Performance Evaluation of Three-Objective Genetic Rule Selection

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

Abstract – We examine the classification performance of fuzzy rule-based systems designed by three-objective genetic rule selection. While a single rule set is usually obtained from a single run of rule generation methods, multiple rule sets are simultaneously obtained by a single run of our rule selection method with three objectives: to maximize the number of correctly classified training patterns, to minimize the number of selected fuzzy rules, and to minimize the total rule length. Our genetic rule selection is a two-stage approach. In the first stage, a pre-specified number of candidate fuzzy rules are extracted in a heuristic manner using a data mining technique. In the second stage, a multiobjective genetic algorithm is used for finding non-dominated rule sets with respect to the three objectives. 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 of obtained non-dominated rule sets. We also examine the effect of the adjustment of rule weights, which is performed after three-objective genetic rule selection.

I. INTRODUCTION

The tradeoff between the accuracy and the complexity of fuzzy rule-based systems was often discussed in recent studies [1]-[4]. While those studies simultaneously took into account the accuracy and the complexity, the design of fuzzy rule-based systems was handled in the framework of single-objective optimization. Thus the final goal in those studies was to find a single fuzzy rule-based system. One of the first studies on fuzzy rule-based systems in the framework of multiobjective optimization was genetic rule selection [5] where a two-objective genetic algorithm was used for finding multiple non-dominated rule sets with respect to the number of correctly classified training patterns and the number of fuzzy rules. The two-objective rule selection was extended to the case of three objectives in [6] where the total rule length was also considered. See [7] for further discussions on the tradeoff between the accuracy and the complexity.

When we design fuzzy rule-based classification systems, it should be noted that the maximization of any accuracy measure does not always mean the maximization of the actual performance. This is because the accuracy is measured on training patterns while the 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 the accuracy on training patterns often leads to the overfitting, which degrades the actual performance of fuzzy rule-based classification systems on test patterns. Thus we need some sort of safeguard for preventing the overfitting. This paper examines the usefulness of complexity measures in multiobjective genetic rule selection as a safeguard against the overfitting in the design of fuzzy rule-based classification systems. In the three-objective formulation in [6], the number of fuzzy rules and their total length were used as complexity measures. While those complexity measures were originally introduced for obtaining comprehensible fuzzy rule-based classification systems, we examine their usefulness as a safeguard against the overfitting to training patterns through computer simulations where classification rates on test patterns as well as training patterns were calculated. We also examine the effect of the adjustment of rule weights on classification rates on test patterns and training patterns.

First, we briefly describe fuzzy rule-based classification in Section II. Then we explain our two-stage approach [8] to the design of fuzzy rule-based classification systems in Section III. In the first stage, a 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 generated candidate rules by a multiobjective genetic algorithm. After genetic rule selection, we apply a simple reward-punishment learning scheme of rule weights [9] to each non-dominated rule set. Thus our approach in this paper can be viewed as a three-stage algorithm. Classification rates of obtained non-dominated rule sets are reported in Section IV. Simulation results clearly show that the two complexity measures improve not only the comprehensibility of rule sets but also their classification performance on test patterns. Finally Section V concludes the paper.

II. FUZZY RULE-BASED CLASSIFICATION

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 pth training pattern for the ith attribute ($i = 1,2,...,n$). For our pattern classification problem with n attributes, we use fuzzy rules of the following form:

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

where R_q is the label of the qth fuzzy rule, $\mathbf{x} = (x_1, ..., x_n)$ is an n-dimensional pattern vector, A_{qi} is an antecedent fuzzy set, C_q is a class label, and CF_q is a rule weight.

We use the product operator for defining the compatibility grade of $\mathbf{x}_p$ with the antecedent part $\mathbf{A}_q = (A_{q1}, ..., A_{qn})$ as

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

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 rule "$\mathbf{A}_q \Rightarrow$ Class h" for each class as an extension of its crisp version [10] as follows [11], [12]:

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

The confidence in (3) is the same as the fuzzy conditional probability [13]. The consequent class C_q is specified by identifying the class with the maximum confidence:

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

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

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

The rule weight of each fuzzy rule has a large effect on the classification ability of fuzzy rule-based systems [14]. There are several alternative heuristic definitions of rule weights (see [15]). Better results were obtained in [15] from the above definition in (5) than the direct use of the confidence.

Let S be the set of fuzzy rules in our fuzzy rule-based classification system. For classifying an input pattern $\mathbf{x}_p$, a single winner rule R_w is chosen for $\mathbf{x}_p$ from the rule set S as

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

Since the winner rule is chosen based on the compatibility grade and the rule weight, the classification ability of the fuzzy rule-based system S can be improved by adjusting the rule weight of each fuzzy rule. Nozaki et al. [9] proposed a simple reward-punishment learning scheme where the rule weight CF_w of the winner rule R_w was increased as follows when a training pattern was correctly classified.

$$CF_w^{\text{New}} = CF_w^{\text{Old}} + \eta^+ \cdot (1 - CF_w^{\text{Old}}), \quad (7)$$

where η^+ is a learning rate. On the other hand, CF_w was decreased when a training pattern was misclassified:

$$CF_w^{\text{New}} = CF_w^{\text{Old}} - \eta^- \cdot CF_w^{\text{Old}}, \quad (8)$$

where η^- is a learning rate. The two learning rates η^+ and η^- are usually specified as $0 < \eta^+ << \eta^- < 1$ because the number of correctly classified training patterns is much larger than that of misclassified patterns. In this paper, we use the heuristic method in (5) for specifying rule weights in genetic rule selection where non-dominated rule sets are found. The learning scheme is applied to each non-dominated rule set after the execution of the genetic rule selection is terminated.

III. RULE GENERATION AND RULE SELECTION

Genetic rule selection was proposed for designing fuzzy rule-based classification systems with high accuracy and high comprehensibility in [16], [17] where a small number of fuzzy rules were selected from a large number of candidate rules. Genetic rule selection was extended to the following three-objective optimization problem in [6]:

$$\text{Maximize } f_1(S), \text{ and minimize } f_2(S) \text{ and } f_3(S), \quad (9)$$

where S is a subset of candidate 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 $f_3(S)$ is the total rule length in S. The number of antecedent conditions of each fuzzy rule is referred to as the rule length in this paper.

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 our early studies [5], [16], [17] on genetic rule selection, 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). In our recent study [8], 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. Since a pre-specified number of candidate rules are extracted using a data mining technique in our two-stage approach, it is applicable to high-dimensional problems (e.g., sonar data with 60 attributes). In this paper, we further extend it to a three-stage approach by incorporating the above-mentioned rule weight learning.

A. Heuristic Rule Extraction

In the field of data mining, association rules are often evaluated by two measures: *support* and *confidence* [10]. In the same manner as the fuzzy version of the confidence in (3), the definition of the support [10] can be also extended to the case of fuzzy rules as follows [11], [12]:

$$s(\mathbf{A}_q \Rightarrow \text{Class } h) = \frac{1}{m} \sum_{\mathbf{x}_p \in \text{Class } h} \mu_{\mathbf{A}_q}(\mathbf{x}_p). \quad (10)$$

The product of the confidence and the support was used in [8] for heuristic selection of candidate rules. Seven heuristic criteria were compared in [18] where good results were obtained from the following criterion:

$$f_{\mathrm{SLAVE}}(R_q) = s(\mathbf{A}_q \Rightarrow \text{Class } C_q) - \sum_{\substack{h=1 \\ h \neq C_q}}^{M} s(\mathbf{A}_q \Rightarrow \text{Class } h). \quad (11)$$

This is a modified version of a rule evaluation criterion in an iterative genetic learning algorithm called SLAVE [19].

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. This restriction on the rule length is consistent with the third objective (i.e., to minimize the total rule length) of our three-objective formulation in (9).

B. Genetic Rule Selection

Let us assume that N fuzzy rules have been extracted as candidate rules using the SLAVE criterion (i.e., N/M fuzzy rules for each class). A subset S of the N candidate rules is represented by a binary string of the length N as

$$S = s_1 s_2 \cdots s_N, \quad (12)$$

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

Recently many multiobjective genetic algorithms with high search ability were proposed (e.g., SPEA [20] and NSGA-II [21]). Since rule sets are represented by binary strings, almost all multiobjective genetic algorithms are applicable. In this paper, we use the NSGA-II because its search ability is high and its implementation is relatively easy.

We use two problem-specific heuristic procedures 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. 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 with respect to the classification accuracy. At the same time, the second and third objectives with respect to the complexity 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.

IV. Computer Simulations

A. Data Sets

We used three 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. Since we did not use incomplete patterns with missing values, the number of patterns in the third column does not include those patterns. As benchmark results, we cited simulation results by Elomaa and Rousu [22] in Table 1. They applied six variants of the C4.5 algorithm [23] 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. We show in the last two columns of Table 1 the best and worst error rates on test patterns among the six variants in [22] 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 rates in [22]	
				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

* Incomplete patterns with missing values are not included.

B. Simulation Conditions

We applied our two-stage and three-stage approaches to three 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 "*don't care*" and 14 triangular fuzzy sets in Fig. 1. We generated 300 fuzzy rules of the length three or less for each class as candidate rules using the SLAVE criterion. Thus the total number of candidate rules was $300M$ where M is the number of classes.

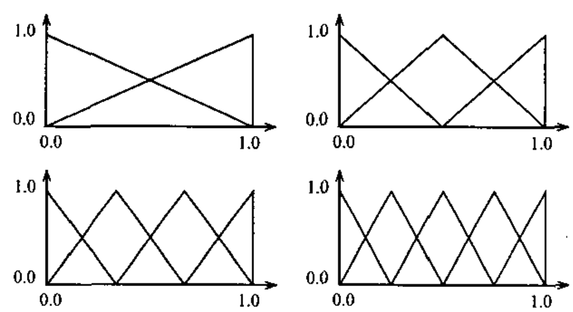


Fig. 1. Four fuzzy partitions used in our computer simulations.

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: $p_{\mathrm{m}}(0 \to 1) = 1/300M$,
$p_{\mathrm{m}}(1 \to 0) = 0.1$,
Stopping condition: 5000 generations.

For evaluating the generalization ability of obtained rule sets, we used the 10-CV technique as in [22]. 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. The train-and-test procedure was iterated ten times so that all the ten subsets were used as test patterns. As in [22], 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.

We also evaluated each non-dominated rule set with adjusted rule weights by the reward-punishment learning algorithm (i.e., each non-dominated rule set obtained by our three-stage approach). Using training patterns, the learning algorithm was iterated ten times (i.e., ten epochs or ten sweeps) for each of the obtained non-dominated rule sets. The classification rate on training patterns was calculated after each epoch. The best rule set with the largest classification rate on training patterns was chosen among ten alternatives, each of which was obtained after each epoch. The classification rate on test patterns was calculated for each of the non-dominated rule sets improved by the rule weight adjustment.

C. Simulation Results

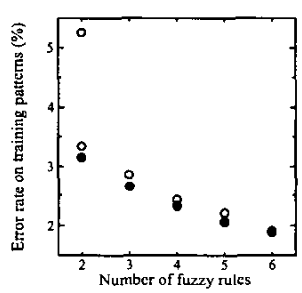
Wisconsin Breast Cancer Data Set: The NSGA-II was applied to the Wisconsin breast cancer data set (Breast W in Table 1) 100 times. From each run of the NSGA-II, 11.5 non-dominated rule sets were obtained on the average. We calculated error rates of each non-dominated rule set on training patterns and test patterns before the rule weight adjustment. 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 runs or less. 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 of rule sets, error rates on training patterns in Table 2 are shown in Fig. 2 (a) where some results with too large error rates are omitted. In Fig. 2 (a), the smallest error rate on training patterns 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. For example, the right-most closed circle is obtained when the weight for the accuracy is very large.

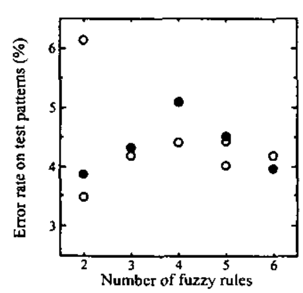
On the other hand, error rates on test patterns are shown in Fig. 2 (b). Rule sets corresponding to closed circles in Fig. 2 (a) are also denoted by closed circles in Fig. 2 (b). From Fig. 2 (b), we can observe the overfitting due to the increase in the number of fuzzy rules. That is, error rates on test patterns were increased by the increase in the number of fuzzy rules in some cases. Moreover we can observe the overfitting due to the increase in the rule length 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. This overfitting was illustrated in Fig. 3 (a) and Fig. 3 (b) for the cases of rule sets consisting of two rules and four rules, respectively.

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

Number of rules	Average length	Average error rate Training	Average error rate Test	Number of runs
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

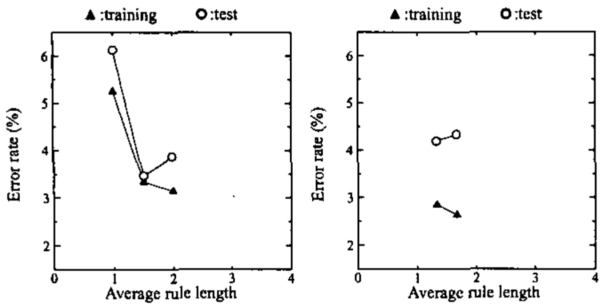


(a) Error rate on training patterns.



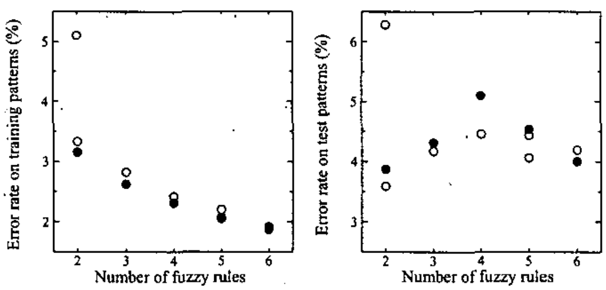
(b) Error rate on test patterns.

Fig. 2. Error rates for the Wisconsin data before rule weight learning.



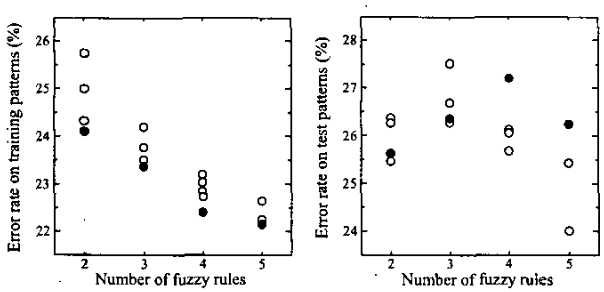
(a) Rule sets with two rules. (b) Rule sets with four rules.

Fig. 3. Error rates of some rule sets in Table 2 and Fig. 2.



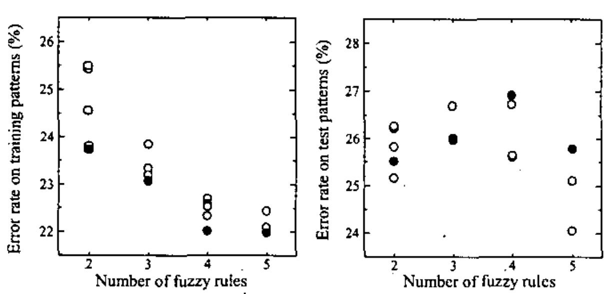
(a) Error rate on training patterns. (b) Error rate on test patterns.

Fig. 4. Error rates for the Wisconsin data after rule weight learning.



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

Fig. 5. Error rates for the diabetes data before rule weight learning.



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

Fig. 6. Error rates for the diabetes data after rule weight learning.

Fig. 4 shows simulation results after the adjustment of rule weights. We applied the learning algorithm with $\eta^+ = 0.0001$ and $\eta^- = 0.1$ to each non-dominated rule set. From the comparison between Fig. 2 and Fig. 4, we can see that the effect of the rule weight adjustment was very small in our computer simulations on the Wisconsin breast cancer data.

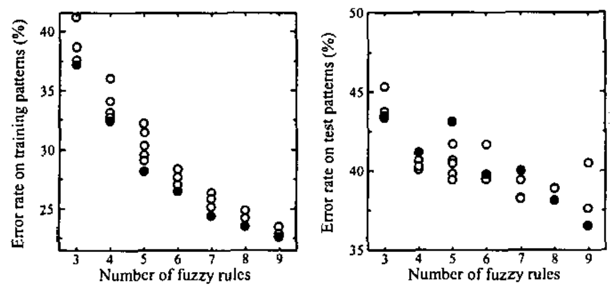
Diabetes Data Set: In the same manner as Fig. 2 and Fig. 4, simulation results on the diabetes data are summarized in Fig. 5 and Fig. 6. In Fig. 5 (a), we can observe a clear tradeoff between the accuracy on training patterns and the number of fuzzy rules. The overfitting due to the increase in the number of fuzzy rules is not clear in Fig. 5 (b). The overfitting due to the increase in the rule length, however, is clear as shown by the location of each closed circle. We can also see that the generalization ability of many rule sets in Fig. 5 (b) and Fig. 6 (b) is comparable with the reported results of the C4.5 algorithm in Table 1(i.e., best: 25.0% and worst: 27.2%).

The effect of the rule weight adjustment was large for the diabetes data set. That is, error rates of many rule sets in Fig. 5 were improved by the adjustment of rule weights in Fig. 6.

Glass Identification Data Set: Simulation results on the glass data are summarized in Fig. 7 and Fig. 8. In Fig. 7 (b), we do not observe the overfitting due to the increase in the number of fuzzy rules while the overfitting due to the increase in the rule length is clear. The generalization ability of rule sets in Fig. 7 (b) and Fig. 8 (b) are inferior to the best result of the C4.5 algorithm in Table 1 (i.e., 27.3%). The rule weight adjustment slightly improved their generalization ability in Fig. 8. Our results outperform the recently reported result (i.e., 42.1% error rate by 8.5 fuzzy rules) in [24].

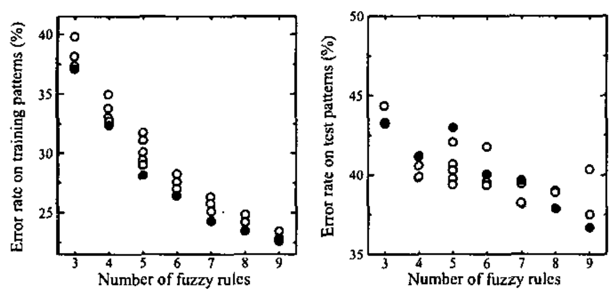
V. Conclusion

We demonstrated the effect of two complexity measures in three-objective genetic rule selection on the generalization ability of obtained rule sets. We observed the overfitting to training patterns due to the increase in the number of fuzzy rules in computer simulations on the Wisconsin breast cancer data set. For this data set, the second objective (i.e., minimization of the number of fuzzy rules) may work as a safeguard against the overfitting. We also observed the overfitting due to the increase in the rule length in computer simulations on three data sets (i.e., Wisconsin, diabetes and glass). 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. We also used a simple reward-punishment scheme for adjusting rule weights after non-dominated rule sets were obtained by genetic rule selection. The effect of the rule weight adjustment was not large in our computer simulations except for those on the diabetes data. Currently we are examining the rule weight adjustment not only after genetic rule selection but also during genetic rule selection.



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

Fig. 7. Error rates for the glass data before rule weight learning.



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

Fig. 8. Error rates for the glass data after rule weight learning.

REFERENCE

[1] 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.

[2] 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.

[3] Jin, Y.: Fuzzy Modeling of High-dimensional Systems: Complexity Reduction and Interpretability Improvement, *IEEE Trans. on Fuzzy Systems* 8 (2000) 212-221.

[4] 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.

[5] 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.

[6] Ishibuchi, H., Nakashima, T., and Murata, T.: Three-Objective Genetics-Based Machine Learning for Linguistic Rule Extraction, *Information Sciences* 136 (2001) 109-133.

[7] Casillas, J., Cordon, O., Herrera, F., and Magdalena, L. (Eds.), *Interpretability Issues in Fuzzy Modeling*, Physica-Verlag (2003 in press).

[8] 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.

[9] Nozaki, K., Ishibuchi, H., and Tanaka, H.: Adaptive Fuzzy Rule-Based Classification Systems, *IEEE Trans. on Fuzzy Systems* 4 (1996) 238-250.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] Ishibuchi, H., and Nakashima, T.: Effect of Rule Weights in Fuzzy Rule-Based Classification Systems, *IEEE Trans. on Fuzzy Systems* 9 (2001) 506-515.

[15] Ishibuchi, H., and Yamamoto, T.: Comparison of Heuristic Rule Weight Specification Methods, *Proc. of 11th IEEE International Conference on Fuzzy Systems* (2002) 908-913.

[16] 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.

[17] 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.

[18] Ishibuchi, H., and Yamamoto, T.: Comparison of Fuzzy Rule Selection Criteria for Classification Problems, *Proc. of 2nd International Conference on Hybrid Intelligent Systems* (2002) 132-141.

[19] Gonzalez, A., and Perez, R.: SLAVE: A Genetic Learning System Based on an Iterative Approach, *IEEE Trans. on Fuzzy Systems* 7 (1999) 176-191.

[20] 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.

[21] 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.

[22] Elomaa, T., and Rousu, J.: General and Efficient Multisplitting of Numerical Attributes, *Machine Learning* 36 (1999) 201-244.

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

[24] 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.