

Selecting Fuzzy If-Then Rules for Classification Problems Using Genetic Algorithms

Hisao Ishibuchi, *Member, IEEE*, Ken Nozaki, Naohisa Yamamoto, and Hideo Tanaka, *Member, IEEE*

Abstract— This paper proposes a genetic-algorithm-based method for selecting a small number of significant fuzzy if-then rules to construct a compact fuzzy classification system with high classification power. The rule selection problem is formulated as a combinatorial optimization problem with two objectives: to maximize the number of correctly classified patterns and to minimize the number of fuzzy if-then rules. Genetic algorithms are applied to this problem. A set of fuzzy if-then rules is coded into a string and treated as an individual in genetic algorithms. The fitness of each individual is specified by the two objectives in the combinatorial optimization problem. The performance of the proposed method for training data and test data is examined by computer simulations on the iris data of Fisher.

I. INTRODUCTION

FUZZY logic has been mainly applied to control problems with fuzzy if-then rules [1]–[2]. In most fuzzy control systems fuzzy if-then rules were derived from human experts. Recently several approaches have been proposed for automatically generating fuzzy if-then rules from numerical data (for example, see [3]–[5]). Self-learning methods have been also proposed for adjusting membership functions of fuzzy sets in fuzzy if-then rules (for example, see [6]–[10]). Since fuzzy control systems in these methods can be trained by learning schemes of neural networks, they are often called fuzzy neural networks [10]. Michalski and Chilausky [11] was one of the first attempts to generate if-then rules (and also other rules) from the given data in the field of many-valued logics.

Genetic algorithms [12]–[13] have been also employed for generating fuzzy if-then rules and adjusting membership functions of fuzzy sets. For example, membership functions were adjusted by genetic algorithms in Karr [14] and Karr and Gentry [15]. Fuzzy partitions of input spaces were determined in Nomura *et al.* [16]. That is, both the number of fuzzy sets and the membership function of each fuzzy set were determined. In Thrift [17], an appropriate fuzzy set in the consequent part of each fuzzy if-then rule was selected. These approaches applied genetic algorithms to fuzzy control problems by coding a fuzzy rule table (i.e., a set of fuzzy if-then rules) as an individual. On the other hand, Valenzuela-Rendon [18] proposed a fuzzy classifier system where a single fuzzy if-then rule was coded as an individual. Appropriate fuzzy sets in the antecedent and consequent parts of each fuzzy if-then rule were selected by the fuzzy classifier system in [18].

Manuscript received July 9, 1993; revised October 14, 1994.

The authors are with the Department of Industrial Engineering, College of Engineering, University of Osaka Prefecture, 1-1 Gakuencho, Sakai, Osaka 593, Japan.

IEEE Log Number 9410135.

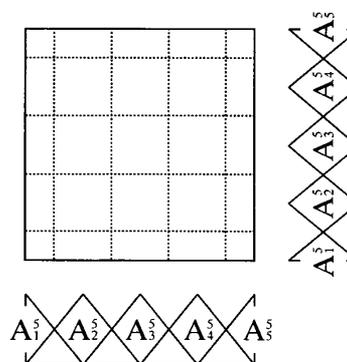


Fig. 1. An example of the fuzzy partition by a simple fuzzy grid.

While various methods have been proposed for generating fuzzy if-then rules and adjusting membership functions, only a few approaches have dealt with classification problems. Ishibuchi *et al.* [19] proposed a generation method of fuzzy if-then rules from numerical data for classification problems. Generation of fuzzy if-then rules from numerical data consists of two phases: fuzzy partition of a pattern space into fuzzy subspaces and determination of a fuzzy if-then rule for each fuzzy subspace. In [19], the fuzzy partition by a simple fuzzy grid was employed. An example of such a fuzzy partition is shown in Fig. 1 where a two-dimensional pattern space is divided into $5^2 = 25$ fuzzy subspaces. The performance of a fuzzy classification system based on fuzzy if-then rules depends on the choice of a fuzzy partition. If a fuzzy partition is too coarse, the performance may be low (i.e., many patterns may be misclassified). If a fuzzy partition is too fine, many fuzzy if-then rules cannot be generated because of the lack of training patterns in the corresponding fuzzy subspaces. Therefore the choice of a fuzzy partition is very important.

Let us consider a two-class classification problem in Fig. 2 where closed circles and open circles denote the patterns in Class 1 and Class 2, respectively. For this classification problem, a fine fuzzy partition is required for the left half of the pattern space but a coarse fuzzy partition is appropriate for the right half. Therefore the choice of an appropriate fuzzy partition based on a simple fuzzy grid is difficult for such a classification problem as shown in Fig. 2.

To cope with this difficulty, the concept of distributed fuzzy if-then rules was proposed in [19] where all fuzzy if-then rules corresponding to several fuzzy partitions were simultaneously employed in fuzzy inference. That is, multiple fuzzy rule tables were simultaneously employed in a single

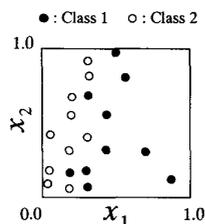


Fig. 2. A classification problem.

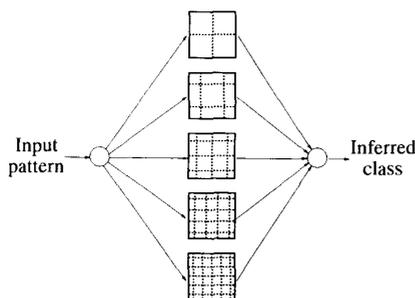


Fig. 3. A fuzzy classification system based on multiple fuzzy rule tables.

fuzzy classification system as shown in Fig. 3. In Fig. 3, a single fuzzy classification system consists of $90 (= 2^2 + 3^2 + 4^2 + 5^2 + 6^2)$ fuzzy if-then rules corresponding to five fuzzy rule tables. Since fuzzy if-then rules corresponding to coarse fuzzy partitions as well as fine fuzzy partitions are simultaneously employed in a single fuzzy classification system, this approach remedies the above-mentioned difficulty in choosing an appropriate fuzzy partition. The main drawback of this approach is that the number of fuzzy if-then rules becomes enormous especially for classification problems with high-dimensional pattern spaces.

If unnecessary fuzzy if-then rules are removed and relevant fuzzy if-then rules are selected, the performance of the selected rule set may be high with much fewer fuzzy if-then rules. This paper proposes a genetic-algorithm-based method for removing unnecessary rules from fuzzy if-then rules corresponding to several fuzzy partitions. The aim of the proposed method is to find a compact rule set with high classification power. A compact fuzzy classification system based on a small number of fuzzy if-then rules has the following advantages:

- i) It does not require a lot of storage.
- ii) The inference speed for new patterns is high.
- iii) Each fuzzy if-then rule can be carefully examined by users.

The last advantage suggests the possibility of knowledge acquisition. When a small number of fuzzy if-then rules are selected, it is possible that a user acquires new knowledge by carefully examining the selected rules. On the contrary, if hundreds of fuzzy if-then rules are included in a fuzzy classification system, it is a quite troublesome task to examine all the rules.

In this paper, a generation method of fuzzy if-then rules for classification problems is first described. Next the rule

selection problem to construct a compact fuzzy classification system is formulated as a combinatorial optimization problem. This problem has two objectives: to maximize the number of correctly classified patterns and to minimize the number of fuzzy if-then rules. Genetic algorithms are applied to this problem. A set of fuzzy if-then rules is coded as an individual in genetic algorithms. The fitness of each individual is specified by the two objectives in the combinatorial optimization problem. Last the performance of the proposed method for training data and test data is examined by computer simulations on the iris data of Fisher [20].

II. FUZZY CLASSIFICATION METHOD WITH FUZZY IF-THEN RULES

In this section, we briefly describe the fuzzy classification method based on fuzzy if-then rules proposed in Ishibuchi *et al.* [19]. Let us assume that a pattern space is the unit square $[0, 1] \times [0, 1]$ for the simplicity of notation. The extension to the case of more general multidimensional pattern spaces will be discussed in Section IV. Suppose that m patterns $\mathbf{x}_p = (x_{p1}, x_{p2})$, $p = 1, 2, \dots, m$, are given as training patterns from M ($M \ll m$) classes: Class 1 ($C1$), Class 2 ($C2$), \dots , Class M (CM). That is, the classification of each \mathbf{x}_p , $p = 1, 2, \dots, m$, is known as one of M classes. Our problem is to generate fuzzy if-then rules that divide the pattern space into M disjoint decision areas.

Let us assume that each axis of the pattern space is partitioned into K fuzzy subsets $\{A_1^K, A_2^K, \dots, A_K^K\}$ where A_i^K is the i th fuzzy subset (the superscript K is attached to indicate the number of fuzzy subsets on each axis, i.e., K denotes the fineness of a fuzzy partition). We can use any type of membership functions (e.g., triangular, trapezoid and exponential) for A_i^K . In this paper, the following symmetric triangular membership function is employed for A_i^K , $i = 1, 2, \dots, K$

$$\mu_i^K(x) = \max\{1 - |x - a_i^K|/b^K, 0\}, \quad i = 1, 2, \dots, K \quad (1)$$

where

$$a_i^K = (i - 1)/(K - 1), \quad i = 1, 2, \dots, K, \quad (2)$$

$$b^K = 1/(K - 1). \quad (3)$$

If we use the same fuzzy subsets $\{A_1^K, A_2^K, \dots, A_K^K\}$ for the two axes of the pattern space, we have K^2 fuzzy subspaces $A_i^K \times A_j^K$, $i = 1, 2, \dots, K$; $j = 1, 2, \dots, K$. The fuzzy partition of the pattern space corresponding to $K = 5$ is shown in Fig. 1 where the pattern space is divided into 25 fuzzy subspaces. In this paper, we use the same fuzzy subsets $\{A_1^K, A_2^K, \dots, A_K^K\}$ for the two axes of the pattern space because the domain intervals of the two attribute values x_{p1} and x_{p2} are the same. If the two domain intervals are totally different from each other, we should use different fuzzy subsets for the two axes. This issue will be discussed in Section V.

Since each fuzzy subspace has one fuzzy if-then rule, the number of fuzzy if-then rules in a single fuzzy rule table is also K^2 . For M -class classification problems in the pattern

space $[0, 1] \times [0, 1]$, we use the following fuzzy if-then rules corresponding to the K^2 fuzzy subspaces.

$$\begin{aligned} \text{Rule } R_{ij}^K; & \text{ If } x_{p1} \text{ is } A_i^K \text{ and } x_{p2} \text{ is } A_j^K \\ & \text{then } \mathbf{x}_p \text{ belongs to Class } C_{ij}^K \\ & \text{with } CF = CF_{ij}^K, \\ & i = 1, 2, \dots, K; j = 1, 2, \dots, K \end{aligned} \quad (4)$$

where R_{ij}^K is the label of the fuzzy if-then rule, A_i^K and A_j^K are triangular fuzzy subsets on the unit interval $[0, 1]$, C_{ij}^K is the consequent (i.e., one of M classes), and CF_{ij}^K is the grade of certainty of the fuzzy if-then rule.

The consequent C_{ij}^K and the grade of certainty CF_{ij}^K of the fuzzy if-then rule in (4) are determined by the following procedure.

Procedure 1—Generation of Fuzzy if-then Rules:

Step 1: Calculate β_{CT} for each class $T (T = 1, 2, \dots, M)$ as

$$\beta_{CT} = \sum_{\mathbf{x}_p \in CT} \mu_i^K(x_{p1}) \cdot \mu_j^K(x_{p2}) \quad (5)$$

where β_{CT} is the sum of the compatibility of \mathbf{x}_p 's in Class T to the fuzzy if-then rule R_{ij}^K in (4).

Step 2: Find Class $X(CX)$ such that

$$\beta_{CX} = \max\{\beta_{C1}, \beta_{C2}, \dots, \beta_{CM}\}. \quad (6)$$

If two or more classes take the maximum value in (6) or all the β_{CT} 's are zero, the consequent C_{ij}^K of the fuzzy if-then rule corresponding to the fuzzy subspace $A_i^K \times A_j^K$ can not be determined uniquely. In this case, let C_{ij}^K be ϕ . If a single class takes the maximum value in (6), C_{ij}^K is determined as CX in (6).

Step 3: If a single class takes the maximum value in (6), CF_{ij}^K is determined as

$$CF_{ij}^K = (\beta_{CX} - \beta) / \sum_{T=1}^M \beta_{CT} \quad (7)$$

where

$$\beta = \sum_{\substack{T=1 \\ T \neq X}}^M \beta_{CT} / (M - 1). \quad (8)$$

In this procedure, the consequent C_{ij}^K is determined as Class $X(CX)$ that has the largest sum of $\mu_i^K(x_{p1}) \cdot \mu_j^K(x_{p2})$ over the M classes in (6). Fuzzy if-then rules with ϕ in the consequent part are dummy rules that have no effect on fuzzy inference for classifying new patterns. If there is no pattern in the fuzzy subspace $A_i^K \times A_j^K$, a dummy rule is generated at that fuzzy subspace because all the β_{CT} 's become zero in this case. The certainty CF_{ij}^K specified by (7)–(8) has the following intuitively acceptable two properties:

- i) If all the patterns in the fuzzy subspace $A_i^K \times A_j^K$ belong to the same class, i.e., if $\beta_{CX} > 0$ and $\beta_{CT} = 0$ for

$T \neq X$ in (6), then $CF_{ij}^K = 1$ (the maximum certainty). In this case, it is certain that any pattern in $A_i^K \times A_j^K$ belongs to the consequent class of the generated fuzzy if-then rule.

- ii) If all the values of β_{CT} 's are not so different from each other, i.e., if $\beta_{C1} \approx \beta_{C2} \approx \dots \approx \beta_{CM}$ in (6), then $CF_{ij}^K \approx 0$ (the minimum certainty). In this case, it is uncertain that a pattern in $A_i^K \times A_j^K$ belongs to the consequent class of the generated fuzzy if-then rule.

Let us denote the set of the generated K^2 fuzzy if-then rules by S^K

$$S^K = \{R_{ij}^K \mid i = 1, 2, \dots, K; j = 1, 2, \dots, K\}. \quad (9)$$

That is, S^K is the rule set corresponding to the $K \times K$ fuzzy rule table. In the approach based on the concept of distributed fuzzy if-then rules, multiple fuzzy rule tables are simultaneously employed as shown in Fig. 3. Let us denote the set of all fuzzy if-then rules corresponding to $K = 2, 3, \dots, L$ by S_{ALL}

$$\begin{aligned} S_{\text{ALL}} &= S^2 \cup S^3 \cup \dots \cup S^L \\ &= \{R_{ij}^K \mid i = 1, 2, \dots, K; j = 1, 2, \dots, K; \\ & \quad K = 2, 3, \dots, L\} \end{aligned} \quad (10)$$

where L is an integer that should be specified depending on the characteristics of each classification problem. The choice of an appropriate value of L may require computer simulations with different values of L . This issue will be discussed in Section IV. By specifying $K = 2, 3, \dots, L$ in Procedure 1, $2^2 + 3^2 + \dots + L^2$ fuzzy if-then rules in S_{ALL} are generated. Let us denote a subset of S_{ALL} by S . The main aim of this paper is to propose a method for selecting fuzzy if-then rules from S_{ALL} to find a compact rule set S with high classification power. This issue will be discussed in the next section.

When a rule set S is given, a new pattern $\mathbf{x}_p = (x_{p1}, x_{p2})$ is classified by the following procedure based on the fuzzy if-then rules in S .

Procedure 2—Classification of a New Pattern $\mathbf{x}_p = (x_{p1}, x_{p2})$:

Step 1: Calculate α_{CT} for each Class $T (T = 1, 2, \dots, M)$ as

$$\begin{aligned} \alpha_{CT} &= \max\{\mu_i^K(x_{p1}) \cdot \mu_j^K(x_{p2}) \cdot CF_{ij}^K \mid C_{ij}^K \\ &= CT \text{ and } R_{ij}^K \in S\}. \end{aligned} \quad (11)$$

Step 2: Find Class $X(CX)$ such that

$$\alpha_{CX} = \max\{\alpha_{C1}, \alpha_{C2}, \dots, \alpha_{CM}\}. \quad (12)$$

If two or more classes take the maximum value in (12) or all the α_{CT} 's are zero, then the classification of \mathbf{x}_p is rejected (i.e., \mathbf{x}_p is left as an unclassifiable pattern), else assign \mathbf{x}_p to Class $X(CX)$ determined by (12).

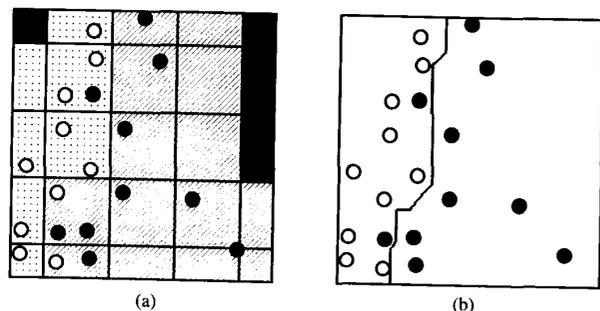


Fig. 4. The rule set S^5 and the classification result. (a) Fuzzy rules. (b) Boundary.

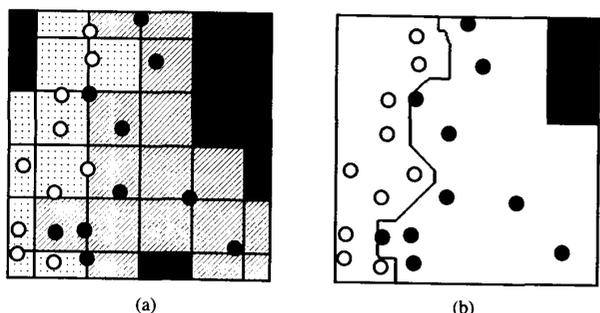


Fig. 5. The rule set S^6 and the classification result. (a) Fuzzy rules. (b) Boundary.

In this procedure, the inferred class is the consequent of the fuzzy if-then rule that has the maximum value of $\mu_i^K(x_{p1}) \cdot \mu_j^K(x_{p2}) \cdot CF_{ij}^K$ over all the fuzzy if-then rules in S .

Using the classification problem in Fig. 2, we explain Procedure 1 and Procedure 2. We applied Procedure 1 with $K = 5$ to the classification problem in Fig. 2. The generated fuzzy if-then rules, i.e., the rule set S^5 in (9), are shown in Fig. 4(a) where hatched areas, dotted areas and painted areas represent the following:

- Hatched area: The consequent of the generated fuzzy if-then rule in this area is Class 1 (closed circles).
- Dotted area: The consequent of the generated fuzzy if-then rule in this area is Class 2 (open circles).
- Painted area: The consequent of the generated fuzzy if-then rule in this area is ϕ , i.e., a dummy rule is generated.

From Fig. 4(a), we can see that 14 rules with Class 1 in the consequent part, seven rules with Class 2 and four dummy rules were generated by Procedure 1. By Procedure 2 with the rule set S^5 in Fig. 4(a), the pattern space was classified into the two disjoint decision areas. The class boundary is shown in Fig. 4(b). From Fig. 4(b), we can see that two patterns from Class 1 were misclassified as Class 2.

In Fig. 5(a), we show the generated fuzzy if-then rules corresponding to $K = 6$. That is, the rule set S^6 is shown in Fig. 5(a). The classification result by the rule set S^6 is shown

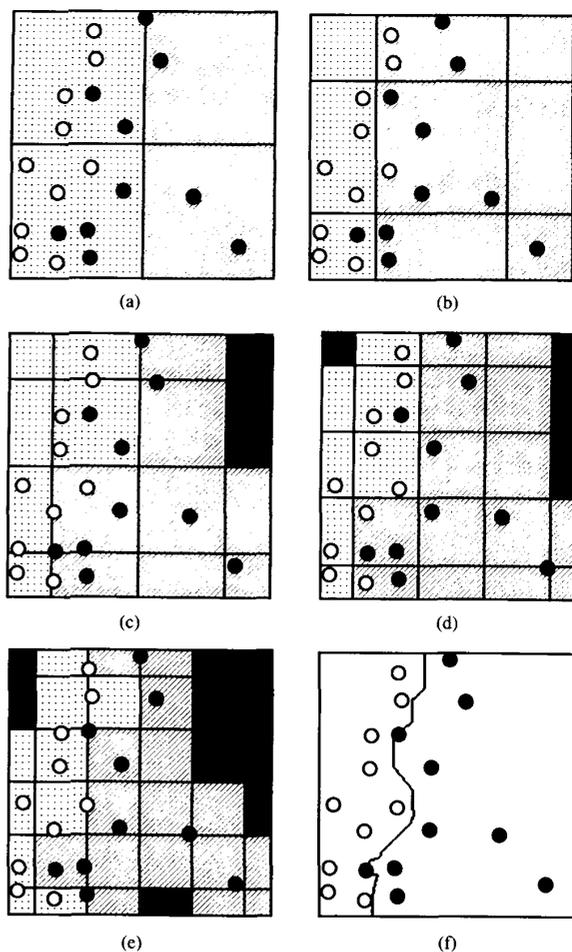


Fig. 6. The rule set S_{ALL} and the classification result. (a) $K=2$. (b) $K=3$. (c) $K=4$. (d) $K=5$. (e) $K=6$. (f) Boundary.

in Fig. 5(b) where a painted area denotes that the classification of a new pattern in this area is rejected. From Fig. 5(b), we can see that all the given patterns were correctly classified.

The rule set S_{ALL} with $L = 6$, i.e., $S_{ALL} = S^2 \cup S^3 \cup S^4 \cup S^5 \cup S^6$, is shown in Fig. 6(a)-(e). The classification result by the rule set S_{ALL} is shown in Fig. 6(f). From Fig. 6(f), we can see that there is no painted rejection area.

III. APPLICATION OF GENETIC ALGORITHMS

A. Formulation of a Combinatorial Optimization Problem

Using Procedure 1 in Section II, the fuzzy if-then rules corresponding to $K = 2, 3, \dots, L$ are generated from the training patterns x_p , $p = 1, 2, \dots, m$. That is, the rule set S_{ALL} in (10) is obtained. Our problem in this section is to select fuzzy if-then rules from S_{ALL} to construct a compact rule set S with high classification power. Therefore the objectives of our problem are to maximize the number of correctly classified patterns by S and to minimize the number of fuzzy if-then rules in S . This problem can be formulated as the following two-objective combinatorial optimization problem.

Problem 1: Maximize $NCP(S)$ and Minimize $|S|$

$$\text{subject to } S \subseteq S_{ALL}$$

where $NCP(S)$ is the number of correctly classified patterns by S and $|S|$ is the cardinality of S (i.e., the number of fuzzy if-then rules in S).

Since Problem 1 has two objectives, it is difficult to directly apply genetic algorithms to this problem. Thus we modify Problem 1 by introducing positive weights W_{NCP} and W_S as follows.

Problem 2: Maximize $f(S) = W_{NCP} \cdot NCP(S) - W_S \cdot |S|$

$$\text{subject to } S \subseteq S_{ALL}.$$

In general, the classification power of a classification system is more important than its compactness. Therefore the weights in Problem 2 should be specified as $0 < W_S \ll W_{NCP}$. The objective function $f(S)$ is treated as the fitness function in genetic algorithms.

B. Coding

In genetic algorithms, each feasible solution of Problem 2 is treated as an individual. That is, a rule set S should be represented by a string. Let us denote a rule set S by a string $s_1 s_2 \cdots s_N$ as $S = s_1 s_2 \cdots s_N$ where

$$N = 2^2 + 3^2 + \cdots + L^2 \quad (\text{i.e., } N \text{ is the total number of rules in } S_{ALL}).$$

$s_r = 1$ denotes that the r th rule belongs to S .

$s_r = -1$ denotes that the r th rule does not belong to S .

$s_r = 0$ denotes that the r th rule is a dummy rule.

The index r of the fuzzy if-then rule R_{ij}^K is determined as

$$r = \begin{cases} 2(i-1) + j, & \text{if } K = 2, \\ \sum_{h=2}^{K-1} h^2 + K(i-1) + j, & \text{if } K \geq 3. \end{cases} \quad (13)$$

The indexes of fuzzy if-then rules corresponding to $K = 2, 3$ are shown in Fig. 7.

Since dummy rules have no effect on fuzzy inference in the classification phase (i.e., on Procedure 2 in Section II), they should be excluded from a rule set S . Therefore they are represented as $s_r = 0$ in this coding to prevent S from including them. A string $s_1 s_2 \cdots s_N$ is decoded as

$$S = \{R_{ij}^K \mid s_r = 1; r = 1, 2, \dots, N\}. \quad (14)$$

C. Operations of Genetic Algorithms

The following genetic operations are employed to generate and handle a set of strings (i.e., a population) in genetic algorithms of this paper.

- i) **Initialization:** Generate an initial population containing N_{pop} strings where N_{pop} is the number of strings in each population. In this operation, each string is generated by assigning zero to dummy rules and randomly assigning one or -1 to the other rules. This initialization operation can be explained as follows. From the viable rule space, a population is chosen. Each string consists of a set of

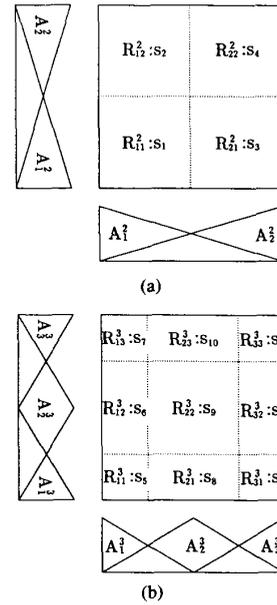


Fig. 7. Labels and indexes of fuzzy if-then rules. (a) $K=2$. (b) $K=3$.

rules. Each rule, which is randomly included in each string, has a 50% chance of being chosen for inclusion in any string. It should be noted that we can use more than a 50% chance of rule inclusion.

- ii) **Selection:** Select $N_{pop}/2$ pairs of strings from the current population. The selection probability $P(S)$ of a string S in a population Ψ is specified as

$$P(S) = \{f(S) - f_{\min}(\Psi)\} / \sum_{S' \in \Psi} \{f(S') - f_{\min}(\Psi)\} \quad (15)$$

where

$$f_{\min}(\Psi) = \min\{f(S) \mid S \in \Psi\}. \quad (16)$$

- iii) **Crossover:** Perform the following one-point crossover operation on the selected $N_{pop}/2$ pairs

$$\begin{array}{c} s_1 s_2 \cdots s_{r-1} s_r s_{r+1} \cdots s_n \quad s_1 s_2 \cdots s_{r-1} \bar{s}_r \bar{s}_{r+1} \cdots \bar{s}_N \\ \rightarrow \\ \bar{s}_1 \bar{s}_2 \cdots \bar{s}_{r-1} \bar{s}_r \bar{s}_{r+1} \cdots \bar{s}_N \quad s_1 s_2 \cdots s_{r-1} s_r s_{r+1} \cdots s_N. \end{array}$$

A crossover point is chosen randomly. By this crossover operation, N_{pop} strings are generated.

- iv) **Mutation:** Perform the following mutation with the mutation probability P_m on each bit of the generated strings by the crossover operation

$$s_r := s_r \times (-1). \quad (17)$$

- v) **Elitist Strategy:** Randomly remove one string from the N_{pop} strings generated by the above operations, and add the best string with the maximum fitness in the previous population to the current one.
- vi) **Termination Test:** If a stopping condition is not satisfied, return to Step ii). The total number of generations is used as a stopping condition in this paper.

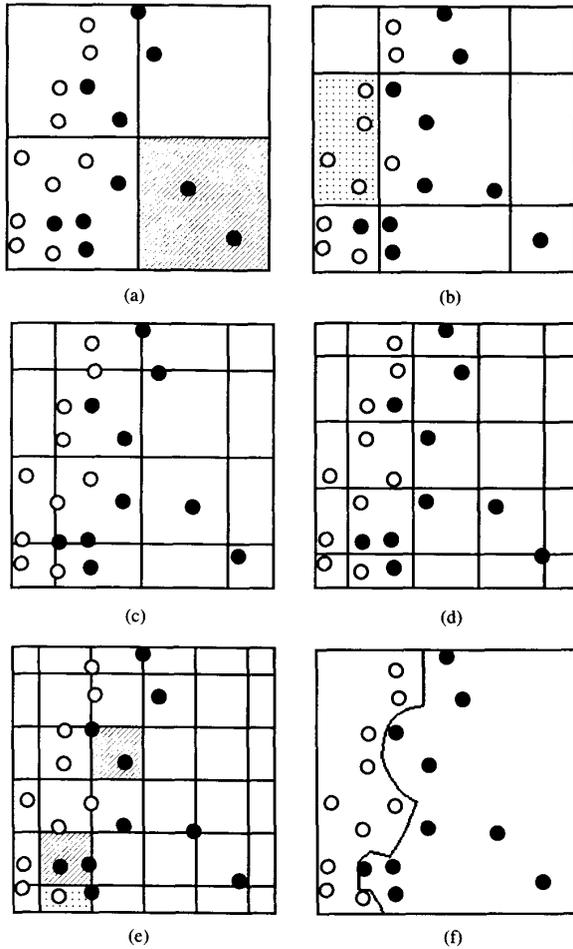


Fig. 8. Selected five fuzzy if-then rules and the classification result. (a) $K=2$. (b) $K=3$. (c) $K=4$. (d) $K=5$. (e) $K=6$. (f) Boundary.

D. Numerical Example

We applied a genetic algorithm with the following parameter specifications to the classification problem in Fig. 2

- Values of weights: $W_{NCP} = 10$, $W_S = 1$,
- Population size: $N_{pop} = 10$,
- Initial population: 50% chance of rule inclusion in each string,
- Mutation probability: $P_m = 0.01$,
- Stopping condition: 1000 generations.

As the rule set S_{ALL} , we used the fuzzy if-then rules in Fig. 6(a)–(e). That is, we specified S_{ALL} as $S_{ALL} = S^2 \cup S^3 \cup S^4 \cup S^5 \cup S^6$. Our problem is to find a compact rule set S with high classification power by selecting fuzzy if-then rules from those in Fig. 6(a)–(e). The number of the fuzzy if-then rules including dummy rules is $2^2 + 3^2 + 4^2 + 5^2 + 6^2 = 90$. Therefore, the length of each string is 90. Since the 90 fuzzy if-then rules in S_{ALL} include 16 dummy rules (see Fig. 6(a)–(e)), the total number of solutions of our problem is

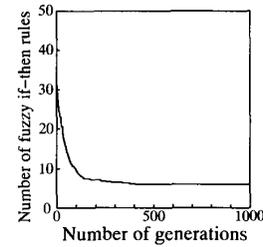


Fig. 9. The number of fuzzy if-then rules by the best individual in each generation (average results over 10 trials).

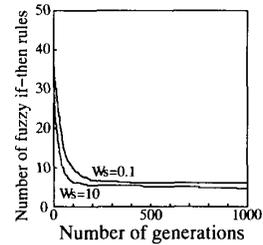


Fig. 10. The number of fuzzy if-then rules by the best individual in each generation (different values of the weight W_S for the number of fuzzy if-then rules).

$2^{74} \approx 1.9 \times 10^{22}$. It should be noted that 2^{74} solutions include some meaningless solutions. For example, a rule set with a single rule is meaningless. Since at least one rule for each of the two classes should be included in a rule set, the number of meaningful solutions of our problem is $(2^{46} - 1) \cdot (2^{28} - 1)$ where 46 and 28 are the numbers of the fuzzy if-then rules with the Class 1 consequent (hatched areas in Fig. 6) and the Class 2 consequent (dotted areas in Fig. 6), respectively.

The genetic operations in Section III-C were applied to this problem. The selected fuzzy if-then rules after 1000 generations and the corresponding classification result are shown in Fig. 8. In Fig. 8(a)–(e), the selected fuzzy if-then rules with the Class 1 and Class 2 consequents are represented by hatched areas and dotted areas, respectively. From Fig. 8(f), we can see that all the training patterns are correctly classified by the selected five fuzzy if-then rules. From the comparison of Fig. 8 with Figs. 4–6, we can observe the effectiveness of the proposed method. All the training patterns were correctly classified in Fig. 8 by much fewer fuzzy if-then rules in comparison with Figs. 4–6. In Fig. 9, we show the number of fuzzy if-then rules by the best individual in each generation. Fig. 9 is not the result of a single trial but the average of ten trials. Fig. 9 shows how the number of fuzzy if-then rules decreased.

To examine the effect of the positive weights W_{NCP} and W_S on the proposed rule selection procedure, we performed the same computer simulation by using different values of the weight W_S for the number of fuzzy if-then rules in S . We examined two parameter specifications: One is $W_{NCP} = 10$ and $W_S = 0.1$, and the other is $W_{NCP} = 10$ and $W_S = 10$.

Average results of 10 trials with each parameter specification are shown in Fig. 10. Fig. 10 shows how the number of fuzzy if-then rules decreased in the same manner as Fig. 9. From

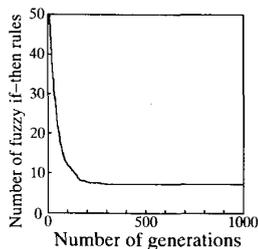


Fig. 11. The number of fuzzy if-then rules by the best individual in each generation (different initial population).

Fig. 10, we can see that the number of fuzzy if-then rules more rapidly decreased with $W_S = 10$ than $W_S = 0.1$.

We also examined the effect of an initial population on the proposed rule selection procedure. We used a 90% chance for each rule of being included in each initial string. This means that we assigned 1 and -1 to each rule with the probabilities of 0.9 and 0.1, respectively. Simulation results with $W_{NCP} = 10$ and $W_S = 1$ are shown in Fig. 11. From the comparison between Figs. 9 and 11, we can see that the number of fuzzy if-then rules in Fig. 11 was larger than that in Fig. 9. This is because initial strings in Fig. 11 included more fuzzy if-then rules than those in Fig. 9.

IV. PERFORMANCE EVALUATION

In this section, the performance of the proposed method for training data and test data is examined by the well-known iris data in Fisher [20]. The iris data consist of the following four-dimensional pattern vectors from three classes:

$$\begin{aligned} \text{Class 1 (Iris setosa): } \mathbf{x}_p &= (x_{p1}, x_{p2}, x_{p3}, x_{p4}), \\ & p = 1, 2, \dots, 50, \\ \text{Class 2 (Iris versicolor): } \mathbf{x}_p &= (x_{p1}, x_{p2}, x_{p3}, x_{p4}), \\ & p = 51, 52, \dots, 100, \\ \text{Class 3 (Iris virginica): } \mathbf{x}_p &= (x_{p1}, x_{p2}, x_{p3}, x_{p4}), \\ & p = 101, 102, \dots, 150 \end{aligned}$$

where x_{p1} is the sepal length, x_{p2} is the sepal width, x_{p3} is the petal length, and x_{p4} is the petal width. In computer simulations of this session, all the attribute values were normalized into real numbers in the unit interval $[0, 1]$ as

$$x_{pi} := (x_{pi} - \min\{x_{pi}\}) / (\max\{x_{pi}\} - \min\{x_{pi}\}), \quad p = 1, 2, \dots, 150; i = 1, 2, 3, 4 \quad (18)$$

where

$$\min\{x_{pi}\} = \min\{x_{pi} \mid p = 1, 2, \dots, 150\}, \quad (19)$$

$$\max\{x_{pi}\} = \max\{x_{pi} \mid p = 1, 2, \dots, 150\}. \quad (20)$$

Therefore the classification problem of the iris data was transformed into a three-class classification problem in the four-dimensional unit cube $[0, 1]^4$.

A. Extension of the Proposed Method to Multidimensional Problems

In the above sections, we explained our method in the two-dimensional pattern space $[0, 1]^2$. Since the classification problem of the iris data has four attributes, our method should be modified. In this subsection, we briefly explain how our method can be extended to the four-dimensional classification problem.

The K^2 fuzzy if-then rules in (4) are extended to the following K^4 rules for the iris data

$$\begin{aligned} \text{Rule } R_{ijkl}^K: \quad & \text{If } x_{p1} \text{ is } A_i^K \text{ and } x_{p2} \text{ is } A_j^K \text{ and} \\ & x_{p3} \text{ is } A_k^K \text{ and } x_{p4} \text{ is } A_l^K \\ & \text{then } \mathbf{x}_p \text{ belongs to Class } C_{ijkl}^K \\ & \text{with } CF = CF_{ijkl}^K, \\ & i = 1, 2, \dots, K; j = 1, 2, \dots, K; \\ & k = 1, 2, \dots, K; l = 1, 2, \dots, K \quad (21) \end{aligned}$$

where R_{ijkl}^K is the label of the fuzzy if-then rule, A_i^K , A_j^K , A_k^K and A_l^K are triangular fuzzy subsets on the unit interval $[0, 1]$, C_{ijkl}^K is the consequent (i.e., one of the three classes), and CF_{ijkl}^K is the grade of certainty of the fuzzy if-then rule. The rule set corresponding to the $K \times K \times K \times K$ fuzzy rule table is denoted by S^K as

$$S^K = \{R_{ijkl}^K \mid i = 1, 2, \dots, K; j = 1, 2, \dots, K; k = 1, 2, \dots, K; l = 1, 2, \dots, K\}. \quad (22)$$

The rule set corresponding to $K = 2, 3, \dots, L$ is denoted by S_{ALL} as

$$S_{ALL} = S^2 \cup S^3 \cup \dots \cup S^L. \quad (23)$$

The rule generation procedure (i.e., Procedure 1) in Section II can be applied to the classification problem of the iris data by modifying the definition of β_{CT} in (5) as

$$\beta_{CT} = \sum_{\mathbf{x}_p \in CT} \mu_i^K(x_{p1}) \cdot \mu_j^K(x_{p2}) \cdot \mu_k^K(x_{p3}) \cdot \mu_l^K(x_{p4}). \quad (24)$$

In the same manner, the fuzzy reasoning procedure (i.e., Procedure 2) in Section II can be applied to the iris data by modifying the definition of α_{CT} in (11) as

$$\alpha_{CT} = \max\{\mu_i^K(x_{p1}) \cdot \mu_j^K(x_{p2}) \cdot \mu_k^K(x_{p3}) \cdot \mu_l^K(x_{p4}) \cdot CF_{ijkl}^K \mid C_{ijkl}^K = CT \text{ and } R_{ijkl}^K \in S\}. \quad (25)$$

The genetic algorithm in Section III can be applied to the iris data by specifying the index r of each of the $N (= 2^4 + 3^4 + \dots + L^4)$ fuzzy if-then rules R_{ijkl}^K 's in S_{ALL} as

$$r = \begin{cases} 2^3(i-1) + 2^2(j-1) + 2(k-1) + l, & \text{if } K = 2, \\ \sum_{h=2}^{K-1} h^4 + K^3(i-1) + K^2(j-1) + K(k-1) + l, & \text{if } K \geq 3. \end{cases} \quad (26)$$

B. Performance for Training Data

Under various parameter specifications in genetic algorithms, the proposed method was applied to the iris data. Three

TABLE I
AVERAGE CLASSIFICATION RATE (%)

Mutation probability	$N_{pop} = 5$	$N_{pop} = 10$	$N_{pop} = 50$
$P_m = 0.1$	99.87	99.87	99.47
$P_m = 0.01$	100.00	99.47	100.00
$P_m = 0.001$	98.67	99.33	99.47
$P_m = 0.0001$	97.87	98.53	99.07
$P_m = P_m^A$	99.20	99.47	99.73
$P_m = P_m^B$	99.33	99.47	99.73
$P_m = P_m^C$	98.27	99.33	99.20

population sizes were examined: $N_{pop} = 5, 10, 50$. The total number of generations was specified as $10000/N_{pop}$, i.e., 2000 generations for $N_{pop} = 5$, 1000 generations for $N_{pop} = 10$, and 200 generations for $N_{pop} = 50$. The weights W_{NCP} and W_S were specified as $W_{NCP} = 10$ and $W_S = 1$. Four mutation probabilities were examined: $P_m = 0.1, 0.01, 0.001, 0.0001$. In addition, the following three types of biased mutations were also examined

$$\begin{aligned} \text{Type A: } P_m^A &= \begin{cases} 0.1, & s_r = 1 \rightarrow s_r = -1, \\ 0.01, & s_r = -1 \rightarrow s_r = 1, \end{cases} \\ \text{Type B: } P_m^B &= \begin{cases} 0.01, & s_r = 1 \rightarrow s_r = -1, \\ 0.001, & s_r = -1 \rightarrow s_r = 1, \end{cases} \\ \text{Type C: } P_m^C &= \begin{cases} 0.001, & s_r = 1 \rightarrow s_r = -1, \\ 0.0001, & s_r = -1 \rightarrow s_r = 1. \end{cases} \end{aligned}$$

In these biased mutations, the probabilities are biased toward the mutation from $s_r = 1$ to $s_r = -1$ (i.e., toward the reduction of the number of fuzzy if-then rules).

First, fuzzy if-then rules corresponding to $K = 2, 3, 4, 5, 6$ were generated by Procedure 1 in Section II from the 150 patterns in the iris data (i.e., we specified L as $L = 6$). The number of the fuzzy if-then rules is $2^4 + 3^4 + 4^4 + 5^4 + 6^4 = 2274$ including 1582 dummy rules. Therefore the length of each string is 2274. Since the 1582 dummy rules are detected in the coding, our problem is to construct a compact fuzzy classification system by selecting significant fuzzy if-then rules from the other 692 rules.

Simulation results by genetic algorithms with the above-mentioned various parameter specifications are summarized in Tables I-III. These tables show the average results over five trials for each value of N_{pop} and P_m . From these tables, we can observe the following.

- i) From Table I: High classification rates over 99% were obtained by almost all the parameter specifications in the case of $N_{pop} = 10$ and $N_{pop} = 50$. The worst classification rate (97.87%) was obtained by the combination of the smallest population size ($N_{pop} = 5$) and the lowest mutation probability ($P_m = 0.0001$).
- ii) From Table II: The number of fuzzy if-then rules was very sensitive to parameter specifications.
- iii) From Table II: Biased mutations ($P_m = P_m^A, P_m^B, P_m^C$) had a significant effect on the reduction of the number of fuzzy if-then rules.
- iv) From Table III: The best fitness value (1479) was obtained by $P_m = P_m^B$ and $N_{pop} = 10$. In this case, the number of fuzzy if-then rules was reduced from 692 to 12.6 (1.82%).

TABLE II
AVERAGE NUMBERS OF FUZZY IF-THEN RULES

Mutation probability	$N_{pop} = 5$	$N_{pop} = 10$	$N_{pop} = 50$
$P_m = 0.1$	260.2	261.2	281.0
$P_m = 0.01$	102.0	130.4	200.0
$P_m = 0.001$	18.0	29.8	168.0
$P_m = 0.0001$	166.6	157.0	212.2
$P_m = P_m^A$	24.2	25.0	42.2
$P_m = P_m^B$	12.4	12.6	32.2
$P_m = P_m^C$	12.8	12.8	110.0

TABLE III
AVERAGE VALUES OF THE FITNESS FUNCTION

Mutation probability	$N_{pop} = 5$	$N_{pop} = 10$	$N_{pop} = 50$
$P_m = 0.1$	1238	1237	1211
$P_m = 0.01$	1398	1362	1300
$P_m = 0.001$	1462	1460	1324
$P_m = 0.0001$	1301	1321	1274
$P_m = P_m^A$	1464	1467	1454
$P_m = P_m^B$	1478	1479	1464
$P_m = P_m^C$	1461	1477	1378

In the above computer simulations, the maximum value of K (i.e., L in the definition of S_{ALL}) was specified as $L = 6$ in consideration of available computation power at hand. Since the length of strings is $N = 2^n + 3^n + \dots + L^n$ in the case of n -dimensional classification problems, N becomes long as the value of L increases. Therefore the computation time for the execution of genetic algorithms also becomes long. Simulation results with different values of L are shown in Table IV. Table IV shows the average results over five trials on a workstation (37 MIPS). In computer simulations, we used the best parameter specifications in the above computer simulations (i.e., $P_m = P_m^B$ and $N_{pop} = 10$). From Table IV, we can see that the computation time drastically increased with the value of L . On the contrary, the performance was not sensitive to the value of L in the range of $L = 5 \sim 8$. Therefore the choice of L (i.e., the choice of S_{ALL}) is not a serious issue from the point of view of the performance.

For comparison, we classified the 150 patterns in the iris data by the fuzzy classification system based on a single fuzzy rule table. That is, the fuzzy if-then rules corresponding to a single value of K were employed for classifying the 150 patterns. Simulation results are summarized in Table V. In Table V, the number of fuzzy if-then rules does not include dummy rules. From the comparison of Table V with Tables I and II, we can see the effectiveness of the proposed method. That is, almost all the patterns were classified by a small number of fuzzy if-then rules in the proposed method (i.e., 99.47% by 12.6 rules on the average in the case of the best parameter specifications) while the best classification rate 98.67% was obtained by 295 fuzzy if-then rules in the classification method based on a single fuzzy rule table.

C. Performance for Test Data

In the above subsection, we have demonstrated high classification power of the proposed method for training data. Since the proposed method selects fuzzy if-then rules based on the performance for training data, it may suffer from the overfitting to training data. To examine the performance of the

TABLE IV
SIMULATION RESULTS WITH DIFFERENT VALUES OF L

L : Maximum number of K	$L=4$	$L=5$	$L=6$	$L=7$	$L=8$
Classification rate (%)	96.67	99.20	99.47	99.47	99.60
Number of fuzzy rules	8.8	11.8	12.6	14.8	19.0
Computation time (min.)	15.7	33.2	68.2	122.3	216.9

TABLE V
PERFORMANCE FOR TRAINING DATA OF THE CLASSIFICATION
METHOD BASED ON A SINGLE FUZZY RULE TABLE

K : Fuzzy Partition	$K=2$	$K=3$	$K=4$	$K=5$	$K=6$
Classification rate (%)	67.33	94.00	92.67	96.00	98.67
Number of fuzzy rules	16	62	129	190	295

proposed method for test data, we performed the leaving-one-out technique [21] for the iris data. In the leaving-one-out technique, a single pattern was left as a test pattern and the other 149 patterns were used as training patterns. This procedure was iterated until all the given 150 patterns were used as a test pattern.

The genetic algorithm with the best parameter specifications in the last subsection (i.e., $P_m = P_m^B$ and $N_{pop} = 10$) was employed to examine the performance of the proposed method for test data by the leaving-one-out technique. The rule set S_{ALL} was generated from 149 training patterns by specifying $K = 2, 3, 4, 5, 6$ in each iteration of the leaving-one-out technique. The following result was obtained after 1000 generations:

Classification rate: 94.67%,
Error rate: 4.00%,
Rejection rate: 1.33%,
Average number of selected rules: 12.90.

For comparison, simulation results by the classification method based on a single fuzzy rule table are shown in Table VI. The last row in Table VI shows the number of fuzzy if-then rules excluding dummy rules. From Table VI, we can see that a large number of fuzzy if-then rules were necessary to high performance in the case of the classification method based on a single fuzzy rule table. From the comparison of Table VI with the above result, we can see that the performance of the proposed method for test data was higher than that of the fuzzy classification method based on a single fuzzy rule table in the case of $K = 2, 3, 4$. In the case of $K = 5, 6$, the classification rate of a single fuzzy rule table was higher than that of the proposed method by 0.66~2.00%, but the single rule table approach requires more than 10 times fuzzy if-then rules in comparison with the proposed method.

A slight deterioration of the generalization ability of the proposed method for test data is due to the overfitting to training data. If our aim is to maximize the classification rate (i.e., to minimize the error rate) for test data, we should carefully choose the number of iterations of the proposed method as in the case of the learning of neural networks. To appropriately specify the number of generations in genetic algorithms, we monitored the performance of the proposed method for test data every 10 generations by using the leaving-one-out technique. From these computer simulations, we observed that the generalization ability of the proposed method was high

TABLE VI
PERFORMANCE FOR TEST DATA OF THE CLASSIFICATION
METHOD BASED ON A SINGLE FUZZY RULE TABLE

K : Fuzzy Partition	$K=2$	$K=3$	$K=4$	$K=5$	$K=6$
Classification rate (%)	67.33	93.33	89.33	95.33	96.67
Error rate (%)	32.67	6.67	10.67	4.67	3.33
Rejection rate (%)	0.00	0.00	0.00	0.00	0.00
Number of fuzzy rules	16.0	62.0	128.9	189.8	294.5

TABLE VII
CLASSIFICATION RATE (%) OBTAINED BY THE PROPOSED METHOD

Number of generations	10	20	30	40	50	100
$P_m = P_m^B, W_{NCP} = 10, W_S = 1$	96.67	96.67	96.67	96.00	96.00	96.00
$P_m = P_m^B, W_{NCP} = 10, W_S = 0.1$	95.33	96.67	97.33*	96.00	97.33*	95.33

* Best result in this table

when the number of generations was small. Simulation results by the leaving-one-out technique with various numbers of generations are shown in Table VII. In Table VII, we show the results with two parameter specifications for the weights: $W_{NCP} = 10, W_S = 1$ and $W_{NCP} = 10, W_S = 0.1$. From the comparison between Tables VI and VII, we can see that the best result (97.33% classification rate and 2.67% error rate) by the proposed method slightly outperforms that of the single rule table approach. In addition, it outperforms all the results reported by Grabisch and Dispot [22] for nine fuzzy classification methods (five methods based on fuzzy pattern matching, three methods based on fuzzy clustering and a fuzzy k -nearest neighbor method). In [22], error rates of the fuzzy classification methods for test data were estimated by the leaving-one-out technique on the iris data as follows:

3.3~8.0% (five methods based on fuzzy pattern matching),
4.7~6.7% (three methods based on fuzzy clustering),
3.3% (fuzzy k -nearest neighbor method).

D. Selected Fuzzy If-Then Rules

Since a small number of significant fuzzy if-then rules are selected by the proposed method, it can be viewed as a knowledge acquisition tool. We applied the genetic algorithm with the best parameter specifications for training data (i.e., $P_m = P_m^B, W_{NCP} = 10, W_S = 1$ and $N_{pop} = 10$) to the iris data. In Fig. 12, we show the selected fuzzy if-then rules after 10000 generations. All the 150 patterns were correctly classified by the selected 13 fuzzy if-then rules. The number of correctly classified patterns by each fuzzy if-then rule is listed in the column labeled patterns.

We can find several characteristic features of the iris data in the selected fuzzy if-then rules. For example, we can see from the comparison of the first three rules with the other rules that the patterns in Class 1 are characterized by small attribute values of the third and fourth attributes (i.e., x_3 and x_4). We can also see that the second attribute (i.e., x_2) of the patterns in Class 1 can take large values as well as small values. It is possible to find these characteristic features by examining the selected rules because the number of fuzzy if-then rules was drastically reduced by the proposed method. If there are hundreds of fuzzy if-then rules in a fuzzy classification system, it is practically impossible to carefully examine all of them.

x_1	x_2	x_3	x_4	Consequent	CF	Patterns*
				Class 1	0.99	44
				Class 1	1.00	5
				Class 1	1.00	1
				Class 2	0.35	4
				Class 2	0.95	9
				Class 2	0.55	3
				Class 2	0.95	17
				Class 2	0.99	5
				Class 2	0.78	8
				Class 2	0.99	4
				Class 3	0.58	43
				Class 3	0.53	1
				Class 3	0.83	6

Patterns* : the number of patterns correctly classified by each fuzzy if-then rule

Fig. 12. Selected fuzzy if-then rules.

V. FUTURE EXTENSIONS

As we have already explained, our genetic-algorithm-based approach to the construction of fuzzy classification systems consists of the following two parts.

Part 1: Generation of a set of fuzzy if-then rules. The rule generation procedure was described in Section II.

Part 2: Selection of significant fuzzy if-then rules from the rule set generated in the first part. The rule selection procedure was described in Section III.

Since the rule selection procedure in the second part can handle any rule set, we do not have to restrict candidate rules within the rule set S_{ALL} defined in (10) or (23). One possible extension of the rule set S_{ALL} is to use different fuzzy subsets for each axis of the pattern space as shown in Fig. 13 where the horizontal axis is divided into five fuzzy subsets while the vertical axis is divided into three. Fuzzy if-then rules corresponding to 5×3 fuzzy subspaces in Fig. 13 can be generated from given training patterns by slightly modifying the rule generation procedure in Section II. The generated rules can be added to the rule set S_{ALL} and treated as candidate rules in the rule selection procedure in Section III. We can also use different types of membership functions for fuzzy partition as shown in Fig. 14 where both triangular and trapezoid membership functions are used. In this way, we can generate enormous fuzzy if-then rules as candidate rules for the rule selection procedure (see also [23]). On the other hand, the computation time required for the execution of the rule selection procedure becomes long as the rule set S_{ALL} becomes large (for example, see Table IV). In addition, since the length of each string in the rule selection procedure is the same as the cardinality of S_{ALL} , the larger S_{ALL} is, the larger the required memory is. Therefore we can not include too many rules in S_{ALL} from a practical point of view.

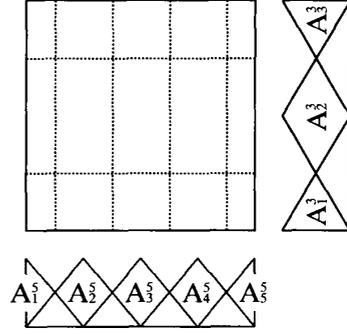


Fig. 13. Fuzzy partition by a 5×3 fuzzy grid.

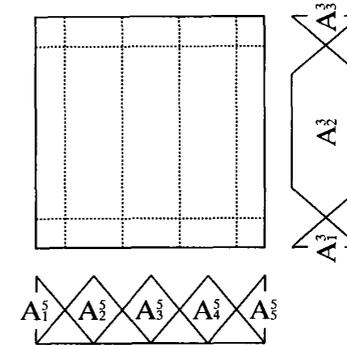


Fig. 14. Fuzzy partition with triangular and trapezoid membership functions.

In this paper, we generated candidate fuzzy if-then rules in S_{ALL} by using the same fuzzy subsets for each axis of a pattern space as shown in Figs. 1 and 2. Since our approach can handle any membership function and any fuzzy partition, this restriction of S_{ALL} can be relaxed as mentioned above. But the relaxation of this restriction may lead to a huge rule set S_{ALL} and long computation time. Therefore a method for effectively restricting the rule set S_{ALL} within tractable size will be necessary. The extension of the rule set S_{ALL} and its restriction are left for future work. The adjustment of membership functions should be also addressed together with the above-mentioned issues in future (see [24]).

VI. CONCLUDING REMARKS

In this paper, we proposed a genetic-algorithm-based method for selecting significant fuzzy if-then rules to construct a compact fuzzy classification system with high classification power. By computer simulations on the iris data, we showed that the number of fuzzy if-then rules was reduced to less than 2% of the generated 692 rules. The biased mutation probabilities played an important role in reducing the number of fuzzy if-then rules. The classification power of selected fuzzy if-then rules was also examined. It was shown by computer simulations on the iris data that the performance of selected fuzzy if-then rules was very high for training data. That is, the classification rates for training data were above 99% in almost all the parameter specifications of genetic algorithms. The performance of the proposed method for

test data was also high when the number of generations in genetic algorithms was small. The best result obtained by the proposed method outperformed all the results reported in [22] for nine fuzzy classification methods. Since the proposed method selects significant fuzzy if-then rules, it can be viewed as a knowledge acquisition tool for classification problems. Knowledge about a particular classification problem may be found out by carefully examining selected fuzzy if-then rules.

REFERENCES

- [1] M. Sugeno, "An introductory survey of fuzzy control," *Inform. Sci.*, vol. 36, pp. 59-83, 1985.
- [2] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller—Part I and Part II," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 2, pp. 404-435, 1990.
- [3] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. 15, no. 1, pp. 116-132, 1985.
- [4] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 6, pp. 1414-1427, 1992.
- [5] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Trans. Fuzzy Systems*, vol. 1, no. 1, pp. 7-31, 1993.
- [6] H. Ichihashi and T. Watanabe, "Learning control by fuzzy models using a simplified fuzzy reasoning," *J. Japan Soc. Fuzzy Theory Syst.*, vol. 2, no. 3, pp. 429-437, 1990 (in Japanese).
- [7] C. T. Lin and C. S. G. Lee, "Neural-network-based fuzzy logic control and decision system," *IEEE Trans. Comput.*, vol. 40, no. 12, pp. 1320-1336, 1991.
- [8] J. S. R. Jang, "Self-learning fuzzy controllers based on temporal back propagation," *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 714-723, 1992.
- [9] H. R. Berenji and P. Khedkar, "Learning and tuning fuzzy logic controllers through reinforcements," *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 724-740, 1992.
- [10] S. Horikawa, T. Furuhashi and Y. Uchikawa, "On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm," *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 801-806, 1992.
- [11] R. S. Michalski and R. L. Chilauskay, "Knowledge acquisition by encoding expert rules versus computer induction from examples: A case study involving soybean pathology," in *Fuzzy Reasoning and its Applications* E. H. Mamdani and B. R. Gaines, Eds. London: Academic, 1981, pp. 247-271.
- [12] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. Michigan Press, 1975.
- [13] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
- [14] C. L. Karr, "Design of an adaptive fuzzy logic controller using a genetic algorithm," in *Proc. Fourth Int. Conf. Genetic Algorithms*, San Diego, July 13-16, 1991, pp. 450-457.
- [15] C. L. Karr and E. J. Gentry, "Fuzzy control of pH using genetic algorithms," *IEEE Trans. Fuzzy Systems*, vol. 1, no. 1 pp. 46-53, 1993.
- [16] H. Nomura, I. Hayashi, and N. Wakami, "A self-tuning method of fuzzy reasoning by genetic algorithm," in *Proc. 1992 Int. Fuzzy Syst. Int. Contr. Conf.*, Louisville, Mar. 16-18, 1992, pp. 236-245.
- [17] P. Thrift, "Fuzzy logic synthesis with genetic algorithms," in *Proc. Fourth Int. Conf. Genetic Algorithms*. San Diego, July 13-16, 1991, pp. 509-513.
- [18] M. Valenzuela-Rendon, "The fuzzy classifier system: A classifier system for continuously varying variables," in *Proc. Fourth Int. Conf. Genetic Algorithms*. San Diego, July 13-16, 1991, pp. 346-353.
- [19] H. Ishibuchi, K. Nozaki, and H. Tanaka, "Distributed representation of fuzzy rules and its application to pattern classification," *Fuzzy Sets and Syst.*, vol. 52, pp. 21-32, 1992.
- [20] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals Eugenics*, vol. 7, pp. 179-188, 1936.
- [21] S. M. Weiss and C. A. Kulikowski, *Computer Systems that Learn*. San Mateo, CA: Morgan Kaufman, 1991.
- [22] M. Grabisch and F. Dispot, "A comparison of some methods of fuzzy classification on real data," in *Proc. of IIZUKA '92*, Iizuka, Japan, July 17-22, 1992, pp. 659-662.
- [23] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Construction of fuzzy classification systems with rectangular fuzzy rules using genetic algorithms," *Fuzzy Sets and Syst.*, vol. 65, pp. 237-253, 1994.
- [24] T. Murata and H. Ishibuchi, "Adjusting membership functions of fuzzy classification rules by genetic algorithms," in *Proc. of FUZZ-IEEE/IFES '95* Yokohama, Japan, Mar. 20-24, 1995, pp. 1819-1824.



Hisao Ishibuchi (M'93) received the B.S. and M.S. degrees in precision mechanics from Kyoto University in 1985 and 1987, respectively. In 1992, he received the Ph.D. degree from University of Osaka Prefecture.

He is currently an Associate Professor in the Department of Industrial Engineering at University of Osaka Prefecture. His current research interests include fuzzy data analysis, rule-based fuzzy systems, fuzzy neural networks, and genetic algorithms.

Dr. Ishibuchi is a member of INNS and IFSA.



Ken Nozaki received the B.S. and M.S. degrees from the Department of Industrial Engineering, University of Osaka Prefecture in 1991 and 1993, respectively. He is currently a Ph.D. candidate.

From Sept. 1993 to Mar. 1994, he was a Visiting Research Associate in University of Toronto. His research interests include fuzzy-rule-based classification systems, neural networks and genetic algorithms.



Naohisa Yamamoto received the B.S. and M.S. degrees from the Department of Industrial Engineering, University of Osaka Prefecture in 1991 and 1993, respectively.

His current research interests include scheduling and fuzzy modeling based on genetic algorithms.



Hideo Tanaka (M'79) received the B.S. degree from the Faculty of Engineering, Kobe University in 1962. He received the M.S. and Ph.D. degrees from the Faculty of Engineering, Osaka City University in 1966 and 1969, respectively.

He spent two years with the Central Research Laboratory of Daikin Corporation. He is now with the Department of Industrial Engineering, University of Osaka Prefecture. He has been a Professor since 1987. From 1972 to 1973, he was a Visiting Research Associate at UC, Berkeley; from 1975 to 1977, an Alexander Von Humboldt Fellow in Technical University of Aachen; and from 1981 to 1982, a Research Associate at Kansas State University. His research interests include possibility data analysis such as regression analysis and discriminant analysis based on possibility theory.