Fuzzy Rule Selection by Data Mining Criteria and Genetic Algorithms

Hisao Ishibuchi
Dept. of Industrial Engineering
Osaka Prefecture University
1-1 Gakuen-cho, Sakai, Osaka 599-8531, JAPAN
E-mail: hisaoi@ie.osakafu-u.ac.jp
Phone: +81-72-254-9350

Takashi Yamamoto
Dept. of Industrial Engineering
Osaka Prefecture University
1-1 Gakuen-cho, Sakai, Osaka, 599-8531, JAPAN
E-mail: yama@ie.osakafu-u.ac.jp
Phone: +81-72-254-9351

Abstract

This paper shows how a small number of fuzzy rules can be selected for designing interpretable fuzzy rule-based classification systems. Our approach consists of two phases: candidate rule generation by data mining criteria and rule selection by genetic algorithms. First a large number of candidate rules are generated and prescreened using two rule evaluation criteria in data mining. Next a small number of fuzzy rules are selected from candidate rules using genetic algorithms. Rule selection is formulated as an optimization problem with three objectives: to maximize the classification accuracy, to minimize the number of selected rules, and to minimize the total rule length. Thus the task of genetic algorithms is to find non-dominated rule sets with respect to the three objectives.

1. INTRODUCTION

Fuzzy rule-based systems have been successfully applied to various fields such as control, modeling, and classification (Leondes 1999). While the main goal in the design of fuzzy rule-based systems has been the performance maximization, their interpretability has also been taken into account in some recent studies (Pene-Reyes & Sipper 1999, Castillo et al. 2001, Roubos & Setnes 2001, and Casillas et al. 2002). In this paper, we consider three objectives in the design of fuzzy rule-based classification systems as in Ishibuchi, Nakashima & Murata (2001): Classification accuracy, the number of fuzzy rules, and the total length of fuzzy rules. The length of a fuzzy rule is the number of its antecedent conditions (i.e., the number of attributes in its antecedent part). The first objective is the performance maximization while the others are related to the interpretability. Usually human users do not want to manually check hundreds of fuzzy rules. Thus the number of fuzzy rules is closely related to the interpretability of fuzzy rule-based systems. Fuzzy rule-based systems with a small number of fuzzy rules are not always interpretable. Human users cannot intuitively understand long fuzzy rules with many antecedent conditions. Thus the rule length is also closely related to the interpretability of fuzzy rule-based systems. In this paper, we maximize the classification accuracy of fuzzy rule-based systems, minimize the number of fuzzy rules, and minimize the total length of fuzzy rules. Multi-objective genetic algorithms are used for finding non-dominated rule sets with respect to these three objectives.

Fuzzy rule generation methods can be categorized into two approaches according to their strategies for dividing the input space into fuzzy subspaces. One approach is based on grid-type fuzzy partitions where the domain interval of each input is divided into antecedent fuzzy sets with linguistic labels. Fig. 1 is an example of such a grid-type fuzzy partition. The other approach uses multi-dimensional antecedent fuzzy sets defined on the input space. Fig. 2 illustrates two-dimensional ellipsoidal antecedent fuzzy sets. Multi-dimensional antecedent fuzzy sets usually lead to fuzzy rule-based systems with high accuracy but low interpretability. On the other hand, fuzzy rule-based systems with high interpretability can be generated from grid-type fuzzy partitions. Since our goal is to generate interpretable fuzzy rule-based systems, we use the first approach (i.e., grid-type fuzzy partitions). As discussed in Suzuki & Furuhashi (2001), homogeneous fuzzy partitions are more interpretable than adjusted ones. Thus we use homogeneous fuzzy partitions as shown in Fig. 1. Usually we do not know an appropriate fuzzy partition for each input. In general, each input may have a different fuzzy partition while the two axes of the input space is divided by the same fuzzy partition in Fig. 1. Moreover, general rules may use coarse fuzzy partitions while specific rules may use fine fuzzy partitions in a single fuzzy rule-based system. For handling such a situation with different fuzzy partitions of different granularities, we specify each antecedent condition of fuzzy rules by choosing an antecedent fuzzy set from various fuzzy partitions for each input. In this paper, we use four fuzzy partitions in Fig. 3 where the total number of antecedent fuzzy sets is 14. For generating short fuzzy rules with a small number of antecedent conditions, we use "*don't care*" as an additional antecedent fuzzy set. Thus an antecedent fuzzy set for each input is chosen from the 14 fuzzy sets in Fig. 3 and "*don't care*". The total number of combinations of antecedent fuzzy sets is 15^n for an *n*-dimensional pattern classification problem.

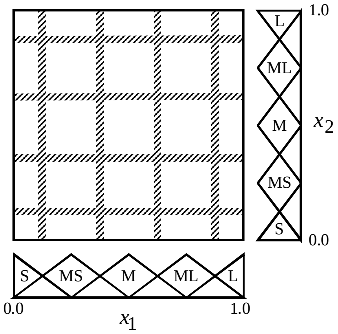

Figure 1: A 5 × 5 fuzzy grid of a two-dimensional input space.

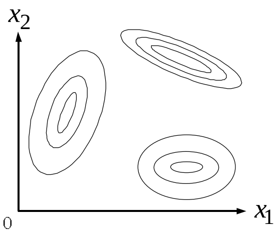

Figure 2: Ellipsoidal antecedent fuzzy sets.

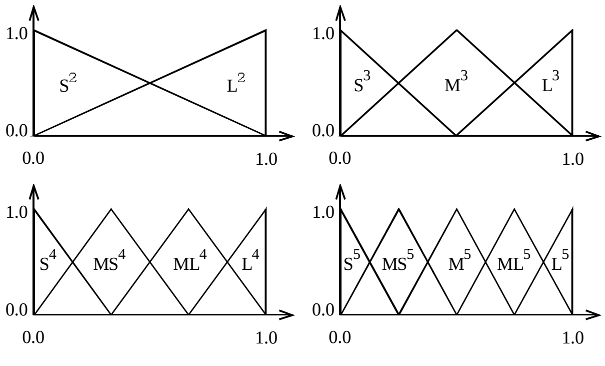

Figure 3: Four fuzzy partitions. The meaning of each label is as follows: S: *small*, MS: *medium small*, M: *medium*, ML: *medium large*, and L: *large*. The superscript of each label denotes the granularity of the corresponding fuzzy partition.

Genetic algorithm-based fuzzy rule selection (Ishibuchi, Nakashima & Murata, 2001) consists of two phases. In the first phase, a large number of candidate rules are generated from various combinations of antecedent fuzzy sets. In the second phase, subsets of the generated candidate rules are examined using genetic algorithms for finding non-dominated rule sets with respect to the above-mentioned three objectives. In Ishibuchi, Nakashima & Murata (2001), a single fuzzy partition was used for all inputs as in Fig. 1. In this case, the total number of combinations of antecedent fuzzy sets including "*don't care*" is $(5+1)^n$ for an *n*-dimensional pattern classification problem. This is much smaller than 15^n in this paper. That is, we have much more candidate rules. It should be noted that the search space for finding non-dominated rule sets exponentially expands as the number of candidate rules increases. The efficiency of genetic algorithms is significantly deteriorated by the increase in the number of candidate rules as shown in this paper. Thus we need a trick for decreasing the number of candidate rules. Our idea is to prescreen candidate rules based on fuzzy versions of two rule evaluation criteria (i.e., *confidence* and *support*) for association rules, which have been frequently used in the field of data mining (Agrawal et al. 1996). In our prescreening procedure, fuzzy rules are divided into several groups according to their consequent classes. Then fuzzy rules in each group are sorted in a descending order of the product of *confidence* and *support*. Finally a pre-specified number of fuzzy rules are chosen from the top of the rule list for each group. The selected fuzzy rules are used as candidate rules in our genetic algorithm-based rule selection method.

In the next section, we show how the design of fuzzy rule-based classification systems can be formulated as a three-objective rule selection problem. In Section 3, we propose a prescreening procedure of candidate rules using fuzzy versions of the two rule evaluation criteria in data mining. In Section 4, we describe a three-objective genetic algorithm for rule selection. The effect of the proposed prescreening procedure on the efficiency of the genetic algorithm-based rule selection method is examined in Section 5 through computer simulations. Finally Section 6 concludes this paper.

2. PROBLEM FORMULATION

Let us consider an *M*-class pattern classification problem with *m* labeled patterns $\boldsymbol{x}_p = (x_{p1}, \ldots, x_{pn})$, $p = 1$, 2,...,*m* in an *n*-dimensional continuous pattern space. For simplicity of explanation, we assume that the pattern space is the *n*-dimensional unit hypercube $[0, 1]^n$. That is, we assume that all attribute values are real numbers in the unit interval $[0, 1]$. For our pattern classification problem, we use fuzzy rules of the following form:

$$\text{Rule } R_q : \text{If } x_1 \text{ is } A_{q1} \text{ and ... and } x_n \text{ is } A_{qn} \\ \text{then Class } C_q \text{ with } CF_q, \qquad (1)$$

where R_q is the q-th fuzzy rule, $\mathbf{x} = (x_1, ..., x_n)$ is an n-dimensional pattern vector, A_{qi} is an antecedent fuzzy set, C_q is a consequent class (i.e., one of the M classes), and CF_q is a rule weight (i.e., certainty factor). The antecedent fuzzy set A_{qi} is one of the 14 fuzzy sets in Fig. 3 or "*don't care*". The rule weight CF_q is a real number in the unit interval $[0, 1]$. As shown in the next section, the consequent class C_q and the rule weight CF_q are determined in a heuristic manner from compatible training patterns with the antecedent part of R_q.

Let S be a subset of 15^n fuzzy rules of the form (1). Our task is to find rule sets with high classification ability and high interpretability. This task can be rephrased as finding a small number of simple fuzzy rules with high classification ability. As in Ishibuchi, Nakashima & Murata (2001), our rule selection problem is formulated as the following three-objective optimization problem:

$$\text{Maximize } f_1(S)\text{, and minimize } f_2(S),\ f_3(S), \quad (2)$$

where $f_1(S)$ is the number of correctly classified training patterns by S, $f_2(S)$ is the number of fuzzy rules in S, and $f_3(S)$ is the total rule length of fuzzy rules in S.

Usually there is no optimal rule set with respect to all the three objectives. Thus our task is to find multiple rule sets that are not dominated by any other rule sets. A rule set S_B is said to dominate another rule set S_A (i.e., S_B is better than S_A : $S_A \prec S_B$) if all the following inequalities hold:

$$f_1(S_A) \le f_1(S_B), \quad (3)$$
$$f_2(S_A) \ge f_2(S_B), \quad (4)$$
$$f_3(S_A) \ge f_3(S_B), \quad (5)$$

and at least one of the following inequalities holds:

$$f_1(S_A) < f_1(S_B), \quad (6)$$
$$f_2(S_A) > f_2(S_B), \quad (7)$$
$$f_3(S_A) > f_3(S_B). \quad (8)$$

The first condition (i.e., all the three inequalities in (3)-(5)) means that no objective of S_B is worse than S_A (i.e., S_B is not worse than S_A). The second condition (i.e., one of the three inequalities in (6)-(8)) means that at least one objective of S_B is better than S_A. When a rule set S is not dominated by any other rule sets, S is said to be a Pareto-optimal solution of our rule selection problem in (2). In many cases, it is impractical to try to find true Pareto-optimal solutions of our rule selection problem whose search space is huge (i.e., the search space is the power set of 15^n fuzzy rules). Thus we try to find near Pareto-optimal solutions. More specifically, first we decrease the search space by prescreening candidate fuzzy rules. Then we search for near Pareto-optimal solutions by a three-objective genetic algorithm.

3. CANDIDATE RULE PRESCREENING

3.1 FUZZIFICATION OF ASSOCIATION RULES

As we have already explained, the total number of combinations of antecedent fuzzy sets is 15^n for our n-dimensional pattern classification problem. When n is small (e.g., $n \le 4$), we can examine all combinations of antecedent fuzzy sets for generating fuzzy rules and use all the generated fuzzy rules as candidate rules in our genetic algorithm-based rule selection method. That is, no prescreening of candidate rules is necessary. On the other hand, we need a prescreening procedure when n is large. It is time-consuming to examine all the 15^n combinations when n is large (e.g., $n = 13$ in wine data used in computer simulations of this paper). In this case, it is also impractical to use all the generated fuzzy rules as candidate rules in our genetic algorithm-based rule selection method. Our idea is to use rule evaluation criteria in data mining for decreasing the number of candidate rules.

In the area of data mining, two criteria called *confidence* and *support* have often been used for evaluating association rules (Agrawal et al. 1996). Our fuzzy rule in (1) can be viewed as an association rule of the form $\mathbf{A}_q \Rightarrow C_q$. We use the two criteria for prescreening candidate rules. In this subsection, we show how the definitions of these two criteria can be extended to the case of the fuzzy association rule $\mathbf{A}_q \Rightarrow C_q$ (Ishibuchi, Yamamoto & Nakashima, 2001). Similar extensions of the two criteria to fuzzy association rules were also proposed in Hong et al. (2001).

Let D be the set of the given m training patterns $\mathbf{x}_p = (x_{p1}, ..., x_{pn})$, $p = 1, 2, ..., m$. The cardinality of D is m (i.e., $|D| = m$). The confidence of $\mathbf{A}_q \Rightarrow C_q$ is defined as follows (Agrawal et al. 1996):

$$c(\mathbf{A}_q \Rightarrow C_q) = \frac{|D(\mathbf{A}_q) \cap D(C_q)|}{|D(\mathbf{A}_q)|}, \quad (9)$$

where the denominator $|D(\mathbf{A}_q)|$ is the number of training patterns compatible with the antecedent part $\mathbf{A}_q$, and the numerator $|D(\mathbf{A}_q) \cap D(C_q)|$ is the number of training patterns compatible with both the antecedent part $\mathbf{A}_q$ and the consequent class C_q. The confidence c indicates the grade of the validity of $\mathbf{A}_q \Rightarrow C_q$. That is, c ($\times$ 100%) of training patterns compatible with $\mathbf{A}_q$ are also compatible with C_q. In the case of standard association rules, neither $\mathbf{A}_q$ nor C_q is fuzzy. Thus the calculations of $|D(\mathbf{A}_q)|$ and $|D(\mathbf{A}_q) \cap D(C_q)|$ can be performed by simply counting compatible training patterns. On the other hand, each training pattern has a different compatibility grade $\mu_{\mathbf{A}_q}(\mathbf{x}_p)$ with the antecedent part $\mathbf{A}_q$ when $\mathbf{A}_q \Rightarrow C_q$ is a fuzzy association rule. Thus such a compatibility grade should be taken into account. Since the consequent class C_q is not fuzzy, the confidence in (9) can be rewritten as follows (Ishibuchi, Yamamoto & Nakashima 2001):

$$c(\boldsymbol{A}_q \Rightarrow C_q) = \frac{|D(\boldsymbol{A}_q) \cap D(C_q)|}{|D(\boldsymbol{A}_q)|} = \frac{\sum_{p \in \text{Class } C_q} \mu_{\boldsymbol{A}_q}(\boldsymbol{x}_p)}{\sum_{p=1}^{m} \mu_{\boldsymbol{A}_q}(\boldsymbol{x}_p)} . \quad (10)$$

The compatibility grade $\mu_{\boldsymbol{A}_q}(\boldsymbol{x}_p)$ is usually defined by the product or minimum operator. In this paper, we use the product operator as

$$\mu_{\boldsymbol{A}_q}(\boldsymbol{x}_p) = \mu_{A_{q1}}(x_{p1}) \times \cdots \times \mu_{A_{qn}}(x_{pn}), \quad (11)$$

where $\mu_{A_{qi}}(x_{pi})$ is the membership function of the antecedent fuzzy set A_{qi} (i.e., each triangle in Fig. 3).

On the other hand, the support of $\boldsymbol{A}_q \Rightarrow C_q$ is defined as follows (Agrawal et al. 1996):

$$s(\boldsymbol{A}_q \Rightarrow C_q) = \frac{|D(\boldsymbol{A}_q) \cap D(C_q)|}{|D|} . \quad (12)$$

The support s indicates the grade of the coverage by $\boldsymbol{A}_q \Rightarrow C_q$. That is, s (× 100%) of all the training patterns are compatible with the association rule $\boldsymbol{A}_q \Rightarrow C_q$ (i.e., compatible with both $\boldsymbol{A}_q$ and C_q). In the same manner as the confidence in (10), the support in (12) can be rewritten as follows (Ishibuchi, Yamamoto & Nakashima 2001):

$$s(\boldsymbol{A}_q \Rightarrow C_q) = \frac{|D(\boldsymbol{A}_q) \cap D(C_q)|}{|D|} = \frac{\sum_{p \in \text{Class } C_q} \mu_{\boldsymbol{A}_q}(\boldsymbol{x}_p)}{m} . \quad (13)$$

3.2 CONSEQUENT CLASS AND RULE WEIGHT

The consequent class C_q of the fuzzy rule R_q with the antecedent part $\boldsymbol{A}_q$ is determined as

$$c(\boldsymbol{A}_q \Rightarrow C_q) = \max\{c(\boldsymbol{A}_q \Rightarrow \text{Class } 1), ..., c(\boldsymbol{A}_q \Rightarrow \text{Class } M)\}. \quad (14)$$

That is, the consequent class has the maximum confidence among the M alternative classes. It should be noted that the same class C_q is obtained for $\boldsymbol{A}_q$ when we use the support s instead of the confidence c. This is because the following relation holds between the confidence c and the support s from their definitions:

$$s(\boldsymbol{A}_q \Rightarrow \text{Class } h) = c(\boldsymbol{A}_q \Rightarrow \text{Class } h) \times \frac{|D(\boldsymbol{A}_q)|}{|D|}, \quad t = 1, 2, ..., M . \quad (15)$$

Since the second term (i.e., $|D(\boldsymbol{A}_q)| / |D|$) of the right-hand side is independent of the consequent class, the class with the maximum confidence is the same as the class with the maximum support. The same class also has the maximum product of these two criteria. Usually we can uniquely specify the consequent class C_q for each combination $\boldsymbol{A}_q$ of antecedent fuzzy sets. Only when multiple classes have the same maximum confidence (including the case of no compatible training pattern with the antecedent part $\boldsymbol{A}_q$: $c(\boldsymbol{A}_q \Rightarrow \text{Class } h) = 0$ for all classes), we cannot specify the consequent class C_q for $\boldsymbol{A}_q$. In this case, we do not generate the corresponding fuzzy rule R_q.

The confidence of R_q can be directly used as its rule weight as in Cordon et al. (1999). Our preliminary simulation results showed that better results were obtained from the following definition of the rule weight than the direct use of the confidence:

$$CF_q = c(\boldsymbol{A}_q \Rightarrow C_q) - c_{\text{Second}}, \quad (16)$$

where c_{Second} is the second largest confidence for the antecedent part $\boldsymbol{A}_q$:

$$c_{\text{Second}} = \max_h \{c(\boldsymbol{A}_q \Rightarrow \text{Class } h) \mid h \neq C_q\}. \quad (17)$$

Our preliminary computer simulations also showed that better results were obtained from the definition in (16) than the following definition used in some studies on fuzzy rule-based classification systems (e.g., Ishibuchi, Yamamoto & Nakashima 2001):

$$CF_q = c(\boldsymbol{A}_q \Rightarrow C_q) - c_{\text{Average}}, \quad (18)$$

where c_{Average} is the average confidence over fuzzy rules with the same antecedent part $\boldsymbol{A}_q$ but different consequent classes:

$$c_{\text{Average}} = \frac{1}{M-1} \sum_{h \neq C_q} c(\boldsymbol{A}_q \Rightarrow \text{Class } h). \quad (19)$$

3.3 PRESCREENING PROCEDURE

The generated fuzzy rules are divided into M groups according to their consequent classes. Fuzzy rules in each group are sorted in a descending order of the product of the confidence and the support (i.e., $s \cdot c$). For selecting N candidate rules, the first N/M rules are chosen from each of the M groups. In this manner, we can choose a pre-specified number of candidate rules as candidate rules in our genetic algorithm-based rule selection method. In our preliminary computer simulations, we also examined the confidence and the support as rule prescreening criteria. The best result among the three criteria for rule prescreening (i.e., confidence, support, and their product) was obtained when we used the product of the confidence and the support.

As we have already mentioned, the total number of combinations of antecedent fuzzy sets is 15^n for our *n*-dimensional pattern classification problem. Thus it is impractical to examine all combinations when *n* is large. In this case, we examine only short fuzzy rules with only a few antecedent conditions (i.e., with many *don't care* conditions). The number of fuzzy rules of the length *L* is calculated as ${}_nC_L \times 14^L$ because we have 14 antecedent fuzzy sets for each input (excluding *don't care*). Even when *n* is large, ${}_nC_L \times 14^L$ is not so large for a small value of *L*. This means that the number of short fuzzy rules is not so large even when the total number of fuzzy rules is huge.

4. GENETIC ALGORITHM

Many genetic algorithms for multi-objective optimization problems have been proposed in the literature (Zitzler & Thiele 1999, and Zitzler et al. 2000). Since each rule set can be represented by a binary string, we can apply those algorithms to our three-objective rule selection problem in Section 2. In this paper, we use a slightly modified version of a three-objective genetic algorithm for rule selection in Ishibuchi, Nakashima & Murata (2001). This algorithm has two characteristic features. One is to use a scalar fitness function with variable random weights for evaluating each string (i.e., each rule set). Whenever a pair of parent solutions is selected for crossover, weights are randomly updated. That is, each selection is governed by a different weight vector. Genetic search in various directions in the three-dimensional objective space is realized by this random weighting scheme. The other characteristic feature is to store all non-dominated solutions as a secondary population separately from a current population. The secondary population is updated at every generation. A small number of non-dominated solutions are randomly chosen from the secondary population and their copies are added to the current population as elite solutions. The convergence speed of the current population to Pareto-optimal solutions is improved by the elitist strategy. Other parts of our three-objective genetic algorithm are the same as standard single-objective genetic algorithms. Note that our task is to find multiple non-dominated solutions while the task of standard genetic algorithms is to find a single optimal solution. Of course, we can use other multi-objective genetic algorithms proposed in the literature.

An arbitrary subset S of *N* candidate fuzzy rules can be represented by a binary string of the length *N* as

$$S = s_1 s_2 \cdots s_N , \tag{20}$$

where $s_q = 0$ means that the *q*-th rule R_q is not included in S while $s_q = 1$ means that R_q is included in S. An initial population is constructed by randomly generating a pre-specified number of binary strings of the length *N*.

The first objective $f_1(S)$ of each string S is calculated by classifying all the given training patterns by *S*. We use a fuzzy reasoning method based on a single winner rule as in Ishibuchi, Nakashima & Murata (2001). In this fuzzy reasoning method, the classification of each pattern by the rule set *S* is performed by finding a single winner rule with the maximum product of the rule weight and the compatibility grade with that pattern. There are many cases where some fuzzy rules in *S* are not chosen as winner rules for any patterns. We can remove those fuzzy rules from *S* without degrading the classification accuracy of *S*. At the same time, the second and third objectives are improved by removing unnecessary fuzzy rules. Thus we remove all fuzzy rules that are not selected as winner rules of any patterns from the rule set *S*. The removal of those rules is performed for each string in the current population by changing the corresponding 1's to 0's before the second and third objectives are calculated.

After the three objectives of each string (i.e., each rule set) in the current population are calculated, the secondary population of non-dominated rule sets is updated. That is, each rule set in the current population is examined whether it is dominated by other rule sets in the current and secondary populations. If it is not dominated by any other rule sets, its copy is added to the secondary population. Then all rule sets dominated by the newly added one are removed from the secondary population. In this manner, the secondary population is updated at every generation.

The fitness value of each rule set S in the current population is defined by the three objectives as

$$fitness(S) = w_1 \cdot f_1(S) - w_2 \cdot f_2(S) - w_3 \cdot f_3(S) , \tag{21}$$

where w_1, w_2 and w_3 are weights satisfying the following conditions:

$$w_1, w_2, w_3 \geq 0 , \tag{22}$$

$$w_1 + w_2 + w_3 = 1. \tag{23}$$

Whenever a pair of parent strings is selected from the current population, these weights are randomly updated. The random specification of the rule weights is to search for a variety of non-dominated rule sets in the three-dimensional objective space. Binary tournament selection with replacement is used for selecting a pair of parent strings using the scalar fitness function in (21) with the randomly specified weights. That is, two strings are randomly selected from the current population and the better one is chosen as a parent string. Then the two strings are returned to the current population. The other parent string is also selected in the same manner using the same weight values. When another pair of parent strings is selected, the weight values are randomly updated.

Uniform crossover is applied to each pair of parent strings to generate a new string. Then biased mutation is applied to the generated string for efficiently decreasing the number of fuzzy rules included in the string. In the biased mutation operation, a larger probability is assigned to the mutation from 1 to 0 (i.e., mutation for decreasing the number of fuzzy rules) than the mutation from 0 to 1 (i.e., mutation for increasing the number of fuzzy rules).

The next population consists of the newly generated strings by the genetic operations. Some non-dominated strings in the secondary population are randomly selected as elite solutions and their copies are added to the new population. The outline of the three-objective genetic algorithm for rule selection is written as follows:

Step 0: Parameter Specification.

Specify the population size N_{pop}, the number of elite solutions N_{elite} that are randomly selected from the secondary population and added to the current population, the crossover probability p_c, two mutation probabilities $p_m(1 \rightarrow 0)$ and $p_m(0 \rightarrow 1)$, and the stopping condition.

Step 1: Initialization.

Randomly generate N_{pop} binary strings of the length N as an initial population. Calculate the three objectives of each string. In this calculation, unnecessary rules are removed from each string. Find non-dominated strings (i.e., non-dominated rule sets) in the initial population. A secondary population consists of copies of those non-dominated strings.

Step 2: Genetic Operations.

Generate $(N_{pop} - N_{elite})$ strings using genetic operations (i.e., binary tournament selection, uniform crossover, and biased mutation) from the current population.

Step 3: Evaluation.

Calculate the three objectives of each of the newly generated $(N_{pop} - N_{elite})$ strings. In this calculation, unnecessary rules are removed from each string. The current population consists of the modified strings.

Step 4: Secondary Population Update.

Update the secondary population by examining each string in the current population as mentioned above.

Step 5: Elitist Strategy.

Randomly select N_{elite} strings from the secondary population and add their copies to the current population.

Step 6: Termination Test.

If the stopping condition is not satisfied, return to Step 2. Otherwise terminate the execution of the algorithm. All the non-dominated strings among examined ones in the execution of the algorithm are stored in the secondary population.

5. COMPUTER SIMULATIONS

We apply the proposed rule selection method to wine data available from the UCI Machine Learning Repository (http://www.ics.uci.edu/~mlearn/MLSummary.html). The wine data set consists of 178 samples with 13 continuous attributes from three classes. We normalized each attribute value into a real number in the unit interval [0, 1]. Thus the wine data set was handled as a three-class pattern classification problem in the 13-dimensional unit hypercube $[0,1]^{13}$. The total number of possible combinations of antecedent fuzzy sets is 15^{13}.

First we generated fuzzy rules of the length three or less using all the 178 samples as training patterns. The number of generated fuzzy rules of each length is summarized in Table 1. The fuzzy rule of the length zero has no antecedent conditions, Class 2 consequent, and a very small certainty grade (i.e., rule weight). This fuzzy rule can be generated because the number of Class 2 samples is the largest among the three classes in the wine data.

Table 1: The number of generated fuzzy rules of each length.

Length of rules	0	1	2	3	Total
Number of rules	1	182	14,781	696,752	711,716

The generated 711,716 fuzzy rules were divided into three groups according to their consequent classes. Fuzzy rules in each class were sorted in a descending order of the product of the confidence and the support. From each group, the first 300 fuzzy rules were selected as candidate rules ($N = 900$: 900 candidate rules in total). Then the three-objective genetic algorithm was applied to the 900 candidate rules using the following parameter specifications.

Population size: $N_{pop} = 50$,
Number of elite solutions: $N_{elite} = 5$,
Crossover probability: $p_c = 0.9$,
Mutation probability: $p_m(1 \rightarrow 0) = 0.1$,
$p_m(0 \rightarrow 1) = 1/N$,
Stopping condition: 10,000 generations.

Our computer simulations were iterated 20 times. Non-dominated rule sets obtained from those 20 trials are summarized in Table 2. Examples of the obtained rule sets in Table 2 are shown in Fig. 4 and Fig. 5. Fig. 4 shows three fuzzy rules with only a single antecedent condition, which correspond to the second rule set with a 94.9% classification rate in Table 2. Fig. 5 shows three fuzzy rules with a few antecedent conditions, which correspond to the sixth rule set with a 100% classification rate in Table 2.

Table 2: Non-dominated rule sets obtained from 20 trials of the proposed method with 900 candidate rules.

Number of rules	Average rule length	Classification rate (%)
3	0.67	88.2
3	1.00	94.9
3	1.33	96.1
3	1.67	98.3
3	2.00	99.4
3	2.33	100.0
4	0.75	96.1
4	1.00	97.2
4	1.25	98.9

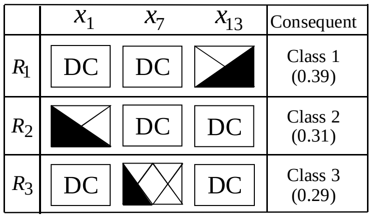

Figure 4: Three fuzzy rules with a 94.9% classification rate.

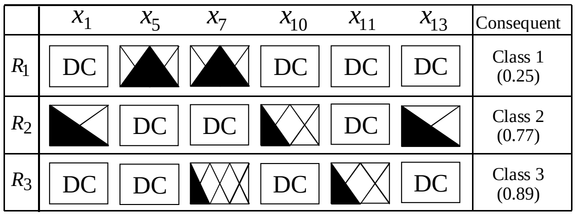

Figure 5: Three fuzzy rules with a 100% classification rate.

From Table 2, we can see that our rule selection method found various rule sets with different classification rates and different sizes. The selected rule sets have high interpretability as shown in Fig. 4 and Fig. 5. From the comparison between Fig. 4 and Fig. 5, we can observe the existence of a tradeoff between classification accuracy and interpretability (i.e., the three fuzzy rules in Fig. 5 have a higher classification rate but less interpretable).

For examining the usefulness of the proposed prescreening procedure of candidate rules, the same computer simulation was performed using randomly selected 900 candidate rules from the generated 711,716 fuzzy rules. Simulation results are summarized in Table 3. From the comparison between Table 2 and Table 3, we can see that the classification ability and/or the interpretability of obtained rule sets were deteriorated by the use of randomly selected candidate rules.

We also performed the same computer simulation using no prescreening procedure. In this case, all the generated 711,716 fuzzy rules were used as candidate rules. Thus the string length was 711,716. As we can expect, the execution of the three-objective genetic algorithm with such a long string required large memory storage and long CPU time. Table 4 shows non-dominated rule sets obtained from ten trials of the three-objective genetic algorithm. Since the search space was too large, good rule sets could not be obtained within a reasonable computation time (especially with respect to the number of fuzzy rules as shown in Table 4). The average CPU time for each trial was about 11 hours in Table 4 while it was about four minutes in Table 2 with 900 candidate rules selected by the proposed prescreening procedure.

Table 3: Simulation results with randomly selected 900 candidate rules.

Number of rules	Average rule length	Classification rate (%)
3	1.67	86.5
3	2.00	93.3
3	2.33	95.5
3	2.67	96.1
4	2.25	96.6
4	2.50	97.2
4	2.75	97.8
5	2.40	98.3
5	2.60	98.9
6	2.50	99.4
7	2.57	100.0
8	2.13	100.0

Table 4: Simulation results with 711,716 candidate rules.

Number of rules	Average rule length	Classification rate (%)
5	1.40	94.4
5	1.60	96.1
6	1.50	96.6
6	1.83	98.3
7	1.71	100.0

Finally we examined the effect of using various fuzzy partitions for each input on the classification performance of fuzzy rule-based classification systems. In the same manner as the computer simulation for Table 2, we applied our rule selection method to the wine data set using only the finest fuzzy partition with five linguistic labels in Fig. 3 (i.e., the bottom-right fuzzy partition in

Fig. 3). Table 5 shows non-dominated rule sets obtained from 20 trials. From the comparison between Table 2 and Table 5, we can see that smaller rule sets with higher classification rates were obtained in Table 2 than Table 5. This result was expected from the fact that the three fuzzy rules with a 100% classification rate in Fig. 5 use various fuzzy partitions with different granularities.

Table 5: Non-dominated rule sets obtained from 20 trials using only a single fuzzy partition with five fuzzy sets.

Number of rules	Average rule length	Classification rate (%)
3	0.67	85.4
3	1.00	91.6
3	1.33	93.3
4	1.00	95.5
4	1.25	96.1
4	1.50	97.2
5	1.00	97.2
5	1.40	97.8
5	1.60	98.3
5	1.80	98.9
6	1.00	97.8
6	1.17	98.3
6	1.33	98.9
6	1.50	99.4
7	1.57	100.0

6. CONCLUSIONS

In this paper, we extended the genetic algorithm-based rule selection method in Ishibuchi, Nakashima & Murata (2001) to the case where various fuzzy partitions with different granularities are used for each input. This extension leads to the increase in the number of candidate rules. Thus we proposed a prescreening procedure for decreasing the number of candidate rules. The proposed prescreening procedure is based on two rule evaluation criteria of association rules in the field of data mining. Through computer simulations, we demonstrated the necessity of candidate rule prescreening in genetic algorithm-based rule selection. The three-objective genetic algorithm could not find good rule sets when candidate rules were randomly chosen. In the case of no prescreening, the CPU time was very long (i.e., about 11 hours) while it was a few minutes in the case with the proposed prescreening procedure.

REFERENCES

R. Agrawal et al. (1996) "Fast discovery of association rules," in U. M. Fayyad et al. (eds.) *Advances in Knowledge Discovery & Data Mining*, 307-328, AAAI Press, Menlo Park.

J. Casillas, O. Cordón, F. Herrera, and L. Magdalena (2002) *Trade-off between Accuracy and Interpretability in Fuzzy Rule-Based Modeling*, Physica-Verlag.

L. Castillo, A. Gonzalez, and P. Perez (2001) "Including a simplicity criterion in the selection of the best rule in a genetic fuzzy learning algorithm," *Fuzzy Sets and Systems* **120**, 309-321.

O. Cordon, M. J. del Jesus, and F. Herrera (1999) "A proposal on reasoning methods in fuzzy rule-based classification systems," *International Journal of Approximate Reasoning* **20**, 21-45.

U. M. Fayyad and K. B. Irani (1993) "Multi-interval discretization of continuous-valued attributes for classification learning," *Proc. of 13th International Joint Conference on Artificial Intelligence*, 1022-1027.

T. -P. Hong, C. -S. Kuo, and S. -C. Chi (2001) "Trade-off between computation time and number of rules for fuzzy mining from quantitative data," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **9**, 587-604.

H. Ishibuchi, T. Nakashima, and T. Murata (2001) "Three-objective genetics-based machine learning for linguistic rule extraction," *Information Sciences* **136**, 109-133.

H. Ishibuchi, T. Yamamoto, and T. Nakashima (2001) "Fuzzy data mining: Effect of fuzzy discretization," *Proc. of 1st IEEE International Conference on Data Mining*, 241-248.

C. T. Leondes (1999) *Fuzzy Theory Systems: Techniques and Applications (Vols. 1-4)*, Academic Press, San Diego.

C. A. Pene-Reyes and M. Sipper (1999) "Designing brest cancer diagnostic systems via a hybrid fuzzy-genetic methodology," *Proc. of IEEE International Conference on Fuzzy Systems* **1**, 135-139.

J. R. Quinlan (1993) *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo.

H. Roubos and M Setnes (2001) "Compact and transparent fuzzy models and classifiers through iterative complexity reduction," *IEEE Trans. on Fuzzy Systems* **9**, 516-524.

T. Suzuki and T. Furuhashi (2001) "Evolutionary algorithm based fuzzy modeling using conciseness measure," *Proc. of Joint IFSA-NAFIPS International Conference*, 1575-1580.

E. Zitzler, K. Deb, and L. Thiele (2000) "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation* **8**, 173-195.

E. Zitzler and L. Thiele (1999) "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. on Evolutionary Computation* **3**, 257-271.