Electronics and Communications in Japan, Part 3, Vol. 80, No. 12, 1997
Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. J80-A, No. 2, February 1997, pp. 379–387

Comparison of the Michigan and Pittsburgh Approaches to the Design of Fuzzy Classification Systems

Hisao Ishibuchi, Tomoharu Nakashima, and Tadahiko Murata

College of Engineering, Osaka Prefecture University, Sakai, Osaka, Japan 593

SUMMARY

Fuzzy systems based on fuzzy if–then rules have been applied to various problems. The main application area has been fuzzy control problems. In many cases, such fuzzy systems can handle only a few input variables. This is because the number of fuzzy if–then rules exponentially increases as the number of input variables increases. In this paper, we try to design fuzzy classification systems based on fuzzy if–then rules for multidimensional pattern classification problems with many attributes. For designing such fuzzy classification systems, we compare two frameworks in the area of genetics-based machine learning: the Michigan approach and the Pittsburgh approach. The performance of fuzzy rule-based classification systems is also compared with that of various pattern classification methods. In computer simulations, we use a wine classification problem with 13 attributes, a cancer diagnosis problem with 9 attributes, and a credit approval problem with 14 attributes. © 1997 Scripta Technica, Inc. Electron Comm Jpn Pt 3, **80**(12): 10–19, 1997

Key words: Fuzzy classification system; rule selection; Michigan approach; Pittsburgh approach.

1. Introduction

Fuzzy systems based on fuzzy if–then rules have been applied to various problems. Those applications have been mainly in the field of control problems [1, 2]. Fuzzy if–then rules used for control problems have often been derived from human experts as linguistic knowledge. Automatic rule generation methods from numerical data have also been proposed. For example, Wang and Mendel [3] proposed a method for generating fuzzy if–then rules with highest compatibility grades with numerical data. Abe and Lan [4] proposed a method for generating hyperrectangle fuzzy if–then rules that covered numerical data. Some methods based on genetic algorithms [5, 6] were also proposed for generating fuzzy if–then rules and adjusting them. For example, Nomura and colleagues [7] and Ishigami and colleagues [8] adjusted the number of fuzzy if–then rules and the shape of the membership function of each antecedent fuzzy set. Valenzuela-Rendón [9] automatically generated fuzzy if–then rules using a fuzzy classifier system.

For pattern classification problems, a fuzzy if–then rule can be generated for each fuzzy subspace after fuzzily partitioning a pattern space into fuzzy subspaces with various sizes. Ishibuchi and colleagues [10, 11] proposed a pattern classification method based on fuzzy if–then rules generated from various fuzzy partitions. In their method, the number of fuzzy sets for each axis was the same in a fuzzy partition. While this method utilized the restriction that the number of fuzzy sets for each axis should be the same, the number of fuzzy if–then rules exponentially increased as the dimensionality of the pattern space increased.

A rule selection problem has been studied for obtaining a compact and high-performance classification system by selecting only significant fuzzy if–then rules from a large number of generated rules. For such a rule selection

CCC1042-0967/97/120010-10

problem of fuzzy if–then rules, Ishibuchi and colleagues [12–14] proposed a genetic-algorithm-based method and obtained a compact rule set that could correctly classify many training patterns. In their method, a rule set was coded as a string (i.e., as an individual) by assigning a bit to each candidate fuzzy if–then rule. Thus, the string length was the same as the number of candidate fuzzy if–then rules. This limited the use of their method to low-dimensional pattern classification problems (e.g., pattern classification problems with fewer than five attributes).

In this paper, we compare two approaches to the construction of fuzzy classification systems for multidimensional pattern classification problems with many attributes. One is the Pittsburgh approach, which handles a rule set of a certain number of fuzzy if–then rules as an individual. The other is the Michigan approach, where each fuzzy if–then rule is handled as an individual [15]. These two approaches are examined herein with respect to the classification ability for training data and the generalization ability for test data by performing computer simulations on multidimensional pattern classification problems: a wine classification problem with 13 attributes [16], a cancer diagnosis problem with 9 attributes [17], and a credit approval problem with 14 attributes [18].

2. Fuzzy If–Then Rules and Pattern Classification Problems

2.1. Generating fuzzy if–then rules

Let us assume that m training patterns $\mathbf{x}_p = (x_{p1}, \ldots, x_{pn})$, $p = 1, \ldots, m$, are given for an n-dimensional c-class pattern classification problem. We also assume that the n-dimensional pattern space is normalized as the unit hypercube $[0, 1]^n$. The following fuzzy if–then rules are employed for such a pattern classification problem:

$$\text{Rule } R_j : \text{If } x_{p1} \text{ is } A_{j1} \text{ and} \ldots \text{and } x_{pn} \text{ is } A_{jn} \text{ then Class } C_j \text{ with } CF = CF_j, \quad j = 1, 2, \ldots, r \tag{1}$$

where R_j is the label of the rule, $A_{j1}, \ldots, A_{jn}$ are fuzzy sets on the unit interval [0, 1], C_j is a consequent class, CF_j is the grade of certainty of the fuzzy if–then rule R_j, and r is the number of fuzzy if–then rules in the fuzzy classification system. The maximum value of r is the total number of possible fuzzy if–then rules (say, N). As the antecedent fuzzy sets $A_{j1}, \ldots, A_{jn}$, we can use any fuzzy sets with arbitrary shapes. For example, we can use trapezoidal fuzzy sets and bell-shape fuzzy sets as well as triangular fuzzy sets in Fig. 1. In this paper, we use the six fuzzy sets in Fig. 1 as the antecedent fuzzy sets of fuzzy if–then rules. In this case, the total number of fuzzy if–then rules N is $N = (1 + 5)^n = 6^n$. Each fuzzy set in Fig. 1 corresponds to a linguistic value, namely, S: small, MS: medium small, M: medium, ML: medium large, L: large, and DC: don't care.

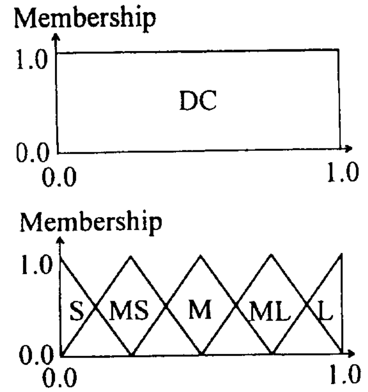


Fig. 1. Antecedent fuzzy sets.

The consequent class C_j and the grade of certainty CF_j in (1) can be determined by the following procedure [10–14]. First, the sum of the compatibility grades of training patterns with the fuzzy if–then rule R_j is calculated for each class. Then, the consequent class C_j of fuzzy if–then rule R_j is specified as the class with the maximum sum of the compatibility grades. If such a class cannot be determined uniquely (i.e., if multiple classes have the same maximum sum), the fuzzy if–then rule R_j becomes a dummy rule that has no influence on the classification. The grade of certainty CF_j of the fuzzy if–then rule R_j is determined from the sum of the compatibility grades of training patterns from each class (for detail, see Refs. 10 to 14).

2.2. Fuzzy reasoning

Let a rule set S be a set of fuzzy if-then rules generated by the above procedure. This rule set S can be arbitrarily specified as any subset (or the universal set) of generated fuzzy if–then rules. The classification of a pattern $\mathbf{x}_p$ is done by the following procedure, using fuzzy if–then rules in the rule set S (for details, see Refs. 10 to 14). First, we calculate the product of the compatibility grade of the pattern $\mathbf{x}_p$ with each fuzzy if–then rule and its grade of certainty. Then, pattern $\mathbf{x}_p$ is classified as the consequent class of the fuzzy if–then rule with the maximum product. If such a class cannot be determined uniquely (i.e., if multiple consequent classes involve the same maximum product), the classification of pattern $\mathbf{x}_p$ is rejected.

2.3. Learning of the grade of certainty

In Ref. 19, the following procedure was used for adjusting the grade of certainty of each fuzzy if–then rule used in the fuzzy classification system. First, a training pattern $\mathbf{x}_p$ is classified by the above fuzzy reasoning method using fuzzy if–then rules in rule set S. This training pattern $\mathbf{x}_p$ is classified by the fuzzy if–then rule R_{j^*} that has the maximum product of the compatibility grade with $\mathbf{x}_p$ and the grade of certainty. When $\mathbf{x}_p$ is correctly classified, the grade of certainty CF_{j^*} of that rule is increased as follows:

$$CF_{j^*}^{\text{new}} = CF_{j^*}^{\text{old}} + \eta_1 \cdot (1 - CF_{j^*}^{\text{old}}) \qquad (2)$$

where η_1 is a learning constant. On the other hand, when $\mathbf{x}_p$ is misclassified, the grade of certainty CF_{j^*} is decreased as follows:

$$CF_{j^*}^{\text{new}} = CF_{j^*}^{\text{old}} - \eta_2 \cdot CF_{j^*}^{\text{old}} \qquad (3)$$

where η_2 is a learning constant.

3. Rule Selection Based on Michigan Approach

3.1. Coding of individuals

In a rule selection method based on a fuzzy classifier system proposed by Ishibuchi and colleagues [15], each fuzzy if–then rule was handled as an individual. Thus, their method can be viewed as a rule selection method based on the Michigan approach. Fuzzy if–then rules are coded as strings by assigning the integers 0, 1, 2, 3, 4, and 5 to the antecedent fuzzy sets DC (don't care), S (small), MS (medium small), M (medium), ML (medium large), and L (large), respectively. For example, the following fuzzy if–then rule R_j is coded as $R_j = 5132$:

If x_{p1} is L and x_{p2} is S and x_{p3} is M and x_{p4} is MS then Class 2 with $CF = 0.6$.

The consequent class and the grade of certainty of each fuzzy if–then rule are not coded (i.e., not included in a string) because they can be determined when the antecedent fuzzy sets are specified. The consequent class and the grade of certainty are determined by the rule generation method in section 2.1.

3.2. Definition of fitness

In the Michigan approach, each rule corresponds to an individual, and a rule set S used in a fuzzy classification system corresponds to a population. For calculating the fitness of each fuzzy if–then rule, we first classify training patterns by means of all fuzzy if–then rules included in a current population. As mentioned in section 2.2, each training pattern is classified by the fuzzy if–then rule with the maximum product of the compatibility grade and the grade of certainty. Then we assign a fitness value to each fuzzy if–then rule as follows:

$$f(R_j) = w_{\text{NCP}} \cdot NCP(R_j) - w_{\text{NMP}} \cdot NMP(R_j) \qquad (4)$$

where $NCP(R_j)$ is the number of training patterns that are correctly classified by the fuzzy if–then rule R_j, $NMP(R_j)$ is the number of training patterns that are misclassified by R_j, and w_{NCP} and w_{NMP} are nonnegative weights for $NCP(R_j)$ and $NMP(R_j)$, respectively.

3.3. Crossover and mutation operations

In the previous subsection, a fitness value was assigned to each individual (i.e., each fuzzy if–then rule). Each individual is selected as a parent with the following probability:

$$P(R_j) = \frac{f(R_j) - f_{\min}(S)}{\sum_{R_i \in S} \{f(R_i) - f_{\min}(S)\}} \qquad (5)$$

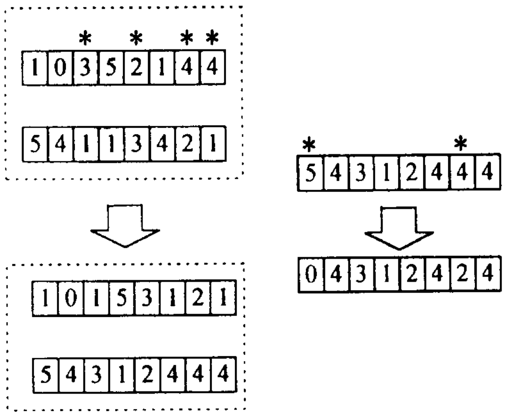
where $f_{\min}(S)$ is the minimum fitness value of the fuzzy if–then rules in population S.

Two fuzzy if–then rules are generated from a pair of selected fuzzy if–then rules by a uniform crossover operation as illustrated in Fig. 2(a). Then, a mutation operation illustrated in Fig. 2(b) is applied to the generated fuzzy if–then rules. The rule generation procedure in section 2. 1 is employed for determining the consequent class and the grade of certainty of each fuzzy if–then rule whose antecedent fuzzy sets have been specified by the crossover and mutation operations.

3.4. Algorithm

Our fuzzy classifier system can be expressed as the following algorithm:

(i) Randomly generate N_{rule} fuzzy if–then rules as an initial population. Each fuzzy if–then rule is generated by randomly selecting its antecedent fuzzy sets and determining its consequent class and its grade of certainty by the rule generation procedure in section 2.1. Let us denote the set of the generated fuzzy if–then rules as S.



(a) Crossover (b) Mutation

Fig. 2. Genetic operations.

(ii) Apply the learning algorithm of the grade of certainty in section 2.3 to the fuzzy if–then rules in rule set S.

(iii) Assign a fitness value to each fuzzy if–then rule in rule set S by (4).

(iv) Select fuzzy if–then rules as parents from the current population S according to the selection probability in (5). Generate two fuzzy if–then rules from each pair of selected fuzzy if–then rules by the crossover operation. Then apply the mutation operation to the generated fuzzy if–then rules.

(v) Replace poor fuzzy if–then rules in the current population (i.e., in rule set S) with the generated fuzzy if–then rules whose antecedent fuzzy sets have been determined in (iv). In this replacement procedure, the worst $P_{rep} \cdot N_{rule}$ fuzzy if–then rules with the smallest fitness values in the current population are replaced with the generated $P_{rep} \cdot N_{rule}$ fuzzy if–then rules where P_{rep} is the replacement rate (i.e., generation gap).

(vi) Return (ii) if a prespecified stopping condition of the algorithm is not satisfied.

4. Rule Selection Based on Pittsburgh Approach

In this section, for making the Pittsburgh approach applicable to high-dimensional pattern classification problems, we propose a rule selection method where the number of fuzzy if–then rules included in each rule set is fixed as a prespecified number. That is, each rule set with the prespecified number of fuzzy if–then rules is handled as an individual of a genetic algorithm in the proposed method.

4.1. Definition of fitness

We used the following fitness function in the proposed method:

$$f(S) = NCP(S) \qquad (6)$$

That is, the fitness value of rule set S is defined only in terms of the number of correctly classified training patterns $NCP(S)$.

4.2. Coding of individuals

In the same manner as in section 3, antecedent fuzzy sets of fuzzy if–then rules in a rule set are denoted by the integers 0 to 5. The consequent class and the grade of certainty of each fuzzy if–then rule are not coded (i.e., not included in a string) because they are determined by the rule generation method in section 2.1 when its antecedent fuzzy sets are specified. Let us denote the prespecified number of fuzzy if–then rules in each rule set by N_{rule}. Because the number of antecedent fuzzy sets of each fuzzy if–then rule is equal to the dimensionality n of the pattern space, the length of a string corresponding to a rule set is $n \cdot N_{rule}$. Thus, the string length does not exponentially increase as the dimensionality of the pattern space increases. The string length increases proportionally to the dimensionality of the pattern space.

4.3. Crossover and mutation operations

Each rule set S_i whose fitness is defined by (6) is selected as a parent for a crossover operation with the following probability:

$$P(S_i) = \frac{\{f(S_i) - f_{\min}(\Psi)\}}{\sum_{S_j \in \Psi} \{f(S_j) - f_{\min}(\Psi)\}} \qquad (7)$$

where $f_{\min}(\Psi)$ is the minimum fitness value of rule sets in the current population Ψ.

After selecting a pair of parents from the current population Ψ with the selection probability in (7), we apply a uniform crossover operation to the selected pair with a prespecified crossover probability P_c. There are two versions for defining crossover points in the implementation of the uniform crossover, both of which are examined in this paper. In one version, crossover points lie only between rules (see Fig. 3). In the other version, strings can be cut at any points (see Fig. 4). In the first version, a substring of n integers is handled as a block as shown in Fig. 3, where $n = 3$.

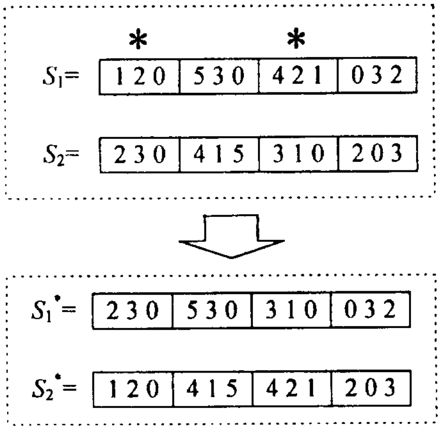


Fig. 3. Uniform crossover with crossover points between rules.

With a prespecified mutation probability P_m, a mutation operation is applied to new individuals generated by the crossover. In the mutation, antecedent fuzzy sets of fuzzy if–then rules included in each individual are replaced with randomly selected other fuzzy sets [see Fig. 2(b)].

4.4. Algorithm

Our rule selection method based on the Pittsburgh approach can be expressed as the following algorithm:

(i) Generate an initial population. That is, randomly generate N_{set} rule sets (i.e., N_{set} individuals), each of which consists of N_{rule} fuzzy if–then rules. In this procedure, each fuzzy if–then rule is generated in the same manner as in section 3. Let us denote each of the generated rule sets by S_i ($i = 1, 2, \ldots, N_{set}$). We also denote the current population by Ψ. That is, Ψ is the population of the generated rule sets.

(ii) Apply the learning procedure in section 2.3 to each rule set.

(iii) Assign the fitness value in (6) to each rule set.

(iv) Select each rule set as a parent for the crossover with the selection probability in (7). Apply the uniform crossover to each pair of selected rule sets. Apply the mutation operation to the antecedent fuzzy sets of fuzzy if–then rules included in each rule set generated by the crossover. Using the rule generation procedure in section 2.1, determine the consequent class and the grade of certainty of each fuzzy if–then rule whose antecedent fuzzy sets have been specified by the crossover and the mutation.

(v) Randomly select an individual from the newly generated population, and replace it with the elite individual in the previous population.

(vi) Return (ii) if a prespecified stopping condition of the algorithm is not satisfied.

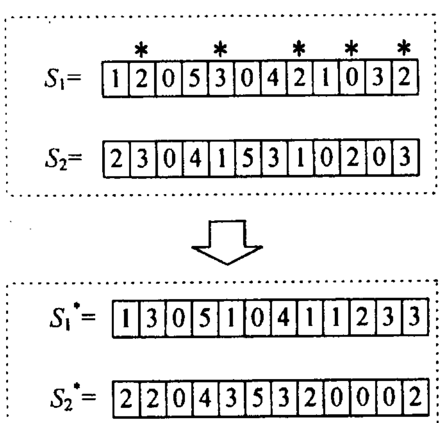


Fig. 4. Uniform crossover with arbitrary crossover points.

5. Similarities and Dissimilarities of the Two Approaches

The two approaches described in sections 3 and 4 have the following similarities.

(1) The crossover operations in the two approaches are basically the same when the uniform crossover in the Pittsburgh approach has no restriction on crossover points (i.e., when the uniform crossover in Fig. 4 is used).

(2) Both approaches use the same mutation operation.

(3) The same procedure is employed for generating initial rule sets.

On the other hand, the dissimilarities between the two approaches can be summarized as follows:

(1) In the Michigan approach, a fitness value is assigned to each fuzzy if–then rule. On the contrary, a fitness value is assigned to each rule set in the Pittsburgh approach.

(2) In the Michigan approach, the current population (i.e., a rule set) is updated by replacing fuzzy if–then rules that have low fitness values. That is, fuzzy if–then rules with high fitness values are inherited by the next population with no change. On the contrary, half of the fuzzy if–then rules in each rule set are replaced with those in another rule set on average in the Pittsburgh approach when each rule is handled as a block in the crossover (i.e., when the uniform crossover in Fig. 3 is used). Such replacement is done

randomly, regardless of the fitness value of each fuzzy if–then rule. When the uniform crossover has no restriction (i.e., when the uniform crossover in Fig. 4 is used), almost all of the fuzzy if–then rules in each rule set are replaced with new rules in the Pittsburgh approach.

(3) In the Michigan approach, the mutation is applied only to new fuzzy if–then rules generated for replacing current rules whose fitness values are low. The mutation is not applied to current rules with high fitness values, which are inherited by the next population with no change. On the contrary, the mutation is applied to all fuzzy if–then rules in the Pittsburgh approach, because each rule set is handled as an individual.

As we can see from these dissimilarities, the main characteristic feature of the Pittsburgh approach is that the performance of each rule set is directly utilized in the genetic algorithm. On the other hand, the main characteristic feature of the Michigan approach is that good fuzzy if–then rules are inherited by the next generation with no change. Good fuzzy if–then rules are also utilized for generating new rules. Such handling of good fuzzy if–then rules is realized by evaluating the performance of each fuzzy if–then rule in the Michigan approach.

We will now examine how these dissimilarities influence the performance of each approach by computer simulations.

6. Computer Simulations

The two rule selection methods are applied to real-world pattern classification problems in order to evaluate their classification performance. One method, described in section 3, is based on the Michigan approach, and the other method, described in section 4, is based on the Pittsburgh approach. First, the classification performance of each rule selection method is examined by a wine classification problem [16], using all patterns as training data. The wine data set is a 13-dimensional three-class pattern classification problem with 178 patterns (59 from Class 1, 71 from Class 2, and 48 from Class 3). Next we examine the classification performance of each rule selection method on test data in order to evaluate its generalization ability. As test problems for examining the performance on test data, we use wine data, cancer data [17], and credit approval data [18]. The cancer data set is a 9-dimensional two-class pattern classification problem with 85 patterns from Class 1 and 201 patterns from Class 2. The credit approval data set is a 14-dimensional two-class pattern classification problem with 383 patterns from Class 1 and 307 patterns from Class 2. In the following, we explain each computer simulation and report its results in detail.

6.1. Classification performance on training data

We examined the classification performance of each rule selection method on training data using the wine data in computer simulations of this subsection. First we performed a computer simulation by the rule selection method based on the Michigan approach, with the following parameter specifications: the number of fuzzy rules in each population, 60; the number of iterations of the learning algorithm for the grade of certainty, 5; the learning rates $(\eta_1, \eta_2) = (0.001, 0.1)$; the weights $(w_{NCP}, w_{NMP}) = (1, 5)$; the crossover probability $P_c = 1.0$; the mutation probability $P_m = 0.1$; the replacement rate $P_{rep} = 0.2$; and the stopping condition, 500 generations. We used the six fuzzy sets in Fig. 1 as antecedent fuzzy sets. This computer simulation was iterated 10 times using the given 178 patterns as training data. The following classification rates were obtained from the 10 trials:

Best result: 100%, Average result: 100%, Worst result: 100%

This means that all 178 training patterns were correctly classified in all 10 trials.

Next, we performed a computer simulation by the rule selection method based on the Pittsburgh approach, with the following parameter specifications: crossover probability $P_c = 0.8$, number of rule sets 10. In the computer simulation, we examined the two versions of uniform crossover: one version had crossover points only between rules (see Fig. 3) and the other version had arbitrary crossover points (see Fig. 4). The other conditions were specified in the same manner as in the rule selection method based on the Michigan approach. We used the elitist strategy: the best rule set in each generation was always inherited by the next generation. This computer simulation was iterated 10 times using the given 178 patterns as training data. The following classification rates were obtained from the 10 trials.

(i) In the case of crossover points between rules:

Best result: 66.9%, Average result: 62.1%, Worst result: 47.2%

(ii) In the case of arbitrary crossover points:

Best result: 83.7%, Average result: 67.4%, Worst result: 60.7%

From the comparison between the above results obtained by the Michigan approach and the Pittsburgh approach, we can see that the rule selection method based on the Michigan approach has a high search ability. That is, this method found rule sets that could correctly classify all of the given training patterns by examining 500 rule sets (i.e., 500 generations). In contrast, the rule selection method based on the Pittsburgh approach found a rule set with an 83.7% classification rate in the best case by examining 5000 rule sets (i.e., 500 generations × 10 individuals).

For comparison, we show the classification rates reported in Corcoran and Sen [20] as simulation results produced by their genetics-based machine learning algorithm for the wine data:

Best result: 100%, Average result: 99.5%, Worst result: 98.3%

These results were obtained in Ref. 20 by 10 trials, using all of the given patterns as training data. In their genetics-based machine learning system, which was based on the Pittsburgh approach, an individual consisted of 60 nonfuzzy if–then rules, and a population with 1500 individuals was updated for 300 generations. Thus, 1500 × 300 = 450,000 rule sets with 60 nonfuzzy if–then rules were examined in each trial. Even though such a large number of rule sets was examined, the obtained classification rates are inferior to those given by our rule selection method based on the Michigan approach.

As mentioned in section 5, an advantage of the rule selection method based on the Michigan approach is that high-performance fuzzy if–then rules are inherited by the next generation with no change. In order to clarify this advantage, we performed a computer simulation by specifying the replacement rate P_{rep} as $P_{rep} = 1.0$. That is, we intentionally removed this advantage by specifying the replacement rate as $P_{rep} = 1.0$ so that all fuzzy if–then rules were replaced in the generation update. From 10 iterations of this computer simulation, the following classification rates were obtained:

Best result: 88.2%, Average result: 79.7%, Worst result: 71.3%

From these results, we can see that the search ability deteriorated significantly when the above-mentioned advantage was intentionally removed from the rule selection method based on the Michigan approach.

6.2. Classification performance on test data

In the previous subsection, we examined the performance of each rule selection method for training data in the wine classification problem. The classification rate for training data, however, does not always correctly indicate the true classification performance. Therefore, we examine the generalization ability of each rule selection method by evaluating the classification performance on test data after dividing the given patterns into training data and test data. In this subsection, we use the 10-fold cross-validation method (10CV method [17, 18]).

First we performed computer simulations on the wine data using the same parameter specifications as in the previous subsection. In the Pittsburgh approach, we examined the two versions: one with crossover points between rules (Pitt 1) and the other with arbitrary crossover points (Pitt 2). In computer simulations, the classification rate and the error rate for test data were examined after selecting a rule set that gave the maximum classification rate on training data by each method. The 10CV was iterated 10 times for the method based on the Michigan approach, and 2 times for the method based on the Pittsburgh approach. Simulation results are summarized in Table 1.

Next we examined the classification performance of each rule selection method using the cancer data. In computer simulations, we used only three linguistic values (i.e., S: small, L: large, DC: don't care) for binary discrete attributes with $\{0, 1\}$, and only four linguistic values (i.e., S: small, M: medium, L: large, DC: don't care) for ternary discrete values with $\{0, 0.5, 1\}$.

After such modification, we applied the Michigan approach and the Pittsburgh approach to the cancer data. We used the same parameter specifications as in the application to the wine data, except that the weights in the Michigan approach were specified as $(w_{NCP}, w_{NMP}) = (1, 0)$, the number of fuzzy if–then rules in each rule set was 100 in each method, and the number of iterations of the learning algorithm for the grade of certainty was 0. In the same manner as in the computer simulations on the wine data, we examined the classification performance of each method for test data using the cancer data. Simulation results are summarized in Table 2.

The cancer data were also used in Grabisch and Dispot [21] and Weiss and Kulikowski [17] for performance evaluation. In Grabisch and Dispot [21], the generalization abilities of various fuzzy classification methods were evaluated by dividing the cancer data into 50% training patterns and 50% test patterns. Grabisch and Dispot [21] reported simulation results by nine fuzzy classification methods, based on fuzzy integrals, fuzzy pattern matching, fuzzy

Table 1. Performance for test patterns (wine data)

Method	Classification rate	Error rate	Rejection rate
Michigan	94.2%	5.7%	0.1%
Pitt 1	54.8%	29.5%	15.7%
Pitt 2	59.9%	19.5%	20.6%

Table 2. Performance for test patterns (cancer data)

Method	Classification rate	Error rate	Rejection rate
Michigan	74.5%	25.5%	0.0%
Pitt 1	71.5%	26.9%	1.6%
Pitt 2	68.8%	28.4%	2.8%

nearest neighbor, fuzzy clustering, and so on. Those results can be summarized as follows:

Maximum error rate: 45.1%
Average error rate: 39.4%
Minimum error rate: 32.0%

In Weiss and Kulikowski [17], the generalization abilities of various nonfuzzy classification methods were evaluated by dividing the cancer data into 70% training patterns and 30% test patterns. Weiss and Kulikowski [17] reported simulation results for 10 nonfuzzy classification methods, such as linear discriminant functions and neural networks. Those results can be summarized as follows:

Maximum error rate: 34.7%
Average error rate: 29.1%
Minimum error rate: 22.9%

From a comparison of Table 2 with those reported results, we can see that the rule sets of fuzzy if–then rules selected by the Michigan approach and the Pittsburgh approach have high generalization abilities (i.e., low error rates) in comparison with various classification methods examined in Grabisch and Dispot [21] and Weiss and Kulikowski [17].

We also examined the classification performance of each rule selection method on credit approval data [18]. The simulation results in Table 3 were obtained by computer simulations with the same parameter specifications as in the case of the cancer data.

The credit approval data were also used in Quinlan [18] for evaluating the performance of his C4.5 algorithm.

Table 3. Performance for test patterns (credit approval)

Method	Classification rate	Error rate	Rejection rate
Michigan	85.3%	14.7%	0.0%
Pitt 1	59.9%	38.9%	1.2%
Pitt 2	64.6%	33.8%	1.6%

The error rates by the C4.5 algorithm with various parameter specifications were reported as 17.5% (worst), 15.7% (average), and 14.2% (best), as evaluated by the 10CV procedure. From a comparison of Table 3 with these reported results, we can see the effectiveness of the Michigan approach.

7. Conclusions

Two frameworks of genetics-based machine learning (i.e., the Michigan approach and the Pittsburgh approach) were applied to rule selection problems of fuzzy if–then rules. The performance of each approach was evaluated for multidimensional pattern classification problems. For the Michigan approach, we used our fuzzy classifier system [15]. For the Pittsburgh approach, we proposed a new algorithm. In the rule selection method based on the Michigan approach, each fuzzy if–then rule was handled as an individual. On the other hand, each rule set was handled as an individual in the rule selection method based on the Pittsburgh approach. A high classification ability for training data and high generalization ability for test data were obtained by the Michigan approach when we applied both approaches to real-world pattern classification problems with many attributes (i.e., wine data, cancer data, and credit approval data). Simulation results showed that the Michigan approach could search for rule sets more efficiently than the Pittsburgh approach. This is because good fuzzy if–then rules in a current generation were inherited by the next generation with no change, and poor fuzzy if–then rules were replaced with newly generated rules. On the other hand, rule sets were updated regardless of the classification performance of each fuzzy if–then rule in the Pittsburgh approach. The Michigan approach is also efficient from the point of view of computation time and memory storage. This is because a population in the Michigan approach corresponds to a single rule set, while the Pittsburgh approach has a number of rule sets as a population. Thus, computation time and memory storage in the Michigan approach are generally much smaller than in the Pittsburgh approach.

The search in the Michigan approach, however, does not always work well when we use inappropriate specifications of the two weights in the definition of the fitness value of each fuzzy if–then rule (i.e., the reward for correct classification w_{NCP}, and the penalty for misclassification w_{NMP}). Appropriate specifications of these weights are very important, especially for pattern classification problems with high error rates due to overlaps between different classes. As an example, we show simulation results on the cancer data in Fig. 5 where all of the given patterns were used as training data. Figure 5 shows how the error rate of the rule set at each generation changed when we specified

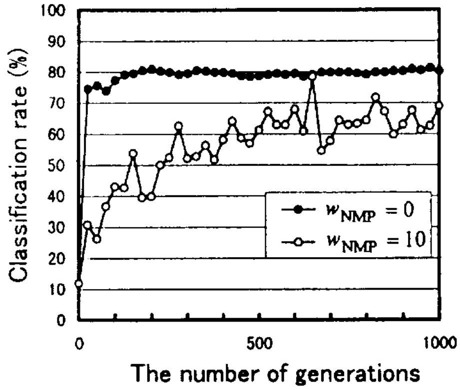


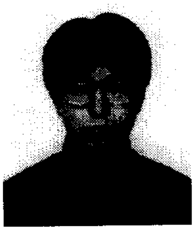
Fig. 5. Simulation results with different specifications of the misclassification penalty w_{NMP}.

the misclassification penalty as $w_{NMP} = 0$ and $w_{NMP} = 10$ (the reward for correct classification was specified as $w_{NCP} = 1$ in both cases). From Fig. 5, we can see that the search for rule sets was not efficient in the case of $w_{NMP} = 10$. This is because the genetic search in the Michigan approach sometimes goes in the direction of decreasing classification rates, because the performance of a rule set is not evaluated. In contrast, the genetic search in the Pittsburgh approach never goes in such a direction because the performance of each rule set is evaluated. These discussions suggest the necessity of fusion of the Michigan approach and the Pittsburgh approach.

REFERENCES

1. M. Sugeno. An introductory survey of fuzzy control. Inf. Sci., **36**, pp. 59–83 (1985).
2. C.C. Lee. Fuzzy logic in control systems: Fuzzy logic controller—Part I and II. IEEE Trans. Sys. Man Cybern. **20**, No. 2, pp. 404–435 (1990).
3. L.X. Wang and J.M. Mendel. Generating fuzzy rules by learning from examples. IEEE Trans. Sys. Man Cybern., **22**, pp. 1414–1427 (1992).
4. S. Abe and M.-S. Lan. A classifier using fuzzy rules extracted directly from numerical data. Proc. 2nd IEEE Int. Conf. on Fuzzy Systems, pp. 1191–1198 (1993).
5. D.E. Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison–Wesley (1989).
6. J.H. Holland. Adaptation in Natural and Artificial Systems. MIT Press (1993).
7. H. Nomura, I. Hayashi, and N. Wakami. A self-tuning method of fuzzy reasoning by genetic algorithm. Proc. 1992 Int. Fuzzy Systems and Intelligent Control Conf., pp. 236–245.
8. H. Ishigami, T. Fukuda, T. Shibata, and F. Arai. Structure optimization of fuzzy neural network by genetic algorithm. Fuzzy Sets and Systems, **71**, pp. 257–264 (1995).
9. M. Valenzuela-Rendón. The fuzzy classifier system: A classifier system for continuously varying variables. Proc. 4th ICGA, pp. 346–353 (1991).
10. H. Ishibuchi, K. Nozaki, and H. Tanaka. Pattern recognition using distributed fuzzy rules. Trans. Institute of Systems, Control and Information Engineers, **4**, No. 12, pp. 517–526 (1991). (in Japanese)
11. H. Ishibuchi, K. Nozaki, and H. Tanaka. Distributed representation of fuzzy rules and its application to pattern classification. Fuzzy Sets and Systems, **51**, pp. 21–32 (1992).
12. H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka. Selecting fuzzy if–then rules for classification problems using genetic algorithms. IEEE Trans. Fuzzy Syst., **3**, No. 3, pp. 260–270 (1995).
13. H. Ishibuchi, K. Nozaki, and N. Yamamoto. Construction of fuzzy classification systems with rectangular fuzzy rules using genetic algorithms. Fuzzy Sets and Systems, **65**, pp. 237–253 (1994).
14. H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka. Selection of fuzzy if–then rules by a genetic method. Trans. I.E.I.C.E. (A), **J76**, pp. 1465–1473 (1993). (in Japanese)
15. H. Ishibuchi, T. Nakashima, and T. Murata. A fuzzy classifier system that generates fuzzy if–then rules for pattern classification problems. Proc. 2nd IEEE ICEC, pp. 759–764 (1995).
16. M. Forina et al. Wine recognition data. Available via anonymous FTP from ics.uci.edu in directory/pub/machine-learning-databases/wine (1992).
17. S.M. Weiss and C.A. Kulikowski. Computer Systems that Learn. Morgan Kaufmann (1991).
18. J.R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann (1991).
19. K. Nozaki, H. Ishibuchi, and H. Tanaka. Knowledge acquisition with trainable fuzzy classification system. J. Japan Ind. Manage. Assoc., **45**, No. 5, pp. 430–441 (1994). (in Japanese)
20. A.L. Corcoran and S. Sen. Using real valued genetic algorithms to evolve rule sets for classification. Proc. 1st IEEE ICEC, pp. 120–124 (1994).
21. M. Grabisch and F. Dispot. A comparison of some methods of fuzzy classification on real data. Proc. 2nd Int. Conf. on Fuzzy Logic and Neural Networks, pp. 659–662 (1992).

AUTHORS (from left to right)

Hisao Ishibuchi graduated in 1985 from Kyoto University, where he received a master's degree in 1987. He then became a research associate, a lecturer in 1993, and an associate professor in 1994 at Osaka Prefecture University, where he also obtained a D.Eng. degree. He was a visiting researcher at the University of Toronto from August 1994 to March 1995 and from July 1997 to March 1998. His current research includes applications of fuzzy set theory, neural networks, and genetic algorithms. He is a member of IEEE, IFSA, and INNS.

Tomoharu Nakashima received his B.S. and M.S. degrees from Osaka Prefecture University in 1995 and 1997, respectively. He is currently a Ph.D. candidate. His research interests include the application of classifier systems to pattern classification problems and the applications of reinforcement learning. He is a member of IEEE.

Tadahiko Murata received his B.S., M.S., and Ph.D. degrees from Osaka Prefecture University in 1994, 1996, and 1997, respectively. He is currently a research associate at Ashikaga Institute of Technology. His research interests include genetic algorithms, scheduling problems, and pattern classification problems. He is a member of the Japan Society for Fuzzy Theory and Systems, the Institute of Systems, Control and Information Engineers, the Japan Industrial Management Association, and IEEE.