part are represented in Fig. 5.

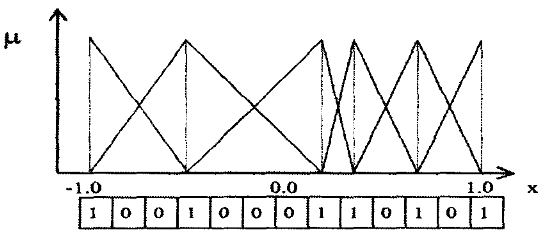


Fig. 5. String representation of membership functions

The binary representation of the membership function consists of strings of "1," and "0,". If the position of bit strings is a center value, it is represented as "1," and other as "0,". The GA searches the numbers and its center values of optimal membership functions to each input variable x_j. The fitness $E(s_r)$ which is able to maximize the center value and the number of membership functions are searched for using a GA as follows.

A. A fitness definition

This paper use the learning pattern number $C(s_r)$ that representing the number of patterns that are classified by the rule set in a evaluation function employed for solving a classification problem using the GA. The number of rule set is represented as $|s_r|$, the fitness function is defined by the maximum $C(s_r)$ and minimum $|s_r|$. The fitness function $E(s_r)$ can shown as the equation(10).

$$E(s_r) = \max\{W_c \cdot C(s_r) - Ws_r \cdot |s_r|\} \qquad (10)$$

B. Definition of a entity

In order to find the optimal solution by using the GA, the entities have to be expressed as strings in order to find an executable solution. In this paper, the number of rules and the membership function are represented by several long strings such as Fig. 5, which are treated as entities.

C. Operations of the GA

This paper uses a simple genetic algorithm that extracts the rule by using an elite preservation strategy. The GA consists of five basic operations. The following operations are applied to a set of individuals (i.e., the population in a generation) in order to generate a new population of the next generation:

(1) Creation of an initial population

The center values of the membership functions and the widths of the neighboring membership functions are initialized as $a_{ij}=1$ and $b_{ij}=0$ respectively. To evaluate the initial population, the function representing the environment is evaluated for the values encoded in each of the n strings in the initial population p(0).

(2) Selection operation

To create the population of next generation by a crossover operation, a selection probability $P_{sr}(t)$ is represented as follow(11).

$$P_{sr}(t) = \frac{E(s_r(t))}{\sum_{r=1}^{R} E(s_r(t))} \qquad (11)$$

(3) Crossover operation

The operation is described as follows: only two strings in a generation are selected. And two positions in the string, that constructs the base of a membership function, are also selected. The string values between these two positions are exchanged between the two string. The offspring generate four species by means of crossover operation. The way to generate offspring 1 is as follows: The values of the left side of the string are inherited from one string (parent 1) and the values on the other side are inherited from the other string (parent 2), where each according to crossover probability value. The new generation the group entity is iterative generated by this crossover operation.

(4) Mutation operation

The mutation operation in this paper operates in such a way that the selected membership function in pruned at probability P_m. One can expect to reduce the number of the fuzzy rules and obtain the maximum of effectiveness in solving classification problem by the mutation operation. The following mutation operation is applied to each bit of the individuals(bit-strings) generated by the crossover operation: $s_r \rightarrow s_r * (-1)$. Each bit of each individual undergo this mutation operation with the mutation probability P_m.

(5) Elite preservation strategy

The best individual (i.e., the individual with the largest fitness value) in each generation always survives and exists in the next generation. This elite preservation strategy has the merit that superior strings do not undergo the GA's operations.

3.2 Self-tuning Procedure

The procedure to acquire an optimal fuzzy rule using a GA is as follows in Fig 6.

Step 1: Randomly generate all entities (population) $s_r(t)$, r=1, ... , R about a initial generation (t = 0).

Step 2: Decide the initial real number of the consequent part, using the gradient descent method.

Step 3: Select two entities $s_i(t)$ and $s_j(t)$ from population S(t) according to selection probabilities $P_{si}(t)$ and $P_{sj}(t)$.

Step 4: Perform a crossover operation to create a new entity $s_k'(t)$ from the two selected entities.

Step 5: Perform a mutation operation on a string of an entity $s_k(t)$, using the mutation probability P_m.

Step 6: Until the new number of entities k, becomes equal to R, repeat from step 3 to step 5.

Step 7: A new group (population) $S(t+1)=\{s_1'(t), s_2'(t), \ldots, s_R'(t)\}$ is generated from step 3 to step 6.

Step 8: Add 1 to the generation number t, and repeat from step 2 to step 8 until the population S converges. The largest fitness entity in the converged group becomes an optimal solution.

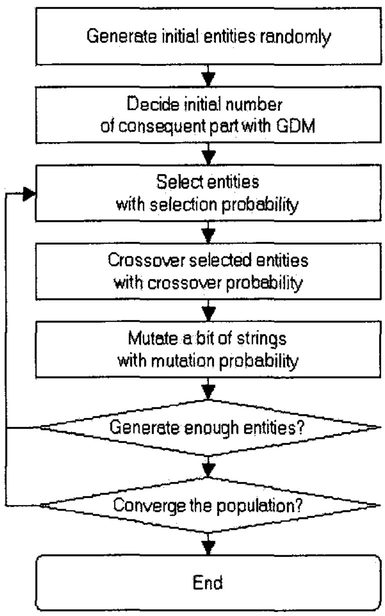


Fig. 6. Steps for GA and GDM based fuzzy learning

4. RESULTS OF COMPUTER SIMULATIONS

In this section, the new method is compared with Ishibuchi's method by means of an experiment[4].

4.1 Rule selection and pattern classification by a fuzzy grid

A classification problem is applied to Fig. 2. Ishibuchi divided the fuzzy space into subspaces using a fuzzy grid and spaces generating fuzzy rules. Fig. 7 shows to be achieved fuzzy division to six-fuzzy-set each dimension of two dimension pattern spaces.

If the fuzzy division is small (k is larger), many patterns may be classified and many fuzzy rules can be generated from them, but the performance of fuzzy classification system may be low. On the other hand, if the fuzzy division is large (k is smaller), many fuzzy rules can not be generated because of the lack of training patterns in the corresponding fuzzy subspaces.

4.2 Rule selection and pattern classification by GA

A genetic algorithm with the following parameter specification is applied to the two-class classification problem in Fig. 2.

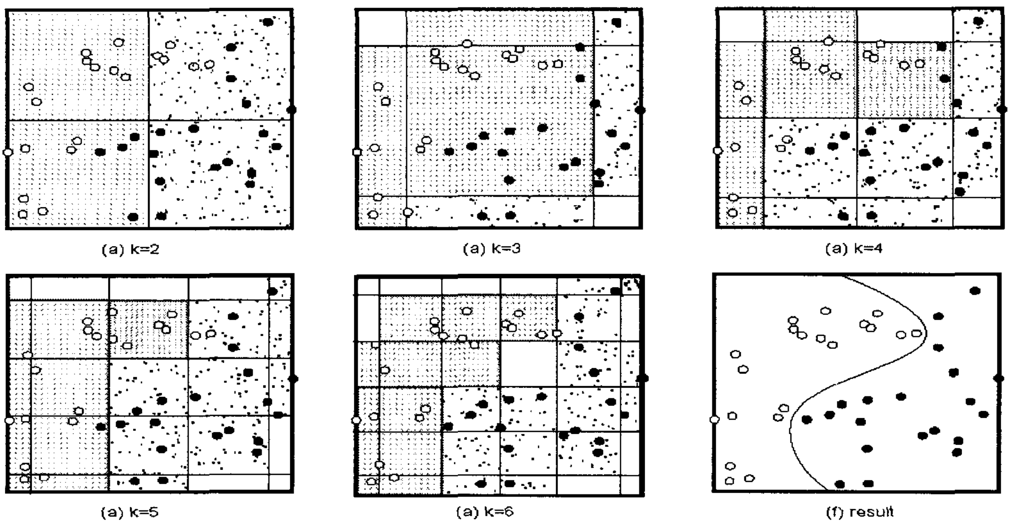
Population size in each generation(R): 20 individuals

Stopping Condition(t) : 1000 generations after, Mutation Probability(P_m) : 0.01

Length of entity(G) : 13, Value of critical : $1.0*10^{-5}$

From the classification results of Fig. 7, when k equals 6, it is possible to see that the whole pattern can be classified. In this case, the total number of rules is 60. The executable numbers of the solution becomes $=1.2*10^{17}$. If a GA is applied, then a total of 2000 rules are generated (20 individual *1000 generations). After these values were set, the first population was randomly changed 20 times. Fig. 8 shows the rules created and the classification results, using a numerical value experiment.

Comparing Fig. 7. with Fig. 8, it is possible to understand the pattern classifications when k equals 4 with even smaller than in Fig. 7. The areas (a)-(c) in Fig. 8 show the fuzzy rules generated by a GA, where (d) is the result of classifying the whole learning pattern. The method was there applied to some searched classification problems as shown in the next section.



Black area: the rules of the first group (white dots): G1 Gray area: the rules of the second group (black dots): G2

Fig. 7. Generated fuzzy rules and classification results

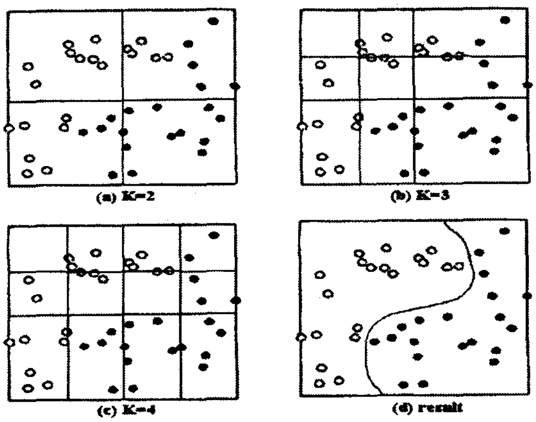


Fig. 8. Simulation results using the proposed method

4.3 Test problems

As test problems, five classification problems were selected. In each problem, the pattern space [0,1]*[0,1] is divided into two classes according to the value of the following function f(x), i.e., if f(x) (0 then x belongs to G1, otherwise x belongs to G2.

Problem 1: $f(x) = -1/4\sin(2\pi x_1) + x_2 - 0.5$

Problem 2: $f(x) = -1/3\sin(2\pi x_1) + x_2 - 0.5$

Problem 3: $f(x) = -1/3\sin(2\pi x_1 - 1/2\pi) + x_2 - 0.5$

Problem 4: $f(x) = -| -2x_1 + 1| + x_2$

Problem 5: $f(x) = (x_1 + x_2 - 1)(-x_1 + x_2)$

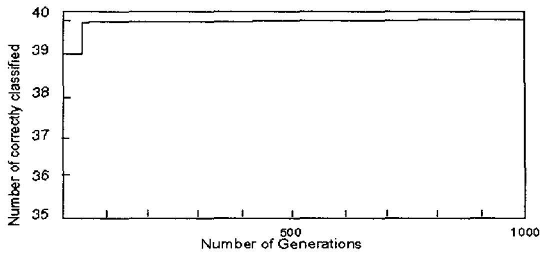


Fig. 9. The number of correctly classified patterns by the best individual in each generation

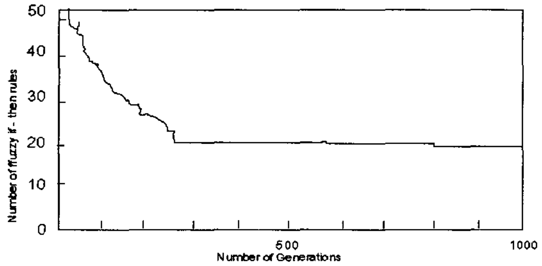


Fig. 10. The number of fuzzy rules by the best individual in each generation

For each classification problem, 20 problem instances were randomly generated, where each of 10 problem instances has 20 patterns in each class as the given patterns (i.e., as the training patterns). Closed circles and open circles in the given patterns belong to G1 and G2, respectively. These patterns are used for deriving the fuzzy rules in

Table 1. The number of classified patterns and fuzzy rules

Conventional method(Problem 1)			Suggested method(Problem 1)		
Number of patterns	Number of Generated Rules	Number of Classified	Number of patterns	Number of Generated Rules	Number of Classified
20	46	20	20	12	20
40	60	40	40	25	40
100	92	100	100	37	100
200	117	200	200	58	200
Conventional method(Problem 5)			**Suggested method(Problem 5)**		
Number of patterns	Number of Generated Rules	Number of Classified	Number of patterns	Number of Generated Rules	Number of Classified
20	42	20	20	10	20
40	58	40	40	22	40
100	96	100	100	34	100
200	110	200	200	50	200

computer simulations. Table 1. shows the usefulness of the suggested method (e. g., in Problem 1).

In the application to the iris data, was the following biased mutation probability employed in order to reduce the number of fuzzy rules by the mutation operation: $P_m = 0.01$ for the mutation from $S_r = 1$ to $S_r = -1$, and $P_m = 0.001$ for the mutation from $S_r = -1$ to $S_r = 1$.

5. CONCLUSIONS

This paper proposes a GA and GDM-based method for removing the unnecessary rules and selecting the relevant rules from the fuzzy rules corresponding to several fuzzy partitions. The aim of the proposed method is to find a minimum set of fuzzy rules, which can correctly classify all the training patterns. This is achieved by formulating and solving a combinatorial optimization problem that has two objectives: to maximize the number of correctly classified patterns, and to minimize the number of fuzzy rules. A fine fuzzy division needs to be chosen, in consist of good fuzzy value classification system. As a solution to this problem, a concept of rules using a large number of fuzzy divisions simultaneously, and a fuzzy classification system can be employed. This method solves the problem of generating a fine fuzzy division but degrades the system efficiency on account of using many of fuzzy rules in inference. In the proposed method, the fuzzy rules corresponding to various fuzzy partitions are simultaneously utilized in the fuzzy inference process. Fuzzy rules are generated from an area with the highest inference errors among those areas which are divided by two membership functions of neighboring antecedent part. The fuzzy of inferences are structured by a set of simple fuzzy rules. In each rule, the antecedent part is made up the membership functions of a fuzzy set, and the consequent part is made up of a real number. The membership functions and number of fuzzy inference rules are tuned by means of the GA, while the real numbers in the consequent parts of the rules are tuned by means of the gradient descent method. A combinatorial optimization problem was formulated for finding a minimum set of fuzzy rules that can correctly classify all the given patterns. The GA and GDM were applied to this problem, and simulation results were shown. But we had only one among probability values of mutation and took a our examination. Future work will involve studying the effects of various probability values on classification problems.

6. REFERENCES

[1] K. H Lee, K. R Oh, Fuzzy Theory and Applications I-II, Hongrung Press, 1991.

[2] H. Ichihshi and T. Watanabe, "Learning control system by a simplified fuzzy reasoning model", IPMU'94, Paris-France, July 2-6, pp. 417-419, 1994.

[3] Shigeo and Ming-Shong Lan, "A method for fuzzy rules extraction directly from numerical data and its application to pattern classification", IEEE Transactions on Fuzzy System, Vol. 3, No. 1, pp. 18-28, 1995.

[4] H. Ishibuchi, K. Nozaki and H. Tanaka, (1992), "Efficient fuzzy partition of pattern space for classification problems." Proc. of the Second International Conference on Fuzzy Logic & Neural Networks (Iizuka, JAPAN), pp. 671-674, 1992.

[5] H. Ishibuchi, K Nozaki and H. Tanaka, "Distributed representation of fuzzy rules and its application to pattern classification," Fuzzy Sets and Systems, Vol. 52, pp. 21-32, 1992.

[6] H. Ishibuchi, K. Nozaki and R. Weber, "Approximate pattern classification with fuzzy boundary", Proc. of International Joint Conference on Neural Networks, Vol. 52, pp. 21-32, 1992.

[7] D. E. Goldberg, Genetic Algorithms in Search,

Optimization, and Machine Learning. Addison-Wesley, Reading, Massachusetts, 1989.

[8] Thrift P., "Fuzzy logic synthesis with genetic algorithms", Proc. of the 12th International Conference on Genetic Algorithm, San Diego, USA, pp. 509-513, 1999.

[9] C. L. Karr, "Design of an adaptive fuzzy logic controller using a genetic algorithm," Proc. of the 12th International Conference on Genetic Algorithms, pp. 450-457, 1999.

[10] Karr C., "Genetic algorithms for fuzzy controllers", AI Expert, February, pp. 26-33, 1998.

[11] H. Nomura, I. Hayashi and N. Wakami, "A self-tuning method of fuzzy control by descent method", Proc. of 4^{th} IFSA Congress, Brussels, pp. 155-158, 1991.

[12] M. Maeda and S. Murakami, "Self-tuning fuzzy logic controller", Transactions of the society of Instrument and Control Engineers, Vol. 34, No. 2, pp. 191-197, 1998.

[13] H. B. Carry, "The method of steepest descent for nonlinear minimization problems", Quart. J. Appl. Math., 2, pp. 258-261, 1994.

[14] D. Park, A. Kandel and G. Langholz, "Genetic-based new fuzzy reasoning models with application to fuzzy control," IEEE Trans. Syst. Man Cybern., Vol. 24, no. 1, pp. 39-47, 1994.

Sangho Choi

He received his B.S. degree in Computer Science from Chung-Ang University, Seoul, Korea, in 1995 and the M.S degree in Computer Science and Engineering from Choong-Ang University, in 1997. At present, he works for SK Communications, the IT and Internet portal Company. His research interests include fuzzy system, artificial life, machine learning, artificial neural networks

Kyungdal Cho

He received his B.S. degree in Computer Science from Kyonggi University, Kyonggi, Korea, in 1990.
He received the M.S and Ph. D. degree in Computer Science and Engineering from Choong-Ang University, Seoul, Korea, in 1992 and 2004, respectively. His research interests include Fuzzy System, Artificial Life, Machine Learning, Artificial Neural Metworks.

Sajoon Park

He received his B.S. degree and M.S degree in Computer Science from Chung-Ang University, Seoul, Korea, in 1995 and 1997, respectively. After M.S degree he had worked for Software Research Center, Hyundai Electronics. He recieved Ph.D degree in Computer Science and Engineering from Chung-Ang University, Seoul, Korea in 2003. His research interests include Fuzzy System, Semantic Web, Artificial Life, Machine Learning, Neural Network.

Malrey Lee

She is the professor in Division of Electronics & Information Engineering at Chonbuk National University from 2004. She received her M.S degree in Computer Science from Chung-Ang University, Seoul, Korea, and in 1993, and Ph.D degree in Computer Science and Engineering from Chung-Ang University in 1997. Until 2003, she was the professor in the Deparment of Multimedia at Yosu National University. Her research interests include Fuzzy System, Artificial Life, Machine Learning, Neural Network.

Kitae Kim

He is a professor in the Department of Computer Science and Engineering at the Chung-Ang University. He received his Ph.D degree from the Department of Computer Science and Engineering at the Soongsil University in 1986. His research interests include Fuzzy System, Artificial Life, Machine Learning, Neural Network.