Effects of Three-Objective Genetic Rule Selection on the Generalization Ability of Fuzzy Rule-based Systems

Hisao Ishibuchi and Takashi Yamamoto

Department of Industrial Engineering, Osaka Prefecture University,
1-1 Gakuen-cho, Sakai, Osaka 599-8531, Japan
{hisaoi, yama}@ie.osakafu-u.ac.jp

Abstract. One advantage of evolutionary multiobjective optimization (EMO) algorithms over classical approaches is that many non-dominated solutions can be simultaneously obtained by their single run. This paper shows how this advantage can be utilized in genetic rule selection for the design of fuzzy rule-based classification systems. Our genetic rule selection is a two-stage approach. In the first stage, a pre-specified number of candidate rules are extracted from numerical data using a data mining technique. In the second stage, an EMO algorithm is used for finding non-dominated rule sets with respect to three objectives: to maximize the number of correctly classified training patterns, to minimize the number of rules, and to minimize the total rule length. Since the first objective is measured on training patterns, the evolution of rule sets tends to overfit to training patterns. The question is whether the other two objectives work as a safeguard against the overfitting. In this paper, we examine the effect of the three-objective formulation on the generalization ability (i.e., classification rates on test patterns) of obtained rule sets through computer simulations where many non-dominated rule sets are generated using an EMO algorithm for a number of high-dimensional pattern classification problems.

1 Introduction

Fuzzy rule-based systems are universal approximators of nonlinear functions as multi-layer feedforward neural networks. These two models have been applied to various problems such as control, function approximation and pattern classification. The main advantage of fuzzy rule-based systems is their comprehensibility because each fuzzy rule is linguistically interpretable. In many studies on the design of fuzzy rule-based systems, however, emphasis has been mainly placed on their accuracy rather than their comprehensibility. Thus the performance maximization has been the primary objective. Recently the tradeoff between the accuracy and the comprehensibility was discussed in some studies [20], [21], [26]-[29]. While those studies took into account several criteria related to the accuracy and the comprehensibility, the design of fuzzy rule-based systems was handled in the framework of single-objective optimization. That is, those studies tried to find a single fuzzy rule-based system by considering

both the accuracy and the comprehensibility. One of the first studies on fuzzy rule-based systems in the framework of multiobjective optimization was a two-objective rule selection [9] where genetic algorithms were used for finding non-dominated rule sets with respect to the classification accuracy and the number of fuzzy rules. The two-objective rule selection was extended to the case of three objectives [12] where the total rule length was considered as the third objective in addition to the above-mentioned two objectives in [9]. See [2] for further discussions on the tradeoff between the accuracy and the comprehensibility of fuzzy rule-based systems.

If compared with standard optimization problems, an additional difficulty in the design of classification systems is that the maximization of any accuracy measure does not always mean the maximization of their actual performance. This is because the accuracy of classification systems is measured on training patterns while their actual performance should be measured on unseen test patterns. That is, any accuracy measure is just an estimation of the actual performance. The maximization of any accuracy measure often leads to the overfitting to training patterns, which degrades the actual performance of classification systems on test patterns. Thus we need some sort of safeguard for preventing the overfitting. A weighted sum of accuracy and complexity measures is often used as a safeguard against the overfitting to training patterns. This paper examines the usefulness of multiobjective formulations as a safeguard. In the three-objective formulation in [12], the number of fuzzy rules and their total length were used as complexity measures together with an accuracy measure. While those complexity measures were originally introduced for obtaining comprehensible fuzzy rule-based systems, we examine their usefulness as a safeguard against the overfitting. That is, we examine the effect of those complexity measures in the three-objective formulation on the generalization ability (i.e., classification rates on test patterns) of obtained fuzzy rule-based classification systems.

In this paper, we first briefly describe fuzzy rules and fuzzy reasoning for fuzzy rule-based classification in Section 2. Then we explain our two-stage approach [16] to the design of fuzzy rule-based systems in Section 3. In the first stage, a pre-specified number of fuzzy rules are generated as candidate rules from training patterns using a data mining technique. In the second stage, non-dominated rule sets are found from the candidate rules by an EMO algorithm. Simulation results on several data sets are reported in Section 4 where the generalization ability of obtained rule sets on test patterns is examined. Simulation results clearly show that the two complexity measures improve not only the comprehensibility of obtained rule sets but also their generalization ability on test patterns. Finally Section 5 summarizes this paper.

2 Fuzzy Rule-Based Classification Systems

Let us assume that we have m training patterns $\mathbf{x}_p = (x_{p1}, ..., x_{pn})$, $p = 1,2,...,m$ from M classes where x_{pi} is the attribute value of the p-th training pattern for the i-th

attribute ($i = 1,2,...,n$). For our n-dimensional M-class pattern classification problem, we use fuzzy rules of the following form:

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

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 (i.e., linguistic value such as *small* and *large*), C_q is a class label, and CF_q is a rule weight. Fuzzy rules of this type were first used for classification problems in [13]. For other types of fuzzy rules, see [4], [11], [23].

We define the compatibility grade of each training pattern $\mathbf{x}_p$ with the antecedent part $\mathbf{A}_q = (A_{q1}, ..., A_{qn})$ using the product operator as

$$\mu_{\mathbf{A}_q}(\mathbf{x}_p) = \mu_{A_{q1}}(x_{p1}) \cdot \mu_{A_{q2}}(x_{p2}) \cdot \ ... \ \cdot \mu_{A_{qn}}(x_{pn}), \quad p = 1, 2, ..., m, \quad (2)$$

where $\mu_{A_{qi}}(\cdot)$ is the membership function of A_{qi}. For determining the consequent class C_q, we calculate the confidence of the fuzzy association rule "$\mathbf{A}_q \Rightarrow \text{Class } h$" for each class as an extension of its non-fuzzy version [1] as follows [8], [19]:

$$c(\mathbf{A}_q \Rightarrow \text{Class } h) = \sum_{\mathbf{x}_p \in \text{Class } h} \mu_{\mathbf{A}_q}(\mathbf{x}_p) \Big/ \sum_{p=1}^{m} \mu_{\mathbf{A}_q}(\mathbf{x}_p), \quad h = 1, 2, ..., M . \quad (3)$$

The confidence is the same as the fuzzy conditional probability [30]. The consequent class C_q is specified by identifying the class with the maximum confidence:

$$c(\mathbf{A}_q \Rightarrow \text{Class } C_q) = \max\{c(\mathbf{A}_q \Rightarrow \text{Class } h) \,|\, h = 1,2,...,M\} . \quad (4)$$

On the other hand, the rule weight CF_q is specified as follows:

$$CF_q = c(\mathbf{A}_q \Rightarrow \text{Class } C_q) - \sum_{\substack{h=1 \\ h \neq C_q}}^{M} c(\mathbf{A}_q \Rightarrow \text{Class } h) . \quad (5)$$

The rule weight of each fuzzy rule has a large effect on the classification ability of fuzzy rule-based systems [10]. There are several alternative definitions of rule weights (see [17]). Better results were obtained in [17] from the above definition in (5) than the direct use of the confidence (i.e., $CF_q = c(\mathbf{A}_q \Rightarrow \text{Class } C_q)$) when we used a single winner-based method for classifying new patterns.

In this paper, we use a single winner-based fuzzy reasoning method [13]. For other fuzzy reasoning methods for pattern classification, see [4], [11], [23]. Let S be the set of fuzzy rules in our fuzzy rule-based system. A single winner rule R_w is chosen from the rule set S for an input pattern $\mathbf{x}_p$ as

$$\mu_{\mathbf{A}_w}(\mathbf{x}_p) \cdot CF_w = \max\{\mu_{\mathbf{A}_q}(\mathbf{x}_p) \cdot CF_q \ | \ R_q \in S\} . \quad (6)$$

Since the winner rule is chosen based on not only the compatibility grade but also the rule weight, high classification accuracy can be achieved by adjusting the rule weight of each fuzzy rule without modifying each antecedent fuzzy set [10].

3 Heuristic Rule Extraction and Genetic Rule Selection

Genetic rule selection was proposed for designing fuzzy rule-based classification systems with high accuracy and high comprehensibility in [14], [15]. A small number of fuzzy rules were selected from a large number of candidate rules based on a scalar fitness function defined as a weighted sum of the number of correctly classified training patterns and the number of fuzzy rules. A two-objective genetic algorithm was used in [9] for finding non-dominated rule sets. Genetic rule selection was further extended to the following three-objective optimization problem in [12]:

$$\text{Maximize } f_1(S)\text{, minimize } f_2(S)\text{, and minimize } f_3(S)\text{,} \tag{7}$$

where S is a subset of candidate rules, $f_1(S)$ is the number of correctly classified training patterns by the rule set 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. The number of antecedent conditions of each fuzzy rule is referred to as the rule length in this paper. As clearly shown in [12], the use of the average rule length as the third objective $f_3(S)$ leads to counter-intuitive results. Thus we use the total rule length as $f_3(S)$ in (7).

When we use K linguistic values and "*don't care*" as antecedent fuzzy sets, the total number of possible combinations of antecedent fuzzy sets is $(K+1)^n$. In early studies [9], [14], [15], all combinations were examined for generating candidate rules. Thus genetic rule selection was applicable only to low-dimensional problems (e.g., iris data with four attributes). On the other hand, only short fuzzy rules were examined for generating candidate rules in [12] where genetic rule selection was applied to higher-dimensional problems (e.g., wine data with 13 attributes).

In our former study [16], we suggested the use of a data mining technique for extracting a pre-specified number of candidate rules in a heuristic manner. That is, genetic rule selection was extended to a two-stage approach with heuristic rule extraction and genetic rule selection. Our two-stage approach is applicable to high-dimensional problems (e.g., sonar data with 60 attributes).

3.1 Heuristic Rule Extraction

In the field of data mining, association rules are often evaluated by two rule evaluation criteria: support and confidence. In the same manner as the fuzzy version of confidence in (3), the definition of support [1] can be also extended to the case of fuzzy association rules as follows [8], [19]:

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

The product of the confidence and the support was used in our former study [16] on the two-stage approach. Seven heuristic criteria were compared with each other in

[18] where good results were obtained from the following criterion:

$$f_{\mathrm{SLAVE}}(R_q) = s(\mathbf{A}_q \Rightarrow \text{Class } C_q) - \sum_{\substack{h=1 \\ h \neq C_q}}^{M} s(\mathbf{A}_q \Rightarrow \text{Class } h) \,. \tag{9}$$

This is a modified version of a rule evaluation criterion used in an iterative fuzzy GBML (genetics-based machine learning) algorithm called SLAVE [3], [7].

In our heuristic rule extraction, a pre-specified number of candidate rules with the largest values of the SLAVE criterion are found for each class. For designing fuzzy rule-based systems with high comprehensibility, only short rules are examined as candidate rules. The restriction on the rule length is consistent with the third objective (i.e., the total rule length) of our three-objective rule selection problem in (7).

3.2 Genetic Rule Selection

Let us assume that N fuzzy rules have been extracted as candidate rules using the SLAVE criterion. A subset S of the N candidate rules is handled as an individual in EMO algorithms, which is represented by a binary string of the length N as

$$S = s_1 s_2 \cdots s_N \,, \tag{10}$$

where $s_j = 1$ and $s_j = 0$ mean that the j-th candidate rule is included in S and excluded from S, respectively.

A simple multiobjective genetic algorithm [24] based on a scalar fitness function with random weights was used in our former studies [9], [12], [16]. Recently several EMO algorithms with much higher search ability have been proposed (for example, NSGA-II [5], PAES [22], and SPEA [32]). Since each rule set is represented by a binary string in our three-objective rule selection problem in (7), most EMO algorithms are applicable. In this paper, we use the NSGA-II because its high search ability has been demonstrated in [5] and its implementation is relatively easy.

We use two problem-specific heuristic tricks in the NSGA-II. One is biased mutation where a larger probability is assigned to the mutation from 1 to 0 than that from 0 to 1. This is for efficiently decreasing the number of fuzzy rules in each rule set. The other is the removal of unnecessary rules. Since we use the single winner-based method for classifying each pattern, some fuzzy rules in S may be chosen as winner rules for no patterns. We can remove those fuzzy rules without degrading the first objective (i.e., the number of correctly classified training patterns). At the same time, the second objective (i.e., the number of fuzzy rules) and the third objective (i.e., the total rule length) are improved by removing unnecessary rules. Thus we remove all fuzzy rules that are not selected as winner rules for any training patterns from the rule set S. The removal of unnecessary rules is performed after the first objective is calculated for each rule set and before the second and third objectives are calculated.

4 Computer Simulations

4.1 Data Sets

We used six data sets in Table 1 available from the UCI ML repository (http://www.ics.uci.edu~mlearn/). Data sets with missing values are marked by "*" in the third column of Table 1. Since we did not use incomplete patterns with missing values, the number of patterns in the third column does not include those patterns with missing values. As benchmark results, we cited simulation results by Elomaa and Rousu [6] in Table 1. They applied six variants of the C4.5 algorithm [25] to 30 data sets in the UCI ML repository. The performance of each variant was examined by ten iterations of the whole ten-fold cross-validation (10-CV) procedure [25], [31]. We show in the last two columns of Table 1 the best and worst error rates on test patterns among the six variants reported in [6] for each data set.

Table 1. Data sets used in our computer simulations.

Data set	Number of attributes	Number of patterns	Number of classes	Error rate by C4.5 in [6]	
				Best	Worst
Breast W	9	683*	2	5.1	6.0
Diabetes	8	768	2	25.0	27.2
Glass	9	214	6	27.3	32.2
Heart C	13	97*	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.

4.2 Simulation Conditions

We applied our two-stage approach to six data sets in Table 1. All attribute values were normalized into real numbers in the unit interval [0, 1]. As antecedent fuzzy sets, we used 14 triangular fuzzy sets generated from four fuzzy partitions with different granularities in Fig. 1 because we did not know an appropriate granularity of the fuzzy partition for each attribute. In addition to the 14 triangular fuzzy sets, we also used *"don't care"* as an additional antecedent fuzzy set. We generated 300 fuzzy rules of the length two or less for each class of the sonar data set as candidate rules in a greedy manner using the SLAVE criterion. That is, the best 300 candidate rules with the largest values of the SLAVE criterion were found for each class. For the other data sets, we generated 300 fuzzy rules of the length three or less for each class. Thus the total number of candidate rules was 300M where M is the number of classes.

The NSGA-II was employed for finding non-dominated rule sets from 300M candidate rules. We used the following parameter values in the NSGA-II:

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

We also examined the combination of 2000 strings and 500 generations. Almost the same results were obtained from this combination and the above parameter values.

For evaluating the generalization ability of obtained rule sets, we used the 10-CV technique as in [6]. First each data set was randomly divided into ten subsets of the same size. One subset was used as test patterns while the other nine subsets were used as training patterns. Our two-stage approach was applied to training patterns for finding non-dominated rule sets. The generalization ability of obtained rule sets was evaluated by classifying test patterns. This train-and-test procedure was iterated ten times so that all the ten subsets were used as test patterns. As in [6], we iterated the whole 10-CV procedure ten times using different data partitions. Thus our two-stage approach was executed 100 times in total for each data set.

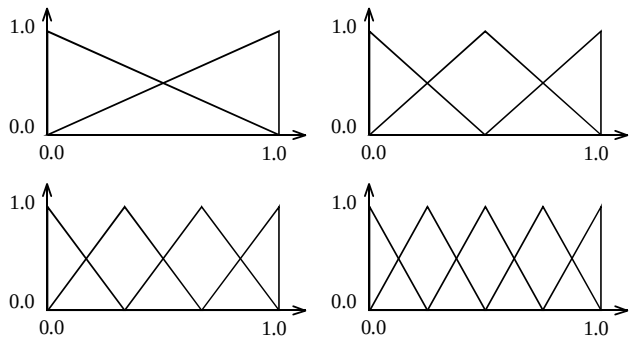


Fig. 1. Four fuzzy partitions used in our computer simulations.

4.3 Simulation Results

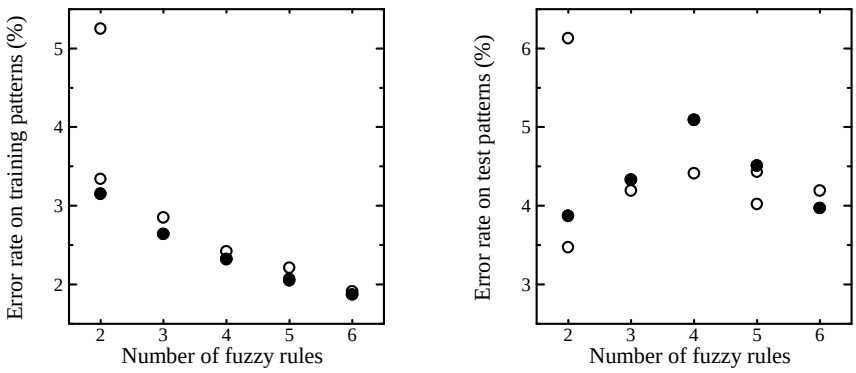
Wisconsin Breast Cancer Data Set: The NSGA-II was applied to the Wisconsin breast cancer data set (Breast W in Table 1) 100 times. From each run of the NSGA-II, 11.5 non-dominated rule sets were obtained on the average. We calculated error rates of each non-dominated rule set on training patterns and test patterns. Simulation results are summarized in Table 2 where the last column shows the number of runs from which the corresponding rule sets (with respect to the number of fuzzy rules and the average rule length) were obtained. For example, rule sets including four rules of the average length 1.50 were obtained from 72 out of 100 runs. We omit from Table 2 some rare combinations of the number of fuzzy rules and the average rule length that were obtained from only 30 (out of 100) runs or less.

Table 2. Performance of obtained rule sets for the Wisconsin breast cancer data set.

Number of rules	Average length	Average error rate		Number of runs
		Training	Test	
0	0.00	100.00	100.00	100
1	1.00	35.43	35.43	100
2	1.00	5.25	6.13	100
2	1.50	3.34	3.47	100
2	2.00	3.15	3.87	92
3	1.33	2.85	4.19	79
3	1.67	2.64	4.33	92
4	1.50	2.42	4.41	72
4	1.75	2.32	5.09	36
5	1.40	2.21	4.43	35
5	1.60	2.05	4.51	61
5	1.80	2.07	4.02	35
6	1.50	1.91	4.19	35
6	1.67	1.87	3.97	45

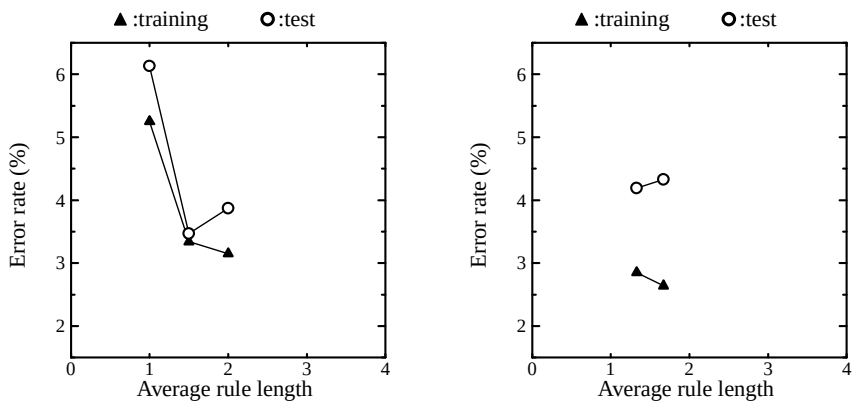
We can see from Table 1 and Table 2 that the generalization ability of many rule sets outperforms the best result of the C4.5 algorithm in Table 1 (i.e., 5.1% error rate). For visually demonstrating the tradeoff between the accuracy and the complexity, error rates on training patterns in Table 2 are shown in Fig. 2 (a) where the smallest error rate is denoted by a closed circle for each number of fuzzy rules. Thus closed circles in Fig. 2 (a) can be viewed as simulation results obtained from the two-objective formulation without the third objective (i.e., total rule length). From this figure, we can observe a clear tradeoff between the error rate on training patterns and the number of fuzzy rules. If we use a weighted sum of the accuracy on training patterns and the number of fuzzy rules as a scalar fitness function, one of the closed circles is obtained as a single optimal solution. For example, the right-most closed circle may be obtained when the weight for the accuracy is very large. On the other hand, error rates on test patterns are shown in Fig. 2 (b). Rule sets corresponding to closed circles in Fig. 2 (a) are also denoted by closed circles in Fig. 2 (b). From Fig. 2 (b), we can observe the overfitting due to the increase in the number of fuzzy rules. That is, error rates on test patterns in Fig. 2 (b) tend to increase with the number of fuzzy rules while error rates on training patterns in Fig. 2 (a) monotonically decrease. Moreover we can notice another kind of overfitting in Fig. 2 (b) from the difference between the closed circle and the smallest error rate on test patterns for each number of fuzzy rules. That is the overfitting due to the increase in the average rule length.

For demonstrating the overfitting due to the increase in the average rule length, a part of the simulation results in Table 2 are depicted in Fig. 3. It should be noted that the horizontal axis of Fig. 3 is the average rule length while it was the number of fuzzy rules in Fig. 2. Fig. 3 (a) and Fig. 3 (b) show error rates of obtained rule sets with two and four fuzzy rules, respectively. From Fig. 3, we can see that error rates on test patterns increased as the average rule length increased in some cases.



(a) Error rates on training patterns. (b) Error rates on test patterns.

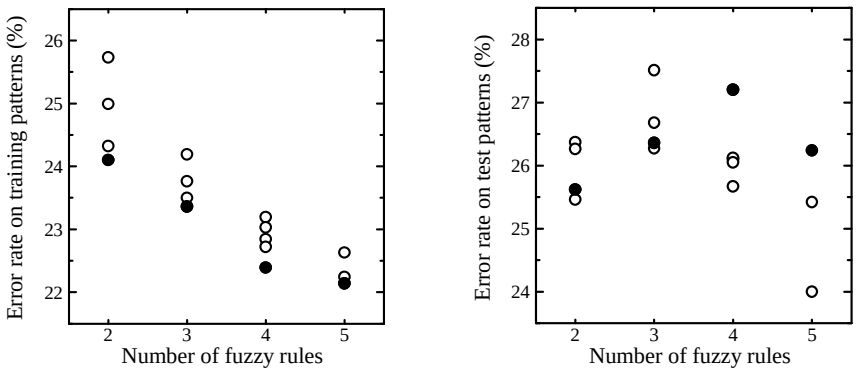
Fig. 2. Error rates of obtained rule sets for the Wisconsin breast cancer data set.



(a) Rule sets with two rules. (b) Rule sets with four rules.

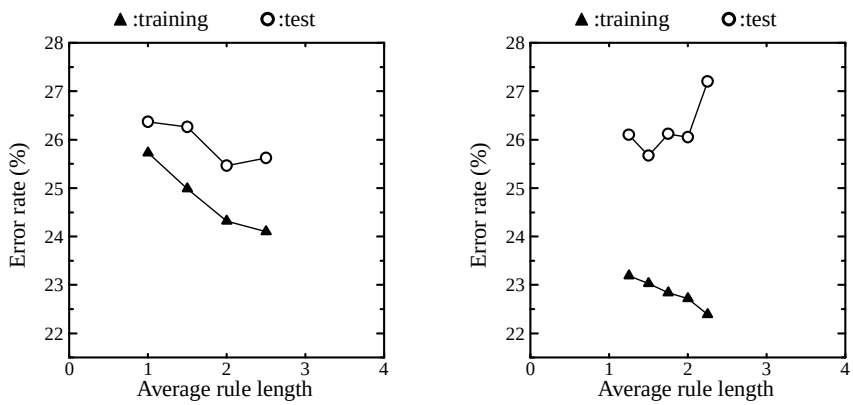
Fig. 3. Error rates of obtained rule sets with the same number of fuzzy rules and different average rule length for the Wisconsin breast cancer data set.

Diabetes Data Set: In the same manner as Fig. 2 and Fig. 3, simulation results on the diabetes data set (Diabetes in Table 1) are summarized in Fig. 4 and Fig. 5. In Fig. 4 (a), we can observe a clear tradeoff between the accuracy on training patterns and the number of fuzzy rules. On the other hand, error rates on test patterns increase in some cases in Fig. 4 (b) as the number of fuzzy rules increases. That is, we can observe the overfitting due to the increase in the number of fuzzy rules in Fig. 4 (b). The overfitting due to the increase in the average rule length is clear in Fig. 5 (b) where we show error rates by obtained rule sets including four rules. We can see from the comparison between Fig. 4 (b) and Table 1 that the generalization ability of many rule sets is slightly inferior to the best result of the C4.5 algorithm (i.e., 25.0% error rate) and slightly superior to its worst result (i.e., 27.2% error rate).



(a) Error rates on training patterns. (b) Error rates on test patterns.

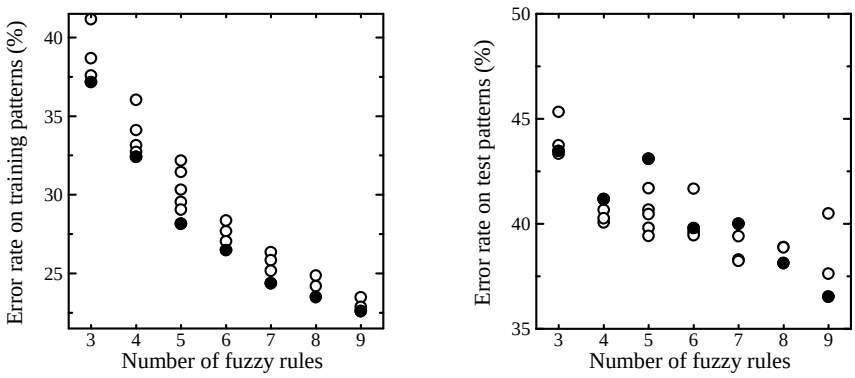
Fig. 4. Error rates of obtained rule sets for the diabetes data set.



(a) Rule sets with two rules. (b) Rule sets with four rules.

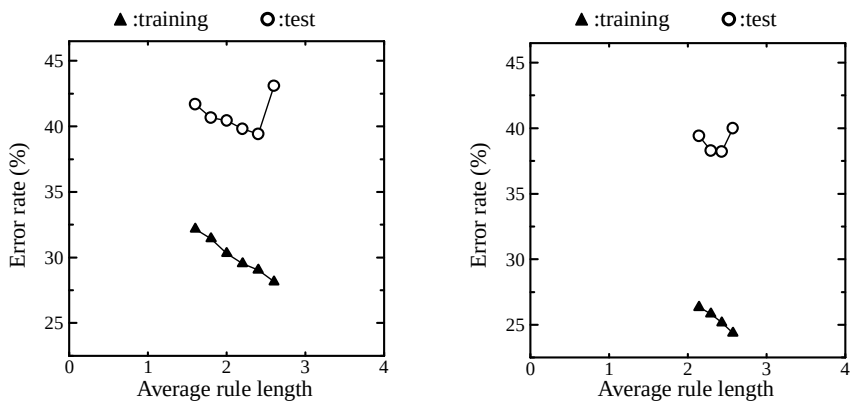
Fig. 5. Error rates of obtained rule sets with the same number of fuzzy rules and different average rule length for the diabetes data set.

Glass Identification Data Set: Simulation results on the glass identification data set (Glass in Table 1) are summarized in Fig. 6 and Fig. 7. In Fig. 6 (b), the overfitting due to the increase in the number of fuzzy rules is not clear. This result may suggest that the generalization ability of fuzzy rule-based systems can be further improved by using more fuzzy rules and/or adjusting each fuzzy rule (e.g., adjusting the rule weight). This is also suggested from the fact that the generalization ability on test patterns in Fig. 6 (b) is significantly inferior to the best result of the C4.5 algorithm in Table 1 (i.e., 27.3% error rate). On the other hand, we can observe the overfitting due to the increase in the average rule length in Fig. 7. For this data set, Sanchez et al.[27] reported a 42.1% error rate on test patterns by fuzzy rule-based systems with 8.5 rules on the average. Many rule sets in Fig. 6 (b) outperform the reported result in [27].



(a) Error rates on training patterns. (b) Error rates on test patterns.

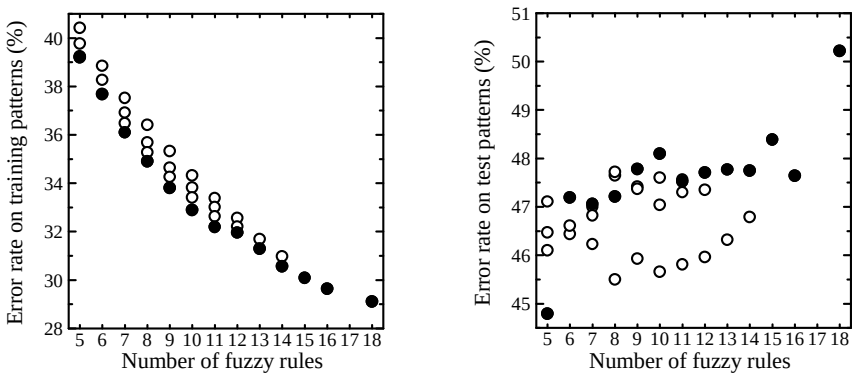
Fig. 6. Error rates of obtained rule sets for the glass identification data set.



(a) Rule sets with five rules. (b) Rule sets with seven rules.

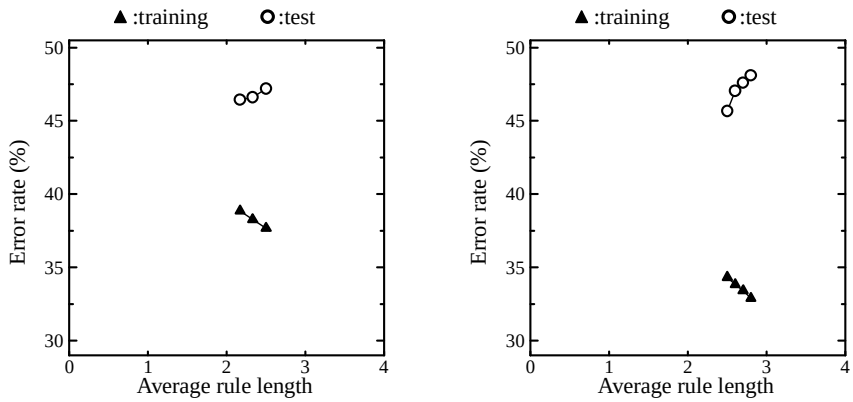
Fig. 7. Error rates of obtained rule sets with the same number of fuzzy rules and different average rule length for the glass identification data set.

Cleveland Heart Disease Data Set: Simulation results on the Cleveland heart disease data set (Heart C in Table 1) are summarized in Fig. 8 and Fig. 9. In Fig. 8 (a), we can observe a clear tradeoff between the accuracy on training patterns and the number of fuzzy rules. On the other hand, the overfitting due to the increase in the number of fuzzy rules is clear in Fig. 8 (b). That is, error rates on test patterns tend to increase with the number of fuzzy rules in Fig. 8 (b) while error rates on training patterns in Fig. 8 (a) monotonically decrease. The worst result on test patterns in Fig. 8 (b) corresponds to the best result on training patterns in Fig. 8 (a). The overfitting due to the increase in the average rule length is also clear in Fig. 9. The generalization ability of some rule sets in Fig. 8 (b) outperforms the best result of the C4.5 algorithm in Table 1 (i.e., 46.3% error rate).



(a) Error rates on training patterns. (b) Error rates on test patterns.

Fig. 8. Error rates of obtained rule sets for the Cleveland heart disease data set.

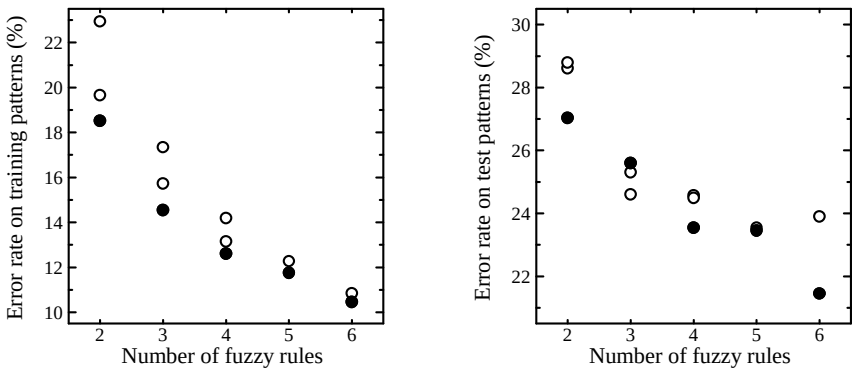


(a) Rule sets with six rules. (b) Rule sets with ten rules.

Fig. 9. Error rates of obtained rule sets with the same number of fuzzy rules and different average rule length for the Cleveland heart disease data set.

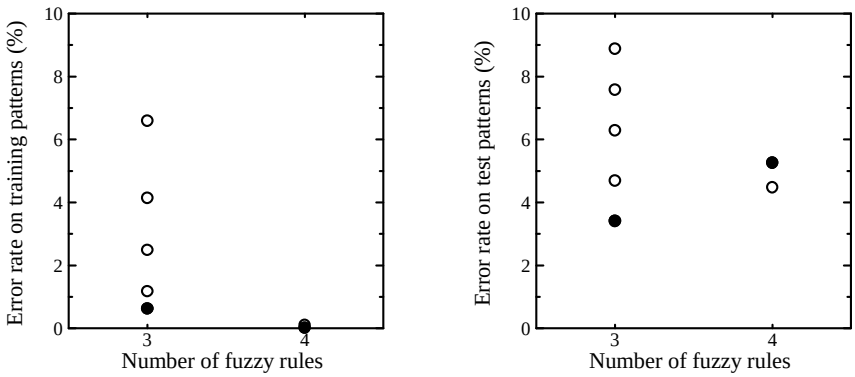
Sonar Data Set: Simulation results on the sonar data set (Sonar in Table 1) are summarized in Fig. 10. We can observe the tradeoff between the accuracy and the number of fuzzy rules in Fig. 10. The overfitting due to the increase in the number of fuzzy rules is not observed in Fig. 10 (b). The overfitting due to the increase in the average rule length is observed in the case of three fuzzy rules in Fig. 10 (b). The generalization ability of some rule sets in Fig. 10 (b) outperforms the best result of the C4.5 algorithm in Table 1 (i.e., 24.6% error rate).

Wine Recognition Data Set: Simulation results on the wine recognition data set (Wine in Table 1) are summarized in Fig. 11. The generalization ability of some rule sets in Fig. 11 (b) outperforms the best result of the C4.5 algorithm in Table 1 (i.e., 5.6% error rate).



(a) Error rates on training patterns. (b) Error rates on test patterns.

Fig. 10. Error rates of obtained rule sets for the sonar data set.



(a) Error rates on training patterns. (b) Error rates on test patterns.

Fig. 11. Error rates of obtained rule sets for the wine recognition data set.

5 Concluding Remarks

We demonstrated the effect of a three-objective formulation of fuzzy rule selection on the generalization ability of obtained rule sets through computer simulations on six data sets. We observed clear overfitting to training patterns due to the increase in the number of fuzzy rules in computer simulations on three data sets: Wisconsin, diabetes and Cleveland. For those data sets, the second objective of our three-objective formulation (i.e., minimization of the number of fuzzy rules) can work as a safeguard against the overfitting. We also observed the overfitting due to the increase in the rule length in computer simulations on all the six data sets. The two-objective formulation is not enough for those data sets where the third objective (i.e., minimization of the total rule length) is necessary as a safeguard against the overfitting. Except for the

glass identification data set and the sonar data set, the maximization of the accuracy on training patterns did not lead to the maximization of the accuracy on test patterns. Thanks to the three-objective formulation, we found fuzzy rule-based systems with high generalization ability for many data sets. Empirical analysis in this paper on the relation between the generalization ability of fuzzy rule-based systems and their complexity strongly relied on the ability of EMO algorithms to simultaneously find many non-dominated rule sets. Without this ability of EMO algorithms, we could not efficiently examine many non-dominated rule sets. Simulation results reported in this paper suggest the potential usefulness of EMO algorithms in the field of knowledge discovery and data mining.

Acknowledgment

The authors would like to thank the financial support from Japan Society for the Promotion of Science (JSPS) through Grand-in-Aid for Scientific Research (B): KAKENHI (14380194).

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, A. I.: Fast Discovery of Association Rules, in Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. (eds.) *Advances in Knowledge Discovery and Data Mining*, AAAI Press, Metro Park (1996) 307-328.
2. Casillas, J., Cordon, O., Herrera, F., and Magdalena, L. (Eds.): *Accuracy Improvement in Linguistic Fuzzy Modelling*, Physica-Verlag (2003 in press).
3. Castillo, L., Gonzalez, A., and Perez, R.: Including a Simplicity Criterion in the Selection of the Best Rule in a Genetic Fuzzy Learning Algorithm, *Fuzzy Sets and Systems* 120 (2001) 309-321.
4. Cordon, O., Del Jesus, M. J., and Herrera, F.: A Proposal on Reasoning Methods in Fuzzy Rule-Based Classification Systems, *International Journal of Approximate Reasoning* 20 (1999) 21-45.
5. Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, *IEEE Trans. on Evolutionary Computation* 6 (2002) 182 –197.
6. Elomaa, T., and Rousu, J.: General and Efficient Multisplitting of Numerical Attributes, *Machine Learning* 36 (1999) 201-244.
7. Gonzalez, A., and Perez, R.: SLAVE: A Genetic Learning System Based on an Iterative Approach, *IEEE Trans. on Fuzzy Systems* 7 (1999) 176-191.
8. Hong, T. -P., Kuo, C. -S., and Chi, S. -C.: 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 (2001) 587-604.
9. Ishibuchi, H., Murata, T., and Turksen, I. B.: Single-Objective and Two-Objective Genetic Algorithms for Selecting Linguistic Rules for Pattern Classification Problems, *Fuzzy Sets and Systems* 89 (1997) 135-149.
10. Ishibuchi, H., and Nakashima, T.: Effect of Rule Weights in Fuzzy Rule-Based Classification Systems, *IEEE Trans. on Fuzzy Systems* 9 (2001) 506-515.
11. Ishibuchi, H., Nakashima, T., and Morisawa, T.: Voting in Fuzzy Rule-Based Systems for Pattern Classification Problems, *Fuzzy Sets and Systems* 103 (1999) 223-238.

12. Ishibuchi, H., Nakashima, T., and Murata, T.: Three-Objective Genetics-Based Machine Learning for Linguistic Rule Extraction, *Information Sciences* 136 (2001) 109-133.
13. Ishibuchi, H., Nozaki, K., and Tanaka, H.: Distributed Representation of Fuzzy Rules and Its Application to Pattern Classification, *Fuzzy Sets and Systems* 52 (1992) 21-32.
14. Ishibuchi, H., Nozaki, K., Yamamoto, N., and Tanaka, H.: Construction of Fuzzy Classification Systems with Rectangular Fuzzy Rules Using Genetic Algorithms, *Fuzzy Sets and Systems* 65 (1994) 237-253.
15. Ishibuchi, H., Nozaki, K., Yamamoto, N., and Tanaka, H.: Selecting Fuzzy If-Then Rules for Classification Problems Using Genetic Algorithms, *IEEE Trans. on Fuzzy Systems* 3 (1995) 260-270.
16. Ishibuchi, H., and Yamamoto, T.: Fuzzy Rule Selection by Data Mining Criteria and Genetic Algorithms, *Proc. of Genetic and Evolutionary Computation Conference* (2002) 399-406.
17. Ishibuchi, H., and Yamamoto, T.: Comparison of Heuristic Rule Weight Specification Methods, *Proc. of 11th IEEE International Conference on Fuzzy Systems* (2002) 908-913.
18. Ishibuchi, H., and Yamamoto, T.: Comparison of Fuzzy Rule Selection Criteria for Classification Problems, in A. Abraham et al. (eds.): *Soft Computing Systems: Design, Management and Applications* (Frontiers in Artificial Intelligence and Applications, Volume 87), pp. 132-141, IOS Press, 2002.
19. Ishibuchi, H., Yamamoto, T., and Nakashima, T.: Fuzzy Data Mining: Effect of Fuzzy Discretization, *Proc. of 1st IEEE International Conference on Data Mining* (2001) 241-248.
20. Jin, Y.: Fuzzy Modeling of High-dimensional Systems: Complexity Reduction and Interpretability Improvement, *IEEE Trans. on Fuzzy Systems* 8 (2000) 212-221.
21. Jin, Y., Von Seelen, W., and Sendhoff, B.: On Generating FC^3 Fuzzy Rule Systems from Data Using Evolution Strategies, *IEEE Trans. on Systems, Man and Cybernetics - Part B: Cybernetics* 29 (1999) 829-845.
22. Knowles, J. D., and Corne, D. W.: Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy, *Evolutionary Computation* 8 (2000) 149-172.
23. Kuncheva, L. I.: *Fuzzy Classifier Design*, Physica-Verlag, Heidelberg (2000).
24. Murata, T., and Ishibuchi, H.: MOGA: Multi-Objective Genetic Algorithms, *Proc. of 2nd IEEE International Conference on Evolutionary Computation* (1995) 289-294.
25. Quinlan, J. R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo (1993).
26. Roubos, H., and Setnes, M.: Compact and Transparent Fuzzy Models and Classifiers Through Iterative Complexity Reduction, *IEEE Trans. on Fuzzy Systems* 9 (2001) 516-524.
27. Sanchez, L., Couso, I., and Corrales, J. A.: Combining GA Operators with SA Search to Evolve Fuzzy Rule Base Classifiers, *Information Sciences* 136 (2001) 175-191.
28. Setnes, M., Babuska, R., Kaymak, U., and Van Nauta Lemke, H. R.: Similarity Measures in Fuzzy Rule Base Simplification, *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics* 28 (1998) 376-386.
29. Setnes, M., Babuska, R., and Verbruggen, B.: Rule-based Modeling: Precision and Transparency, *IEEE Trans. on Systems, Man, and Cybernetics - Part C: Applications and Reviews* 28 (1998) 165-169.
30. Van den Berg, J., Kaymak, U., and Van den Bergh, W. -M.: Fuzzy Classification Using Probability Based Rule Weighting, *Proc. of 11th IEEE International Conference on Fuzzy Systems* (2002) 991-996.
31. Weiss, S. M., and Kulikowski, C. A.: *Computer Systems That Learn*, Morgan Kaufmann Publishers, San Mateo (1991).
32. Zitzler, E., and Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach, *IEEE Trans. on Evolutionary Computation* 3 (1999) 257-271.