# Comparison between Fuzzy and Interval Partitions in Evolutionary Multiobjective Design of Rule-Based Classification Systems

Hisao Ishibuchi and Yusuke Nojima
Graduate School of Engineering, Osaka Prefecture University
1-1 Gakuen-cho, Sakai, Osaka 599-8531, Japan
{hisaoi, nojima}@cs.osakafu-u.ac.jp

***Abstract*- This paper compares fuzzy rules with interval rules through computational experiments on benchmark data sets from the UCI database using an evolutionary multiobjective rule selection method. In the design of fuzzy and interval rule-based systems for classification problems, we use three types of partitions: homogeneous fuzzy partitions, inhomogeneous entropy-based interval partitions, and inhomogeneous fuzzy partitions derived from the interval partitions. A large number of rule-based systems are designed from each type of partitions using our evolutionary multiobjective rule selection method with three objectives: to maximize the number of correctly classified training patterns, to minimize the number of rules, and to minimize the total number of antecedent conditions. Experimental results show that the fuzzification of interval rules improves their generalization ability for many data sets.**

## I. INTRODUCTION

One goal in the design of fuzzy rule-based classification systems is to extract interpretable knowledge for human users in the form of fuzzy if-then rules. High interpretability of extracted knowledge is the main advantage of fuzzy rule-based classification systems over other non-linear systems such as neural networks and support vector machines. It is, however, very difficult to design fuzzy rule-based systems with both high accuracy and high interpretability due to the tradeoff between accuracy and interpretability (i.e., the tradeoff between the accuracy maximization and the complexity minimization of fuzzy rule-based systems). In the 1990s, a large number of learning methods were proposed to improve the accuracy of fuzzy rule-based systems. Those techniques were often based on learning algorithms of neural networks and optimization techniques in evolutionary computation. Such an attempt to improve the accuracy of fuzzy rule-based systems, however, tends to degrade their interpretability. In the 2000s, the importance of not only the accuracy but also the interpretability in the design of fuzzy rule-based classification systems was pointed out by some researchers (e.g., see [1], [2]). For further discussions on the accuracy-interpretability tradeoff, see Casillas et al. [3], [4].

One of the first attempts to simultaneously perform the accuracy maximization and the complexity minimization of fuzzy rule-based classification systems was GA-based rule selection of Ishibuchi et al. [5], [6] in the mid-1990s. They used the following fitness function in fuzzy rule selection:

$$fitness(S) = w_1 \cdot f_1(S) - w_2 \cdot f_2(S), \quad (1)$$

where $S$ is a set of fuzzy rules, $f_1(S)$ is the number of correctly classified training patterns by $S$, $f_2(S)$ is the number of fuzzy rules in $S$, and $w_1$ and $w_2$ are positive constants. A standard single-objective genetic algorithm was used to maximize the fitness function in (1). The GA-based rule selection was extended to the case of two objectives in [7] where a simple multiobjective genetic algorithm was used to find a large number of non-dominated rule sets of the following two-objective rule selection problem:

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

This formulation was further extended to the case of three objectives in [8], [9] as follows:

$$\text{Maximize } f_1(S) \text{ and minimize } f_2(S) \text{ and } f_3(S), \quad (3)$$

where $f_3(S)$ is the total number of antecedent conditions of fuzzy rules in $S$. Since the number of antecedent conditions of each rule is often referred to as the rule length, $f_3(S)$ can be viewed as the total rule length.

The three-objective fuzzy rule selection method in [9] consists of candidate rule generation and genetic rule selection. First a data mining technique is used to efficiently generate a prespecified number of promising candidate rules based on a heuristic rule evaluation measure. Then an evolutionary multiobjective optimization algorithm is used to find a large number of non-dominated subsets of candidate rules with respect to the three objectives in (3). This two-stage approach is also applicable to the design of non-fuzzy interval rule-based classification systems [10].

In this paper, we compare fuzzy and interval rule-based classification systems with each other through computational experiments using the two-stage rule selection method [9], [10]. We use three types of partitions of continuous attributes to generate candidate rules: homogeneous (i.e., uniform) fuzzy partitions, inhomogeneous entropy-based interval partitions, and inhomogeneous fuzzy partitions derived from the interval partitions. Using each type of partitions, a prespecified number of candidate rules are generated based on a heuristic rule evaluation measure. Then an evolutionary multiobjective optimization algorithm is used to find a large number of non-dominated rule sets with respect to the three objectives in (3) from the candidate rules. By examining the classification performance of obtained non-dominated rule sets, we compare the three types of partitions.

## II. THREE TYPES OF PARTITIONS

Let us assume that we have $m$ labeled patterns $\mathbf{x}_p = (x_{p1}, ..., x_{pn})$, $p = 1,2,...,m$ from $M$ classes as training data (i.e., we have an $n$-dimensional $M$-class pattern classification problem with $m$ training patterns). We use if-then rules of the following type for our $n$-dimensional pattern classification problem:

$$\text{Rule } R_q : \text{If } x_1 \text{ is } A_{q1} \text{ and ... and } x_n \text{ is } A_{qn} \text{ then Class } C_q \text{ with } CF_q,\ q = 1,2,...,N_{\text{rule}}, \quad (4)$$

where $R_q$ is the label of the $q$-th rule, $\mathbf{x} = (x_1, ..., x_n)$ is an $n$-dimensional pattern vector, $A_{qi}$ is an antecedent fuzzy set or interval, $C_q$ is a consequent class, $CF_q$ is a rule weight, and $N_{\text{rule}}$ is the number of fuzzy rules. The consequent class $C_q$ and the rule weight $CF_q$ of each rule $R_q$ are specified from compatible training patterns with its antecedent part $\mathbf{A}_q = (A_{q1}, ..., A_{qn})$ in a heuristic manner [11]-[13]. The rule weight $CF_q$ is used as the strength of $R_q$ when new patterns are to be classified by the rule-based system with the $N_{\text{rule}}$ rules in (4). In the case of fuzzy rules, classification boundaries can be adjusted by changing the rule weight of each fuzzy rule without modifying the membership functions of antecedent fuzzy sets [11]-[13]. On the other hand, classification boundaries can not be adjusted by the rule weight of each rule in the case of interval rules. In this case, the discretization of each attribute into antecedent intervals has a dominant effect on classification boundaries.

Homogenous fuzzy partitions were frequently used in the design of fuzzy rule-based classification systems with high interpretability. Examples of such fuzzy partitions are shown in Fig. 1. One advantage of homogeneous fuzzy partitions over inhomogeneous ones is high linguistic interpretability of each antecedent fuzzy set. Since an appropriate granularity of fuzzy partitions is not usually known for each attribute, we simultaneously use multiple fuzzy partitions with different granularity. In our computational experiments, we use the four fuzzy partitions in Fig. 1 (i.e., 14 fuzzy sets) and "*don't care*". Thus the total number of possible combinations of antecedent fuzzy sets is $15^n$ for our $n$-dimensional problem.

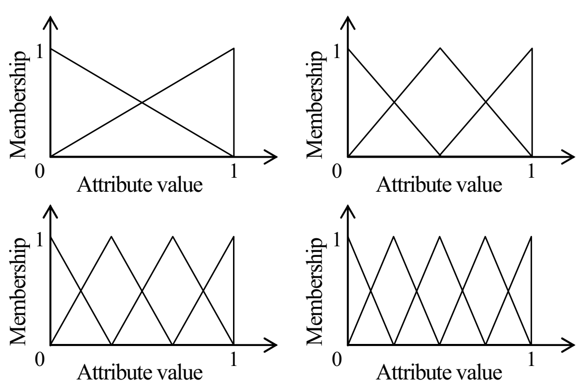


Fig. 1. Homogeneous fuzzy partitions.

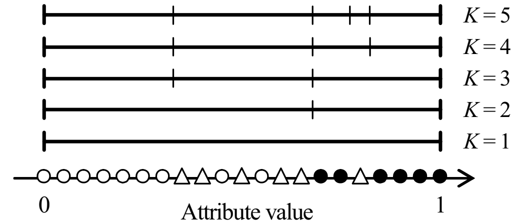


Fig. 2. Illustration of interval partitions.

In the area of machine learning, a number of discretization methods of continuous attributes into disjoint intervals have been proposed in the literature [14]-[17]. We use the following class entropy measure [14] to divide a continuous attribute into $K$ intervals:

$$H(A_1, ..., A_K) = -\sum_{j=1}^{K} \frac{|D_j|}{|D|} \sum_{h=1}^{M} \left( \frac{|D_{jh}|}{|D_j|} \cdot \log_2 \frac{|D_{jh}|}{|D_j|} \right), \quad (5)$$

where $A_j$ is an interval, $D_j$ is the set of training patterns in the interval $A_j$, and $D_{jh}$ is the set of training patterns from Class $h$ in $D_j$. The cardinality of each subset of training patterns is denoted by $|\cdot|$ such as $|D_j|$ in (5). Using an optimal splitting method [17], we can efficiently find the optimal $(K-1)$ cutting points that minimize the class entropy measure in (5). As in Fig. 1, we simultaneously use multiple interval partitions for generating interval rules. In our computational experiments, we use five partitions in Fig. 2 corresponding to $K = 1,2,3,4,5$ where $K = 1$ means "*don't care*" since the domain interval is not divided in this case.

The number of possible combinations of antecedent intervals is $15^n$ for our $n$-dimensional problem since each antecedent interval $A_{qi}$ in (4) can assume one of 15 intervals in Fig. 2. It should be noted that each antecedent interval $A_{qi}$ can be viewed as having the following membership function.

$$\mu_{A_{qi}}(x_i) = \begin{cases} 1, & \text{if } A_{qi}^L \le x_i \le A_{qi}^U \\ 0, & \text{otherwise} \end{cases}, \quad (6)$$

where $A_{qi}^L$ and $A_{qi}^U$ are the lower and upper bounds of the antecedent interval $A_{qi}$, respectively. Using (6), we can handle fuzzy and interval rules in the same framework.

As shown in Fig. 3, inhomogeneous fuzzy partitions can be derived from interval partitions [18]. We also examine such inhomogeneous fuzzy partitions in this paper.

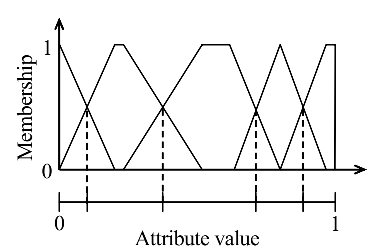


Fig. 3. Inhomogeneous fuzzy partitions.

## III. MULTIOBJECTIVE RULE SELECTION

Our evolutionary multiobjective rule selection method [9], [10] consists of the two stages: candidate rule generation and genetic rule selection. Each stage is briefly explained in the following subsections.

### *A. Candidate Rule Generation*

In this stage, a prespecified number of promising rules are chosen from possible rules (e.g., $15^n$ rules in computational experiments) based on a heuristic rule evaluation measure. A number of heuristic rule evaluation measures (e.g., support, confidence and their product) were examined in our former study [19]. We use the following measure in this paper:

$$f_{\text{SLAVE}}(R_q) = s(\mathbf{A}_q \Rightarrow \text{Class } C_q) - \sum_{\substack{h=1 \\ h \neq C_q}}^{M} s(\mathbf{A}_q \Rightarrow \text{Class } h), \qquad (7)$$

where $s(\cdot)$ is the support measure of fuzzy or interval rules, which is defined as follows:

$$s(\mathbf{A}_q \Rightarrow \text{Class } h) = \frac{1}{m} \sum_{\mathbf{x}_p \in \text{Class } h} \mu_{\mathbf{A}_q}(\mathbf{x}_p). \qquad (8)$$

In this formulation, $\mu_{\mathbf{A}_q}(\mathbf{x}_p)$ is the compatibility grade of each training pattern $\mathbf{x}_p$ from Class $h$ with the antecedent part $\mathbf{A}_q = (A_{q1}, ..., A_{qn})$. We use the product operator to calculate the compatibility grade as

$$\mu_{\mathbf{A}_q}(\mathbf{x}_p) = \mu_{A_{q1}}(x_{p1}) \cdot \;...\; \cdot \mu_{A_{qn}}(x_{pn}), \qquad (9)$$

where $\mu_{A_{qi}}(\cdot)$ is the membership function of the antecedent fuzzy set (or interval) $A_{qi}$. The heuristic rule evaluation measure in (7) is a modified version of a rule evaluation criterion used in an iterative fuzzy GBML (genetics-based machine learning) algorithm called SLAVE [20].

We only examine short rules with a few antecedent conditions when we choose a prespecified number of candidate rules for each class. This is because we want to construct interpretable rule-based systems (i.e., because it is very difficult for human users to intuitively understand long rules with many antecedent conditions). More specifically, we choose 300 rules with the largest values of the rule evaluation measure in (7) for each class among short rules of length three or less in computational experiments except for the case of the sonar data set with 60 attributes. For the sonar data set, we only examine short rules of length two or less. The total number of candidate rules is $300M$ where $M$ is the number of classes.

### *B. Genetic Rule Selection*

Let us assume that $N$ (i.e., $N = 300M$) rules have been extracted as candidate rules using the rule evaluation measure in (7). A subset $S$ of the $N$ candidate rules is handled as an individual in genetic rule selection, which is represented by a binary string of length $N$ as $S = s_1 s_2 \cdots s_N$ where $s_j = 1$ and $s_j = 0$ mean that the $j$-th candidate rule is included in $S$ and excluded from $S$, respectively. The classification performance of each rule set $S$ is evaluated by classifying all the given training patterns by $S$. We use a single winner rule-based method where each pattern $\mathbf{x}_p$ is classified by a single winner rule $R_w$ chosen from the rule set $S$ as

$$\mu_{\mathbf{A}_w}(\mathbf{x}_p) \cdot CF_w = \max\{\mu_{\mathbf{A}_q}(\mathbf{x}_p) \cdot CF_q \mid R_q \in S\}. \qquad (10)$$

The winner rule $R_w$ has the maximum product of the compatibility grade $\mu_{\mathbf{A}_q}(\mathbf{x}_p)$ and the rule weight $CF_q$ in the rule set $S$. If multiple rules have the same maximum product but different consequent classes for $\mathbf{x}_p$, the classification of $\mathbf{x}_p$ is rejected. The classification is also rejected if no rule is compatible with $\mathbf{x}_p$ (i.e., $\mu_{\mathbf{A}_q}(\mathbf{x}_p) = 0$ for $\forall R_q \in S$).

For executing multiobjective rule selection with respect to the three objectives in (3), we use a well-known state-of-the-art evolutionary multiobjective optimization algorithm called NSGA-II [21]. In the NSGA-II algorithm, we use two problem-specific heuristic tricks. One is biased mutation probabilities where a larger probability is assigned to the mutation from 1 to 0 than that from 0 to 1. This heuristic trick is used to efficiently decrease the number of rules in each rule set by the mutation operation. The other is the removal of unnecessary rules. Since we use the single winner-based method for classifying each pattern, some rules in $S$ may be chosen as winner rules for no pattern. We can remove those rules without degrading the number of correctly classified training patterns (i.e., $f_1(S)$). At the same time, the removal of such an unnecessary rule decreases the number of rules (i.e., $f_2(S)$) and the total rule length (i.e., $f_3(S)$). Thus we remove all rules that are not selected as winner rules for any training pattern from the rule set $S$. The removal of unnecessary rules is performed for each rule set after $f_1(S)$ is calculated and before $f_2(S)$ and $f_3(S)$ are calculated.

A large number of non-dominated rule sets with respect to the three objectives are found by a single run of the NSGA-II algorithm with the two heuristic tricks. Some rule sets have high accuracy and low interpretability, and other rule sets have low accuracy and high interpretability.

## IV. COMPUTATIONAL EXPERIMENTS

Through computational experiments on five data sets in Table 1 from the UC Irvine Machine Learning Repository, we compare the three types of partitions in Section II (i.e., homogeneous fuzzy partitions, inhomogeneous entropy-based interval partitions, and inhomogeneous fuzzy partitions derived from the interval partitions). The last two columns of Table 1 show the reported error rates in [17] where several versions of the C4.5 algorithm were examined. We use the ten-fold cross-validation (10-CV) technique to examine the generalization ability (i.e., accuracy on unseen test patterns). In each iteration in the 10-CV technique, 300 candidate rules are extracted from training patterns (90% of the whole data set) for each class using one of the three types of partitions. Then the NSGA-II algorithm is applied to the $300M$ candidate rules to find a number of non-dominated rule sets.

TABLE 1. DATA SETS IN COMPUTATIONAL EXPERIMENTS.

| Data set | Attributes | Patterns | Classes | C4.5 in [17] | |
|---|---|---|---|---|---|
| | | | | Best | Worst |
| Breast W | 9 | 683* | 2 | 5.1 | 6.0 |
| Glass | 9 | 214 | 6 | 27.3 | 32.2 |
| Heart C | 13 | 297* | 5 | 46.3 | 47.9 |
| Sonar | 60 | 208 | 2 | 24.6 | 35.8 |
| Wine | 13 | 178 | 3 | 5.6 | 8.8 |

* Incomplete patterns with missing values are not included.

The NSGA-II algorithm is executed using the following parameter specifications:

Population size: 200 strings,
Crossover probability: 0.8 (uniform crossover),
Biased mutation probabilities:
$p_{\rm m}(0 \rightarrow 1) = 1/300M$ and $p_{\rm m}(1 \rightarrow 0) = 0.1$,
Stopping condition: 5000 generations.

When the execution is terminated, a number of non-dominated rule sets are obtained. The classification rate of each non-dominated rule set is calculated for the training patterns and the remaining test patterns. We also examine the classification performance of an ensemble classification system of non-dominated rule sets where the final classification decision on each pattern is made by the simple majority vote by the non-dominated rule sets. The 10-CV technique is applied to each data set 10 times using different divisions into ten subsets of the same size. That is, our multiobjective rule selection method is executed 100 times (i.e., $10 \times 10$-CV) for each data set. This means that we obtain 100 collections of non-dominated rule sets. A single ensemble classification system is constructed from each collection. The average classification rate is calculated over non-dominated rule sets with the same number of rules and the same total rule length for training patterns as well as for test patterns. The average classification rate is also calculated over 100 ensemble classification systems.

In Table 2 and Table 3, we show the average error rates by ensemble classification systems constructed from each type of partitions. From Table 2, we can see that the best error rates on training patterns are obtained from the interval partitions on the average. On the other hand, good results are obtained for test patterns from the inhomogeneous fuzzy partitions derived from the interval partitions and the homogeneous fuzzy partitions in Table 3. These observations suggest that the fuzzification of the interval partitions improves the generalization ability of interval rules for test patterns while it decreases the accuracy on training patterns.

TABLE 2. ERROR RATES ON TRAINING PATTERNS BY ENSEMBLE CLASSIFICATION SYSTEMS. THE BEST RESULT FOR EACH DATA SET IS SHOWN BY BOLDFACE.

| Data set | Homogeneous fuzzy | Interval | Inhomogeneous fuzzy |
|---|---|---|---|
| Breast W | 1.76 | **1.27** | 1.50 |
| Glass | 20.98 | 16.49 | **15.63** |
| Heart C | 28.40 | **16.36** | 19.61 |
| Sonar | 9.91 | **6.46** | 8.53 |
| Wine | **0.01** | **0.01** | 0.39 |

TABLE 3. ERROR RATES ON TEST PATTERNS BY ENSEMBLE CLASSIFICATION SYSTEMS. THE BEST RESULT FOR EACH DATA SET IS SHOWN BY BOLDFACE.

| Data set | Homogeneous fuzzy | Interval | Inhomogeneous fuzzy |
|---|---|---|---|
| Breast W | 3.63 | **3.32** | 3.34 |
| Glass | 38.94 | 32.73 | **31.56** |
| Heart C | 47.68 | 47.62 | **46.46** |
| Sonar | **22.59** | 24.56 | 22.93 |
| Wine | **4.38** | 4.73 | 4.72 |

Next we examine the relation between the accuracy and the complexity of obtained rule sets in detail. Experimental results on the Cleveland heart disease data set (Heart C) are shown in Fig. 4 for training patterns and Fig. 5 for test patterns. In these figures, a single spot (i.e., open circle, triangle, or closed circle) shows the average error rate of obtained rule sets with the same number of rules and the same total rule length. In Fig. 4, we can observe the tradeoff between the classification accuracy on training patterns and the complexity of rule sets. That is, the error rates on training patterns monotonically decrease with the number of fuzzy rules in Fig. 4. Since not only the number of fuzzy rules but also the total rule length are taken into account in our multiobjective rule selection method, several non-dominated rule sets have the same number of rules as shown in Fig. 4. They are different from each other in the total rule length. With respect to the comparison among the three partitions, better results are obtained from the interval partitions and the inhomogeneous fuzzy partitions than the homogeneous fuzzy partitions in Fig. 4. Fuzzification of the interval partitions slightly increases the error rates on training patterns.

Totally different results are obtained for test patterns in Fig. 5. While the error rates on training patterns monotonically decrease with the number of rules in Fig. 4, the increase in the number of rules degrades the classification accuracy on test patterns in Fig. 5. That is, we can observe the overfitting to training patterns in Fig. 5. We can also see in Fig. 5 that the fuzzification of the interval partitions improves the generalization ability of interval rules.

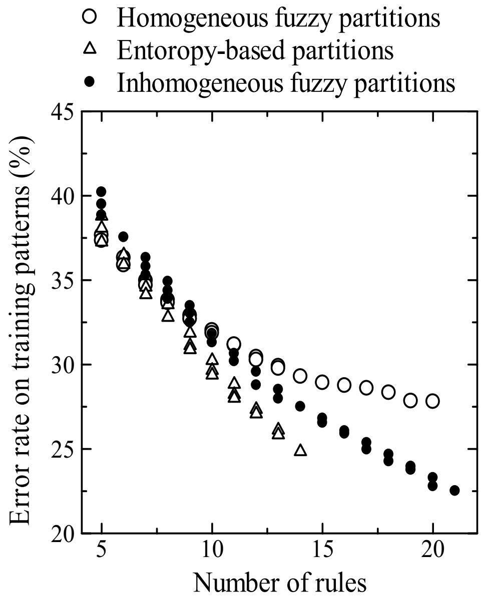


Fig. 4. Error rates on training patterns of the Cleveland heart disease data set (Heart C in Table 1).

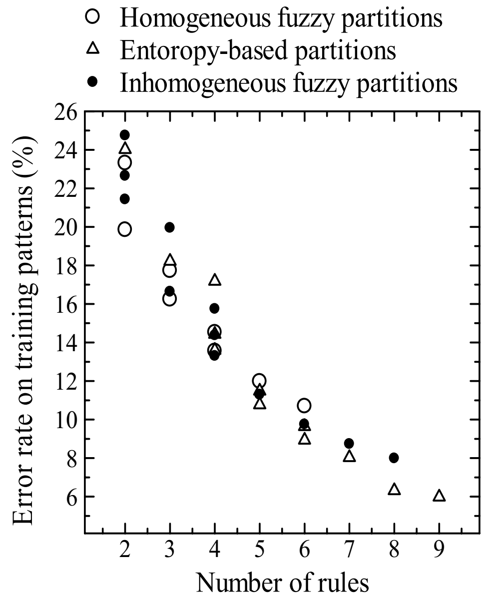


Fig. 6. Error rates on training patterns of the sonar data set (Sonar in Table 1).

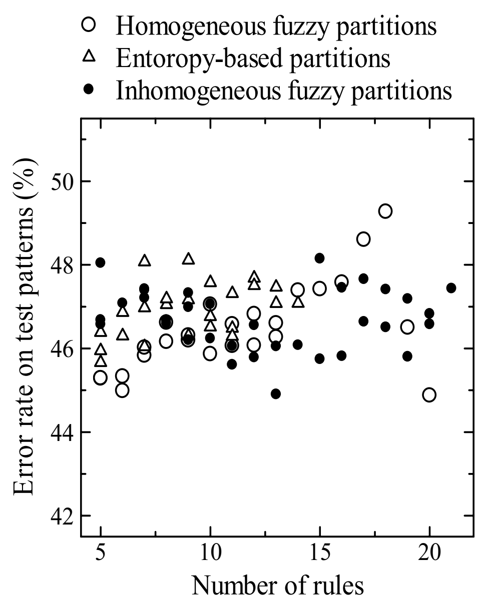


Fig. 5. Error rates on test patterns of the Cleveland heart disease data set (Heart C in Table 1).

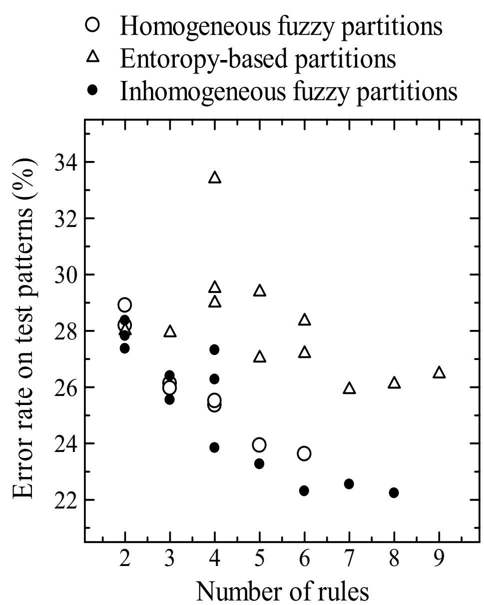


Fig. 7. Error rates on test patterns of the sonar data set (Sonar in Table 1).

In Fig. 6 and Fig. 7, we show experimental results on the sonar data set. As in Fig. 4, we can observe the tradeoff between the error rates on training patterns and the number of rules in Fig. 6. The best results on training patterns are obtained from the interval partitions in Fig. 6. On the other hand, Fig. 7 on test patterns is different from Fig. 5. The deterioration in the error rates due to the increase in the number of rules is not clear in Fig. 7 on the sonar data set (i.e., there is no clear overfitting).

As we can see from the comparison between Fig. 5 and Fig. 7, the relation between the generalization ability and the complexity of rule-based systems is problem-dependent. We can examine such a problem-specific characteristic feature for each classification problem using our multiobjective rule selection method. This is because a large number of non-dominated rule sets with different accuracy and different complexity are obtained by its single run while only a single rule set is usually obtained by single-objective approaches.

## V. CONCLUSIONS

In this paper, we compared three types of partitions of continuous attributes in the design of rule-based classification systems: Homogeneous fuzzy partitions, inhomogeneous entropy-based interval partitions, and inhomogeneous fuzzy partitions derived from the interval partitions. It was shown through computational experiments that the fuzzification of the interval partitions improved the generalization ability of interval rules while it slightly deteriorated the classification accuracy on training patterns. An interesting observation is that the homogeneous fuzzy partitions were not always inferior to the inhomogeneous ones when they were compared in terms of the generalization ability for unseen test patterns.

As we have already stressed in this paper, the advantage of our multiobjective approach is that a large number of non-dominated rule sets with different accuracy and different complexity can be obtained by its single run. From those rule sets, we can visually examine the relation between the accuracy and the complexity of rule-based systems. Such knowledge on the accuracy-complexity tradeoff structure is useful for human users in the design (or choice) of a final rule-based classification system for a particular pattern classification problem at hand because the accuracy-complexity tradeoff structure is strongly problem-dependent.

## ACKNOWLEDGMENT

The authors would like to thank Mr. Satoshi Namba for his help with computational experiments. The authors also thank the financial support from the Okawa Foundation for Information and Telecommunications.

## REFERENCES

[1] M. Setnes and H. Roubos, "GA-based modeling and classification: Complexity and performance," *IEEE Trans. on Fuzzy Systems*, vol. 8, no. 5, pp. 509-522, 2000.

[2] Y. Jin, "Fuzzy modeling of high-dimensional systems: Complexity reduction and interpretability improvement," *IEEE Trans. on Fuzzy Systems*, vol. 8, no. 2, pp. 212-221, 2000.

[3] J. Casillas, O. Cordon, F. Herrera, and L. Magdalena (eds.), *Interpretability Issues in Fuzzy Modeling*, Springer-Verlag, 2003.

[4] J. Casillas, O. Cordon, F. Herrera, and L. Magdalena (eds.), *Accuracy Improvements in Linguistic Fuzzy Modeling*, Springer-Verlag, 2003.

[5] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Construction of fuzzy classification systems with rectangular fuzzy rules using genetic algorithms," *Fuzzy Sets and Systems*, vol. 65, no. 2/3, pp. 237-253, 1994.

[6] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selecting fuzzy if-then rules for classification problems using genetic algorithms," *IEEE Trans. on Fuzzy Systems*, vol. 3, no. 3, pp. 260-270, 1995.

[7] H. Ishibuchi, T. Murata, and I. B. Turksen, "Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems," *Fuzzy Sets and Systems*, vol. 89, no. 2, pp. 135-150, 1997.

[8] H. Ishibuchi, T. Nakashima, and T. Murata, "Three-objective genetics-based machine learning for linguistic rule extraction," *Information Sciences*, vol. 136, no. 1-4, pp. 109-133, 2001.

[9] H. Ishibuchi and T. Yamamoto, "Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining," *Fuzzy Sets and Systems*, vol. 141, no. 1, pp. 59-88, 2004.

[10] H. Ishibuchi and S. Namba, "Evolutionary multiobjective knowledge extraction for high-dimensional pattern classification problems," *Lecture Notes in Computer Science 3242: Proc. of PPSN VIII*, pp. 1123-1132, Springer-Verlag, 2004.

[11] H. Ishibuchi and T. Nakashima, "Effect of rule weights in fuzzy rule-based classification systems," *IEEE Trans. on Fuzzy Systems*, vol. 9, no. 4, pp. 506-515, 2001.

[12] H. Ishibuchi and T. Yamamoto, "Rule weight specification in fuzzy rule-based classification systems," *IEEE Trans. on Fuzzy Systems* (in press).

[13] H. Ishibuchi, T. Nakashima, and M. Nii, *Classification and Modeling with Linguistic Information Granules*, Springer-Verlag, 2004.

[14] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[15] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," *Proc. of 12th International Conference on Machine Learning*, pp. 194-202, 1995.

[16] J. R. Quinlan, "Improved use of continuous attributes in C4.5," *Journal of Artificial Intelligence Research*, vol. 4, pp. 77-90, 1996.

[17] T. Elomaa and J. Rousu, "General and efficient multisplitting of numerical attributes," *Machine Learning*, vol. 36, no. 3, pp. 201-244, 1999.

[18] H. Ishibuchi and T. Yamamoto, "Performance evaluation of fuzzy partitions with different fuzzification grades," *Proc. of 2002 IEEE International Conference on Fuzzy Systems*, pp. 1198-1203, 2002.

[19] H. Ishibuchi and T. Yamamoto, "Comparison of heuristic criteria for fuzzy rule selection in classification problems," *Fuzzy Optimization and Decision Making*, vol. 3, no. 2, pp. 119-139, 2004.

[20] A. Gonzalez and R. Perez, "SLAVE: A genetic learning system based on an iterative approach," *IEEE Trans. on Fuzzy Systems*, vol. 7, no. 2, pp. 176-191, 1999.

[21] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.