Evolutionary Multiobjective Optimization for Generating an Ensemble of Fuzzy Rule-Based Classifiers

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. In this paper, we propose an idea of using EMO algorithms for constructing an ensemble of fuzzy rule-based classifiers with high diversity. The classification of new patterns is performed based on the vote of multiple classifiers generated by a single run of EMO algorithms. Even when the classification performance of individual classifiers is not high, their ensemble often works well. The point is to generate multiple classifiers with high diversity. We demonstrate the ability of EMO algorithms to generate various non-dominated fuzzy rule-based classifiers with high diversity by their single run. Through computational experiments on some well-known benchmark data sets, it is shown that the vote of generated fuzzy rule-based classifiers leads to high classification performance on test patterns.

1 Introduction

A promising approach to the design of reliable classifiers is to combine multiple classifiers into a single one [2], [6]. Several methods have been proposed for generating multiple classifiers such as bagging [3] and boosting [8]. In the **bagging** (**b**ootstrap **agg**regat**ing**) algorithm of Breiman [3], different data sets are generated by bootstrapping (i.e., random sampling with replacement from the whole data set) for the design of multiple classifiers. Thus the design of multiple classifiers can be performed in parallel. On the other hand, multiple classifiers are sequentially designed in boosting methods such as the **AdaBoost** (**Ada**ptive **Boost**ing) algorithm of Freund & Schapire [8]. After one classifier is designed, the weight of each training pattern is updated based on the classification result (i.e., correct classification or misclassification) in the AdaBoost algorithm. The training patterns with the updated weights are used for designing another classifier. The design of a classifier and the weight update of training patterns are iterated for generating multiple classifiers. Classifier aggregation has been studied in various fields [19], [21]. For example, evolutionary computation is used for generating multiple classifiers [20], [22]. In the field of neural networks, the aggregation of multiple classifiers is often referred to as

"mixture of local experts [17], [18]". Classifier aggregation has also been studied in the field of fuzzy logic [4], [12].

The point in classifier aggregation is to generate an ensemble of classifiers with high diversity. Ideally the classification errors by each classifier should be uncorrelated. In this paper, we propose the use of evolutionary multiobjective optimization (EMO) algorithms for generating an ensemble of classifiers with high diversity. In our computational experiments, we apply the NSGA-II algorithm of Deb et al. [5] to a three-objective rule selection problem [13] for generating a number of non-dominated fuzzy rule-based classifiers with respect to the classification accuracy on training patterns, the number of fuzzy rules, and the total length of fuzzy rules. Of course, we can apply other EMO algorithms to our task. We use the NSGA-II because its implementation is relatively easy and its high performance is well-known [5].

One advantage of EMO algorithms over classical approaches is that many non-dominated solutions (i.e., classifiers in the context of this paper) can be obtained by their single run. That is, multiple classifiers are obtained by applying an EMO algorithm to training patterns just once. Through computational experiments on some well-known benchmark data sets, it is shown that high classification performance on test patterns (i.e., high generalization ability) can be obtained from the vote of non-dominated fuzzy rule-based classifiers. That is, we can design a high-performance aggregated fuzzy rule-based classifier using an EMO algorithm for generating multiple classifiers and the majority rule for classifying new patterns.

In this paper, we first briefly describe fuzzy rule-based classifiers in Section 2. Then we explain our two-stage approach [15], [16] to the design of fuzzy rule-based classifiers in Section 3. In the first stage, a pre-specified number of fuzzy rules are extracted as candidate rules from training patterns using a data mining technique. In the second stage, a number of non-dominated rule sets are found from the candidate rules by the NSGA-II algorithm. Experimental results on some well-known benchmark data sets are reported in Section 4 where the generalization ability of an ensemble of obtained non-dominated rule sets for each data set is examined using the majority rule for classifying new patterns. Finally Section 5 summarizes this paper.

2 Fuzzy Rule-Based Classifiers

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, C_q is a class label, and CF_q is a rule weight. We define the compatibility grade of each training pattern $\mathbf{x}_p$ with the antecedent part $\mathbf{A}_q = (A_{q1}, ..., A_{qn})$ of the fuzzy rule R_q in (1) using the product operator as

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

where $\mu_{A_{qi}}(\cdot)$ is the membership function of A_{qi}.

For determining the consequent class C_q and the rule weight CF_q, we first calculate the confidence of the fuzzy association rule "$\mathbf{A}_q \Rightarrow$ Class h" for each class h by extending its original definition for non-fuzzy association rules [1] as

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

The confidence $c(\cdot)$ can be viewed as a fuzzy conditional probability of Class h. The consequent class C_q is specified as the class with the maximum confidence:

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

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

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

The rule weight of each fuzzy rule has a large effect on the classification ability of fuzzy rule-based classifiers [11].

Let S be a fuzzy rule-based classifier (i.e., a set of fuzzy rules). When an input pattern $\mathbf{x}_p$ is to be classified, a single winner rule R_w is chosen from the rule set S as

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

The input pattern $\mathbf{x}_p$ is assigned to the consequent class C_w of the winner rule R_w.

In this paper, we use multiple fuzzy rule-based classifiers. An input pattern is classified by each individual classifier using the single winner-based method as shown in (6). Then the final classification is performed through the majority rule (i.e., simple majority vote scheme) based on the classification result by each individual classifier (see [12] for various voting methods for fuzzy rule-based classification).

3 Heuristic Rule Extraction and Genetic Rule Selection

Genetic rule selection was proposed for designing fuzzy rule-based classifiers with high accuracy and high comprehensibility in [14] where a scalar fitness function was defined as a weighted sum of two objectives: to maximize the number of correctly classified training patterns and to minimize the number of fuzzy rules. A two-

objective genetic algorithm was used in [10] for finding non-dominated rule sets with respect to these two objectives. Genetic rule selection was further extended to the following three-objective optimization problem in [13]:

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

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. It should be noted that the third objective $f_3(S)$ is not the average rule length but the total rule length. While we use the average rule length for describing each rule set in some parts of this paper, its use as $f_3(S)$ leads to many meaningless non-dominated rule sets [13].

A two-stage approach to the three-objective fuzzy rule selection problem in (7) was proposed for handling high-dimensional classification problems in [15], [16]. This approach is briefly explained in this section (for details, see [15], [16]).

3.1 Heuristic Rule Extraction

When we use K linguistic values and "*don't care*" as antecedent fuzzy sets for each of n attributes, the total number of possible combinations of those $(K+1)$ antecedent fuzzy sets is $(K+1)^n$. Among those combinations, a pre-specified number of candidate rules are generated in a heuristic manner using a data mining criterion. 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 the confidence in (3), the definition of the support [1] is also extended as

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

The support $s(\cdot)$ can be viewed as measuring the coverage of training patterns by the fuzzy rule. We use the following rule evaluation criterion in this paper:

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

This is a modified version of a rule evaluation criterion used in an iterative fuzzy GBML (genetics-based machine learning) algorithm called SLAVE [9].

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 classifiers with high comprehensibility, only short fuzzy rules are examined as candidate rules. This restriction on the rule length is consistent with the third objective (i.e., the total rule length) of our three-objective rule selection problem.

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.

As in our former studies [15], [16], we use two problem-specific heuristic tricks together with the NSGA-II [5] for efficiently finding non-dominated rule sets. One trick is the 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 trick is the removal of unnecessary rules, which is a kind of local search. Since we use the single winner-based method for classifying each pattern by the rule set S, some fuzzy rules in S may be chosen as winner rules for no training 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 Computational Experiments

4.1 Data Sets

We use six data sets with many numerical attributes: Wisconsin breast cancer, Diabetes, Glass, Cleveland heart disease, Sonar, and Wine. These data sets are available from the UCI ML repository (http://www.ics.uci.edu/~mlearn/). In our former study [16], we examined the performance of individual non-dominated rule sets (i.e., individual fuzzy rule-based classifiers) on each data set. In this paper, we examine the performance of their ensemble (i.e., their aggregation using the simple majority vote scheme). We evaluate the performance of the aggregated classifier on each data set by comparing it with the reported results on the same data set in Elomaa & Rousu [7] where six variants of the C4.5 algorithm [23] were examined. The performance of each variant was evaluated by ten independent iterations (with different data partitions) of the whole ten-fold cross-validation (10-CV) procedure (i.e., $10 \times 10\text{-CV}$) in [7]. We use the same performance evaluation procedure as [7].

Incomplete patterns with missing values are included in the Wisconsin beast

cancer data set and the Cleveland heart disease data set. Those patterns were not used in our computational experiments as in [16]. See UCI ML repository and [16] for details of each data set.

4.2 Experimental Conditions

As in Elomaa & Rousu [7] and our former study [16], we iterated the whole 10-CV procedure ten times using different data partitions into ten subsets. Since the whole 10-CV procedure consisted of ten iterations of the design of a classifier ensemble and its performance evaluation, the NSGA-II was employed 100 times for each data set. A number of non-dominated rule sets were simultaneously obtained from each run of the NSGA-II. Among those non-dominated rule sets, too small rule sets were excluded from the classifier ensemble. More specifically, we used the number of classes as the lower bound on the number of fuzzy rules. That is, we excluded non-dominated rule sets with less than M fuzzy rules for an M-class classification problem. After the classifier ensemble was designed, each pattern was independently classified by each individual non-dominated rule set in the ensemble. Then the majority class was chosen as the final classification result of that pattern by the classifier ensemble.

Our computational experiments in this paper were performed in the same manner as in our former study [16] where the performance of individual non-dominate rule sets was examined. Here we briefly describe the experimental conditions (for details, see [16]). All attribute values of each data set 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 generated from four fuzzy partitions with different granularities in Fig. 1. We generated 300 fuzzy rules for each class as candidate rules in a greedy manner using the SLAVE criterion. Thus the total number of candidate rules was 300M where M is the number of classes. The upper bound on the length of candidate rules was two for the Sonar data set and three for the other data sets.

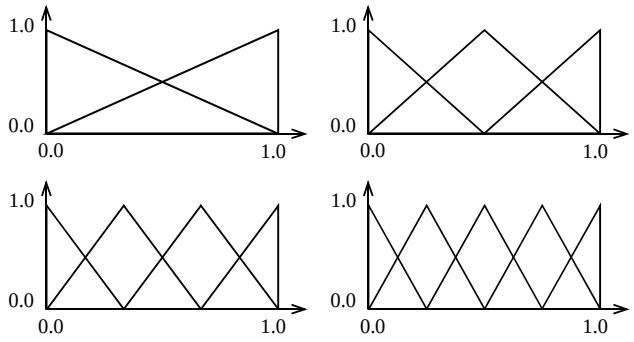


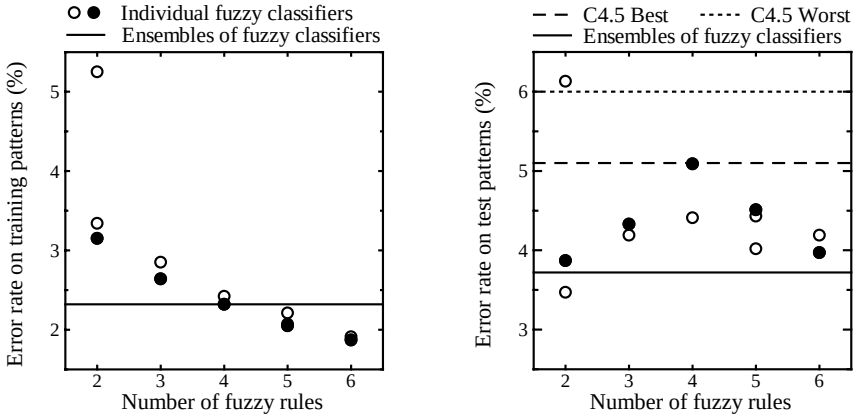
Fig. 1. Four fuzzy partitions used in our computer simulations.

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

4.3 Experimental Results

Wisconsin Breast Cancer Data Set: Average error rates by our classifier ensembles on training patterns and test patterns are shown by the solid lines in Fig. 2 (a) and Fig. 2 (b), respectively. They were 2.32% on training patterns in Fig. 2 (a) and 3.75% on test patterns in Fig. 2 (b). Average error rates of individual classifiers (i.e., individual non-dominated rule sets) are shown by closed and open circles in each figure. Closed circles are used for indicating individual classifiers with low error rates on training patterns. As shown in Fig. 2 (and other figures in this paper), individual classifiers with low error rates on training patterns did not always have low error rates on test patterns. This makes it very difficult to choose a single classifier from multiple alternative ones. The aggregation of many classifiers using a voting scheme avoids this difficult task (i.e., the choice of a single classifier). The performance of individual classifiers (i.e., closed and open circles) was examined in our former study [16]. For the Wisconsin breast cancer data set, the best and worst error rates among the six variants of the C4.5 algorithm were reported as 5.1% and 6.0% by Elomaa & Rousu [7], respectively. Those results are shown by the two dotted lines in Fig. 2 (b).

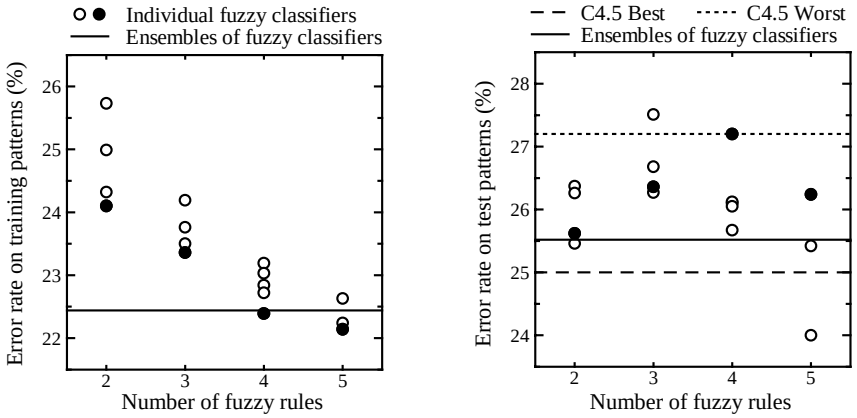


(a) Error rates on training patterns. (b) Error rates on test patterns.

Fig. 2. Experimental results on the Wisconsin breast cancer data set.

From Fig. 2 (b), we can see that the performance of our classifier ensembles was much better than the best result of the C4.5 algorithm in [7]. We can also see that the performance of our classifier ensembles was better than many individual classifiers while it was slightly inferior to the best individual classifier in Fig. 2 (b).

Diabetes Data Set: In the same manner as Fig. 2, experimental results on the diabetes data set are summarized in Fig. 3. As in Fig. 2 (b), we can observe a positive effect of aggregating multiple non-dominated rule sets in Fig. 3 (b). That is, the performance of our classifier ensembles (i.e., 25.5% error rate) was better than many individual classifiers. We can also see that the performance of our classifier ensembles was close to the reported best result in [7] by the C4.5 algorithm (i.e., 25.0% error rate) and much better than the worst result (i.e., 27.2% error rate).

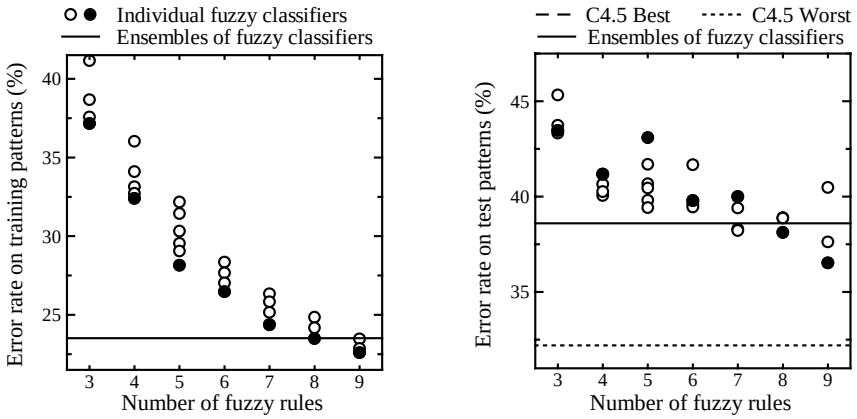


(a) Error rates on training patterns. (b) Error rates on test patterns.

Fig. 3. Experimental results on the diabetes data set.

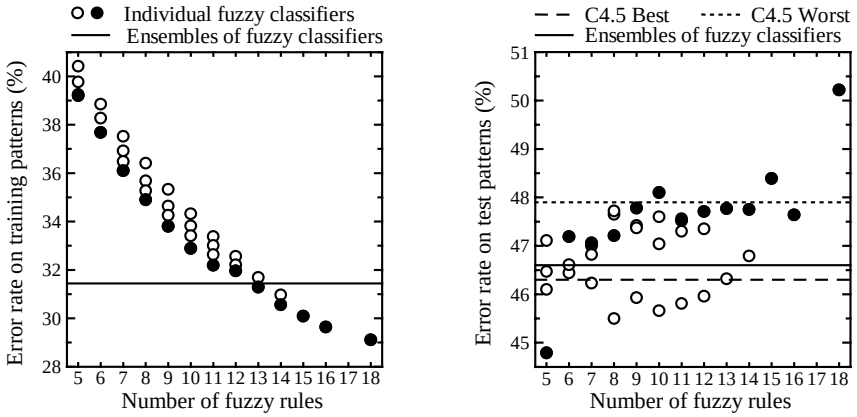
Glass Data Set: Experimental results on the glass data set are summarized in Fig. 4. Since the performance of our classifier ensembles was significantly inferior to the reported results of the C4.5 algorithm in [7], the best result (27.3%) of the C4.5 algorithm is not shown in Fig. 4 (b). To the best of our knowledge, good results have not been reported on the glass data set by descriptive fuzzy rules of the form in (1). Thus we feel that descriptive fuzzy rules with homogeneous fuzzy partition are not suitable for the glass data set. Further studies may be required for improving the performance of descriptive fuzzy rules on the glass data set.

Cleveland Heart Disease Data Set: Experimental results on the Cleveland heart disease data set are summarized in Fig. 5. The average error rate of our classifier ensembles on test patterns in Fig. 5 (b) was 46.6%. The best and worst reported results of the C4.5 algorithm in [7] were 46.3% and 47.9%, respectively.



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

Fig. 4. Experimental results on the glass data set.



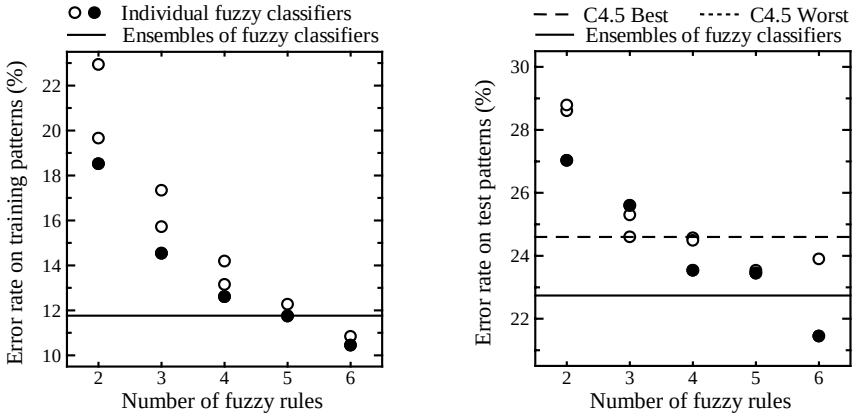
(a) Error rates on training patterns. (b) Error rates on test patterns.

Fig. 5. Experimental results on the Cleveland heart disease data set.

Sonar Data Set: Experimental results on the sonar data set are summarized in Fig. 6. The reported worst result (35.8%) of the C4.5 algorithm in [7] is not shown in Fig. 6 (b) because it is out of the range of the figure. In Fig. 6 (b), the average error rate of our classifier ensembles was 22.74%, which outperformed almost all the individual classifiers and the best result (24.6%) of the C4.5 algorithm in [7].

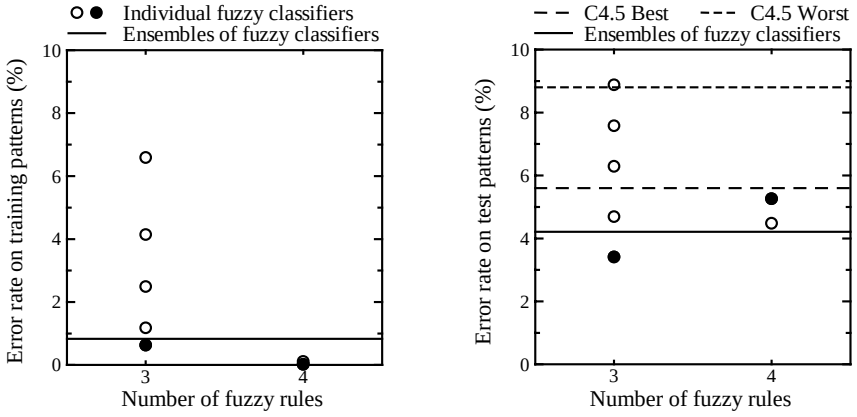
Wine Data Set: Experimental results on the wine data set are summarized in Fig. 7. The best and worst results of the C4.5 algorithm reported in [7] were 5.6% and 8.8%, respectively. The average error rate of our classifier ensembles on test patterns was 4.21%, which was better than the reported best result of the C4.5 algorithm. In

Fig. 7 (b), the performance of our classifier ensembles was better than many individual classifiers while it was slightly inferior to the best individual classifier.



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

Fig. 6. Experimental results on the sonar data set.



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

Fig. 7. Experimental results on the wine recognition data set.

5 Concluding Remarks

We proposed an idea of using EMO algorithms for designing an ensemble of classifiers with high diversity. EMO algorithms seem to be suitable for this task

because a number of classifiers can be simultaneously obtained from their single run. Moreover, many EMO algorithms have some mechanisms for maintaining the diversity of populations (i.e., maintaining the diversity of classifiers). In our computational experiments, we generated a number of non-dominated fuzzy rule-based classifiers by applying the NSGA-II algorithm to the three-objective fuzzy rule selection problem. Of course, other EMO algorithms are applicable to our classifier generation task. Experimental results on six well-known benchmark data sets showed that the performance of classifier ensembles was better than many individual classifiers. It was also shown that the performance of classifier ensembles was comparable with or superior to the reported best results of the C4.5 algorithm in [7] for five benchmark data sets (except for the glass data set).

Our experimental results in this paper suggest that the use of EMO algorithms is a promising approach to the design of classifier ensembles. Moreover, the aggregation of non-dominated classifiers avoids the difficult task of choosing a single classifier from multiple alternatives. As shown in many figures in this paper (and discussed in our former study [16]), low error rates of classifiers on training patterns do not always mean low error rates on test patterns. On the contrary, the minimization of the error rate on training patterns often leads to the deterioration in the error rate on test patterns due to the overfitting to training patterns. Thus it is very difficult to choose a single classifier from multiple alternatives based on their classification performance on training patterns. The proposed idea avoids this difficulty. The proposed idea can be also used as a simple performance measure of EMO algorithms for classification problems because the performance of many non-dominated classifiers can be summarized as an aggregated scalar measure (i.e., the average error rate of their ensemble).

Our experimental results can be further improved in several manners because we used very simple settings for generating classifier ensembles and classifying new patterns. For example, careful selection of classifiers from non-dominated rule sets may improve the performance of classifier ensembles. Adjustment of rule weights and/or membership functions in each individual classifier may also improve the performance of classifier ensembles. We can also use a weighted vote scheme (or other voting schemes [19], [21]) instead of the simple majority vote scheme for improving the performance of ensembles.

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. Bauer, E., and Kohavi, R.: An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants, *Machine Learning* 36 (1999) 105-139.
3. Breiman, L.: Bagging Predictors, *Machine Learning* 24 (1996) 123-140.
4. Cho, S. B., and Kim, J. H.: Combining Multiple Neural Networks by Fuzzy Integral for Robust Classification, *IEEE Trans. on Systems, Man, and Cybernetics* 25 (1995) 380-384.
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. Dietterich, T. G.: An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization, *Machine Learning* 40 (2000) 139-157.
7. Elomaa, T., and Rousu, J.: General and Efficient Multisplitting of Numerical Attributes, *Machine Learning* 36 (1999) 201-244.
8. Freund, Y., and Schapire, R. E.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, *Journal of Computer and System Sciences* 55 (1997) 119-139.
9. Gonzalez, A., and Perez, R.: SLAVE: A Genetic Learning System Based on an Iterative Approach, *IEEE Trans. on Fuzzy Systems* 7 (1999) 176-191.
10. 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.
11. Ishibuchi, H., and Nakashima, T.: Effect of Rule Weights in Fuzzy Rule-Based Classification Systems, *IEEE Trans. on Fuzzy Systems* 9 (2001) 506-515.
12. 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.
13. Ishibuchi, H., Nakashima, T., and Murata, T.: Three-Objective Genetics-Based Machine Learning for Linguistic Rule Extraction, *Information Sciences* 136 (2001) 109-133.
14. 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.
15. 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.
16. Ishibuchi, H., and Yamamoto, T.: Effects of Three-Objective Genetic Rule Selection on the Generalization Ability of Fuzzy Rule-based Systems, *Proc. of 2nd International Conference on Evolutionary Multi-criteria Optimization* (2003) (in press).
17. Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton G. E.: Adaptive Mixtures of Local Experts, *Neural Computation* 3 (1991) 79-87.
18. Jordan, M. I., and Jacobs, R. A.: Hierarchical Mixtures of Experts and the EM Algorithm, *Neural Computation* 6 (1994) 181-214.
19. Kittler, J., Hatef, M., Duin, R. P. W., and Matas, J.: On Combining Classifiers, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 20 (1998) 226-239.
20. Kuncheva, L. I., and Jain, L. C.: Designing Classifier Fusion Systems by Genetic Algorithms, *IEEE Trans. on Evolutionary Computation* 4 (2000) 327-336.
21. Lam, L., and Suen, C. Y.: Optimal Combinations of Pattern Classifiers, *Pattern Recognition Letters* 16 (1995) 945-954.
22. Langdon, W.: A Hybrid Genetic Programming Neural Network Classifier for Use in Drug Discovery, *Proc. of 2nd International Conference on Hybrid Intelligence Systems* (2002) 6 (Prenary Presentation).
23. Quinlan, J. R.: *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo (1993).