

Fuzzy Ensemble Design through Multi-Objective Fuzzy Rule Selection

Hisao Ishibuchi and Yusuke Nojima

1-1 Gakuen-cho, Sakai, Osaka 599-8531, Japan
Department of Computer Science and Intelligent Systems
Osaka Prefecture University
{hisaoi,nojima}@cs.osakafu-u.ac.jp

Summary. The main advantage of evolutionary multi-objective optimization (EMO) over classical approaches is that a variety of non-dominated solutions with a wide range of objective values can be simultaneously obtained by a single run of an EMO algorithm. In this chapter, we show how this advantage can be utilized in the design of fuzzy ensemble classifiers. First we explain three objectives in multi-objective formulations of fuzzy rule selection. One is accuracy maximization and the others are complexity minimization. Next we demonstrate that a number of non-dominated rule sets (i.e., fuzzy classifiers) are obtained along the accuracy-complexity tradeoff surface from multi-objective fuzzy rule selection problems. Then we examine the effect of combining multiple non-dominated fuzzy classifiers into a single ensemble classifier. Experimental results clearly show that the combination into ensemble classifiers improves the classification ability of individual fuzzy classifiers for some data sets.

22.1 Introduction

A promising approach to the design of reliable classifiers is to combine multiple classifiers into a single one [2, 10]. Several methods have been proposed for generating multiple classifiers such as bagging [3] and boosting [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 chapter, we examine the use of evolutionary multiobjective optimization (EMO) algorithms for generating an ensemble of fuzzy rule-based classifiers with high diversity.

The main advantage of evolutionary multiobjective optimization (EMO) over classical approaches is that many non-dominated solutions can be simultaneously obtained by a single run of an EMO algorithm. When an EMO algorithm is used in the design of an ensemble classifier, a number of non-dominated classifiers with high diversity can be obtained by its single run. This advantage has already been utilized in some studies on the design of neural

network ensembles. For example, reference [1] formulated a two-objective optimization problem by dividing the given training patterns into two subsets. Mean squared errors (MSEs) of a neural network on the two subsets were used as two objectives to be minimized by an EMO algorithm. An ensemble classifier was constructed by combining non-dominated neural networks with respect to the two objectives. On the other hand, Chandra and Yao [4, 5] used a different formulation in order to increase the diversity of neural networks in a more direct manner. They formulated a two-objective optimization problem using an accuracy measure and a diversity measure. The MSE on all the given training patterns was used as the accuracy measure while the diversity measure was calculated for each neural network as the difference from the other individuals in each population during the execution of an EMO algorithm. An ensemble classifier was constructed by combining non-dominated neural networks on the accuracy-diversity tradeoff surface. Jin et al. [29, 30] used an EMO algorithm to minimize the MSE and a complexity measure. They defined the complexity of a neural network by the number of connections. Non-dominated neural networks on the accuracy-complexity tradeoff surface were used to construct an ensemble classifier. Oliveira et al. [32, 31] defined the complexity of a neural network by the number of input nodes. They used an EMO algorithm to find non-dominated feature subsets on the accuracy-complexity tradeoff surface. An ensemble classifier was constructed using the obtained non-dominated feature subsets.

In the field of fuzzy rule-based systems, the accuracy-complexity tradeoff is often referred to as the interpretability-accuracy tradeoff. This is because high interpretability is the main advantage of fuzzy rule-based systems over other nonlinear systems such as neural networks. EMO algorithms have been used to design fuzzy rule-based systems with high interpretability and high accuracy in some studies (e.g., [7], [27, 28], and [35, 36]). In these studies, a number of non-dominated fuzzy rule-based systems were obtained on the interpretability-accuracy tradeoff surface. The two-objective fuzzy rule selection method [14] was one of the first EMO-based approaches to the interpretability-accuracy tradeoff analysis of fuzzy rule-based systems. A number of non-dominated rule sets were found with respect to the classification accuracy and the number of fuzzy rules. This method was extended to the case of three objectives in [16] by considering the total number of antecedent conditions as an additional complexity measure. The three-objective fuzzy rule selection method was improved by using a state-of-the-art EMO algorithm [21] and a memetic EMO algorithm [18]. The same idea as the EMO-based multiobjective fuzzy rule selection was also used for the design of non-fuzzy rule-based classification systems [18]. In these studies on fuzzy and non-fuzzy rule selection, a data mining technique was used to find promising candidate rules. An idea of constructing fuzzy rule-based ensemble classifiers using EMO algorithms was proposed in [22]. Almost the same three-objective formulation was also used in [35].

In this chapter, we examine the following three formulations of multiobjective fuzzy rule selection for the design of fuzzy rule-based ensemble classifiers:

Problem 1 (P1): Maximize $f_1(S)$ and minimize $f_2(S)$,

Problem 2 (P2): Maximize $f_1(S)$ and minimize $f_3(S)$,

Problem 3 (P3): Maximize $f_1(S)$, minimize $f_2(S)$, and minimize $f_3(S)$,

where

S : A subset of fuzzy rules (i.e., an individual fuzzy rule-based classifier),

$f_1(S)$: The number of correctly classified training patterns by S ,

$f_2(S)$: The number of fuzzy rules in S ,

$f_3(S)$: The total number of antecedent conditions of fuzzy rules in S .

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

An EMO algorithm is applied to the above-mentioned three fuzzy rule selection problems to find non-dominated rule sets with different interpretability-accuracy tradeoffs. We use the NSGA-II algorithm [9], which is one of the most well-known and frequently used EMO algorithms. An ensemble classifier is constructed from the obtained non-dominated rule sets for each rule selection problem. The three rule selection problems are compared with each other in terms of the generalization ability of the constructed ensemble classifiers. When we choose the members of each ensemble classifier, we examine three strategies for member selection: all the obtained non-dominated rule sets, a prespecified number of the best rule sets with respect to the classification accuracy on the training patterns, and non-dominated rule sets satisfying a prespecified minimum requirement for the classification accuracy.

This chapter is organized as follows. First we briefly describe some basic concepts in multiobjective optimization in Section 22.2 where we also describe the NSGA-II algorithm [9]. Next we explain our two-stage fuzzy rule selection method to find a number of non-dominated rule sets with different interpretability-accuracy tradeoffs in Section 22.3. The first stage is heuristic extraction of promising candidate fuzzy rules while the second stage is evolutionary multiobjective rule selection. Then we show experimental results in Section 22.4 where our two-stage fuzzy rule selection method is applied to the above-mentioned three multiobjective fuzzy rule selection problems. The three strategies for ensemble member selection are also examined in Section 22.4. Finally we conclude this chapter in Section 22.5.

22.2 Evolutionary Multiobjective Optimization

22.2.1 Some Basic Concepts in Multiobjective Optimization

In this subsection, we briefly describe some basic concepts in multiobjective optimization. Let us consider the following k -objective maximization problem:

$$\text{Maximize } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})), \quad (22.1)$$

$$\text{subject to } \mathbf{x} \in \mathbf{X}, \quad (22.2)$$

where $\mathbf{f}(\mathbf{x})$ is the objective vector, $f_i(\mathbf{x})$ is the i -th objective to be maximized, \mathbf{x} is the decision vector, and \mathbf{X} is the feasible region in the decision space. When the following conditions are satisfied, a feasible solution $\mathbf{x} \in \mathbf{X}$ is said to be dominated by another feasible solution $\mathbf{y} \in \mathbf{X}$ (i.e., \mathbf{y} dominates \mathbf{x} : \mathbf{y} is better than \mathbf{x}):

$$\forall i, \quad f_i(\mathbf{x}) \leq f_i(\mathbf{y}) \text{ and } \exists j, f_j(\mathbf{x}) < f_j(\mathbf{y}). \quad (22.3)$$

If there is no feasible solution \mathbf{y} that dominates \mathbf{x} , \mathbf{x} is said to be a Pareto-optimal solution of the multiobjective optimization problem. The set of all the Pareto-optimal solutions is referred to as the Pareto-optimal solution set. The image of the Pareto-optimal solution set onto the objective space is the Pareto front. The dominance relation in 22.3 is also used to define non-dominated solutions in a population of solutions. When there is no solution \mathbf{y} in a population that dominates \mathbf{x} , we refer to \mathbf{x} as a non-dominated solution in that population. The concept of non-dominated solutions is used to evaluate solutions in EMO algorithms.

The task of EMO algorithms is to find all the Pareto-optimal solutions. It is, however, impractical to try to find all the Pareto-optimal solutions of a large-scale multiobjective optimization problem. In this case, EMO algorithms try to find a number of well-distributed near Pareto-optimal solutions.

The dominance relation in 22.3 is modified when it is applied to each of the three fuzzy rule selection problems in Section 22.1. For example, the dominance relation is modified for **P3** as follows: A rule set $S_{\mathbf{x}}$ is said to be dominated by another rule set $S_{\mathbf{y}}$ (i.e., $S_{\mathbf{y}}$ dominates $S_{\mathbf{x}}$: $S_{\mathbf{y}}$ is better than $S_{\mathbf{x}}$) when all the following inequalities hold:

$$f_1(S_{\mathbf{x}}) \leq f_1(S_{\mathbf{y}}), \quad f_2(S_{\mathbf{x}}) \geq f_2(S_{\mathbf{y}}), \quad f_3(S_{\mathbf{x}}) \geq f_3(S_{\mathbf{y}}), \quad (22.4)$$

and at least one of the following inequalities holds:

$$f_1(S_{\mathbf{x}}) < f_1(S_{\mathbf{y}}), \quad f_2(S_{\mathbf{x}}) > f_2(S_{\mathbf{y}}), \quad f_3(S_{\mathbf{x}}) > f_3(S_{\mathbf{y}}). \quad (22.5)$$

Roughly speaking, when a rule set $S_{\mathbf{x}}$ has lower classification accuracy and higher complexity than another rule set $S_{\mathbf{y}}$, $S_{\mathbf{x}}$ is said to be dominated by $S_{\mathbf{y}}$ in all the three fuzzy rule selection problems in Section 22.1. The task of EMO algorithms is to find a number of non-dominated rule sets with different interpretability-accuracy tradeoffs, which are Pareto-optimal or near Pareto-optimal solutions of each fuzzy rule selection problem.

22.2.2 NSGA-II Algorithm

The NSGA-II algorithm [9] is one of the most well-known and frequently used EMO algorithms in the literature. We use this algorithm in our two-stage fuzzy rule selection method since it has a number of advantages such as high performance, algorithmic simplicity, and high popularity. An outline of the NSGA-II algorithm is written as follows (for details, see [8, 9]):

Step 1 (Initialization): Generate an initial population with N_{pop} solutions where N_{pop} is the population size.

Step 2 (Creation of Offspring Population): Generate an offspring population by iterating the following procedures N_{pop} times:

(1) Choose a pair of parent solutions from the current population using binary tournament selection. Each solution is evaluated by Pareto ranking and a crowding measure.

(2) Generate an offspring from the selected parent solutions by crossover and mutation.

Step 3 (Generation Update): Combine the current population and the offspring population into a merged one. Then choose the best N_{pop} solutions from the merged population to construct the next population. Each solution is evaluated by Pareto ranking and a crowding measure in the same manner as the selection phase of parent solutions in Step 2.

Step 4 (Termination Test): If a prespecified stopping condition is not satisfied, return to Step 2. Otherwise terminate the execution of the algorithm. In the latter case, we choose all the non-dominated solutions in the merged population in Step 4 as the final solutions.

In Step 2 of the NSGA-II algorithm, each solution in the current population is evaluated in the following manner. First, Rank 1 is assigned to all the non-dominated solutions in the current population. All solutions with Rank 1 are tentatively removed from the current population. Next, Rank 2 is assigned to all the non-dominated solutions in the reduced current population. All solutions with Rank 2 are tentatively removed from the reduced current population. This procedure is iterated until all solutions are tentatively removed from the current population. In this manner, a different rank is assigned to each solution. Solutions with smaller ranks are viewed as being better than those with larger ranks. Among solutions with the same rank, an additional criterion called a crowding measure is taken into account. The crowding measure for a solution calculates the distance between its adjacent solutions with the same rank in the objective space (for details, see [8, 9]). Less crowded solutions with larger values of the crowding measure are viewed as being better than more crowded solutions with smaller values of the crowding measure. Solutions in the merged population in Step 4 of the NSGA-II algorithm are evaluated in the same manner based on Pareto ranking and the crowding measure.

22.3 Heuristic Rule Extraction and Evolutionary Multiobjective Rule Selection

In this section, we explain how a number of non-dominated fuzzy rule-based classifiers with different interpretability-accuracy tradeoffs can be obtained by our two-stage rule selection method.

22.3.1 Fuzzy Rule-Based Classifiers

Let us assume that we have m training patterns $\mathbf{x}_p = (x_{p1}, \dots, x_{pn})$, $p = 1, 2, \dots, m$ from M classes where x_{pi} is the attribute value of the p -th training pattern for the i -th attribute ($i = 1, 2, \dots, n$). For our n -dimensional M -class pattern classification problem, we use fuzzy rules of the following form:

Rule R_q : If x_1 is A_{q1} and ... and x_n is A_{qn} then Class C_q with CF_q ,
(22.6)

where R_q is the label of the q -th rule, $\mathbf{x} = (x_1, \dots, x_n)$ 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 $\mathbf{A}_q = (A_{q1}, \dots, A_{qn})$ of the fuzzy rule R_q in 22.6 using the product operator as

$$\mu_{\mathbf{A}_q}(\mathbf{x}_p) = \mu_{A_{q1}}(x_{p1}) \cdot \mu_{A_{q2}}(x_{p2}) \cdot \dots \cdot \mu_{A_{qn}}(x_{pn}), \quad p = 1, 2, \dots, m, \quad (22.7)$$

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 rule " $\mathbf{A}_q \Rightarrow \text{Class } h$ " for each class h as follows (see the textbook on fuzzy data mining [17] for fuzzy versions of some basic concepts in data mining such as confidence and support):

$$c(\mathbf{A}_q \Rightarrow \text{Class } h) = \sum_{\mathbf{x}_p \in \text{Class } h} \mu_{\mathbf{A}_q}(\mathbf{x}_p) \bigg/ \sum_{p=1}^m \mu_{\mathbf{A}_q}(\mathbf{x}_p), \quad h = 1, 2, \dots, M. \quad (22.8)$$

The consequent class C_q is specified as the class with the maximum confidence:

$$c(\mathbf{A}_q \Rightarrow \text{Class } C_q) = \max \{c(\mathbf{A}_q \Rightarrow \text{Class } h) \mid h = 1, 2, \dots, M\}. \quad (22.9)$$

Rule weights have a significant effect on the classification accuracy of a fuzzy rule-based classifier. Several methods have been examined to determine the rule weight of each fuzzy rule in the literature [25] where good results are obtained from the following specification:

$$CF_q = c(\mathbf{A}_q \Rightarrow \text{Class } C_q) - \sum_{\substack{h=1 \\ h \neq C_q}}^M c(\mathbf{A}_q \Rightarrow \text{Class } h). \quad (22.10)$$

We use this definition in this chapter.

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 by the fuzzy rule-based classifier S , a single winner rule R_w is chosen from S as follows:

$$\mu_{\mathbf{A}_w}(\mathbf{x}_p) \cdot CF_w = \max\{\mu_{\mathbf{A}_q}(\mathbf{x}_p) \cdot CF_q \mid R_q \in S\}. \quad (22.11)$$

The input pattern \mathbf{x}_p is assigned to the consequent class C_w of the winner rule R_w . When multiple rules with different consequent classes have the same maximum value in (17.7), the classification of the input pattern \mathbf{x}_p is rejected. The classification of \mathbf{x}_p is also rejected when there is no compatible fuzzy rules with positive compatibility grades for \mathbf{x}_p . In this case, all fuzzy rules have the same maximum value of zero in the right-hand side in (17.7).

In this chapter, we use an ensemble of multiple fuzzy rule-based classifiers to classify input patterns. First an input pattern is classified by each individual fuzzy rule-based classifier using the single winner-based method in (17.7). Then the final classification is performed through the simple majority vote scheme based on the classification result by each individual classifier (see [15]) for various voting methods for fuzzy rule-based classifiers). When multiple classes have the same maximum number of votes, one class is randomly chosen among those classes with the maximum vote.

22.3.2 Heuristic Rule Extraction

Genetic rule selection was proposed for designing fuzzy rule-based classifiers with high accuracy and high comprehensibility in [14, 15] where a scalar fitness function was defined as the weighted sum of the first two objectives of our fuzzy rule selection: to maximize the number of correctly classified training patterns (i.e., to maximize $f_1(S)$) and to minimize the number of fuzzy rules (i.e., to minimize $f_2(S)$). That is, fuzzy rule selection was handled in the framework of single-objective optimization with the following scalar fitness function:

$$\text{fitness}(S) = w_1 \cdot f_1(S) - w_2 \cdot f_2(S), \quad (22.12)$$

where w_1 and w_2 are prespecified positive constants. As we have already mentioned in Section 22.1, the single-objective formulation in (17.8) has been extended to two-objective and three-objective formulations.

One difficulty in the design of fuzzy rule-based classifiers through rule selection is that the number of possible candidate fuzzy rules exponentially increases with the number of input variables. 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 $(K + 1)^n$. In the first stage of our two-stage fuzzy rule selection method, a prespecified number of promising candidate fuzzy rules are generated from those combinations in a heuristic manner using a data mining criterion. That is, the first stage is the heuristic rule extraction phase. In the field of data

mining, rules are often evaluated by two rule evaluation criteria: support and confidence. In the same manner as the fuzzy version of the confidence in (17.4), the support of the fuzzy rule “ $\mathbf{A}_q \Rightarrow \text{Class } h$ ” for each class h is defined as follows:

$$s(\mathbf{A}_q \Rightarrow \text{Class } h) = \frac{1}{m} \sum_{\mathbf{x}_p \in \text{Class } h} \mu_{\mathbf{A}_q}(\mathbf{x}_p). \quad (22.13)$$

Various rule evaluation criteria were examined in [23] where good results were obtained from the following criterion:

$$f_{\text{SLAVE}}(R_q) = s(\mathbf{A}_q \Rightarrow \text{Class } C_q) - \sum_{\substack{h=1 \\ h \neq C_q}}^M s(\mathbf{A}_q \Rightarrow \text{Class } h). \quad (22.14)$$

This is a modified version of a rule evaluation criterion used in an iterative fuzzy GBML (Genetics-Based Machine Learning) algorithm called SLAVE [13].

In the heuristic rule extraction phase (i.e., the first stage) of our two-stage fuzzy rule selection method, a prespecified number of candidate fuzzy rules with the largest values of the SLAVE criterion in (17.10) are found for each class. For designing fuzzy rule-based classifiers with high comprehensibility, only short fuzzy rules are examined as candidate fuzzy rules. This restriction on the rule length is consistent with the third objective (i.e., $f_3(S)$: the total rule length) of the three-objective rule selection problem (i.e., $\mathbf{P3}$ in Section 22.1).

22.3.3 Evolutionary Multiobjective Fuzzy Rule Selection

The second stage of our two-stage method is evolutionary multiobjective fuzzy rule selection where the NSGA-II algorithm is used to find a number of non-dominated rule sets from candidate fuzzy rules extracted in the first stage. Let us assume that N fuzzy rules (i.e., N/M fuzzy rules for each class where M is the number of classes) have already been extracted as candidate fuzzy rules using the SLAVE criterion in the first stage of our two-stage method. A subset S of the N candidate fuzzy rules is handled as an individual in the NSGA-II algorithm in the second stage. Each individual is represented by a binary string of the length N as

$$S = s_1 s_2 \cdots s_N, \quad (22.15)$$

where $s_j = 1$ and $s_j = 0$ mean that the j -th candidate rule is included in S and excluded from S , respectively. From this coding, we can see that the size of the search space in the second stage is 2^N , which depends on the number of candidate fuzzy rules (i.e., N) extracted in the first stage.

Since each individual is represented by a binary string, we can use standard genetic operations in the second stage (i.e., in the NSGA-II algorithm) for multiobjective fuzzy rule selection. We use uniform crossover and bit-flip mutation in our computational experiments.

In order to efficiently find non-dominated rule sets, we use two problem-specific heuristic tricks in the NSGA-II algorithm. One trick is the use of 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 each training pattern is classified by the single winner-based method in 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., $f_1(S)$: the number of correctly classified training patterns). At the same time, the second objective (i.e., $f_2(S)$: the number of fuzzy rules) and the third objective (i.e., $f_3(S)$: 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.

The NSGA-II algorithm is applied to one of the three rule selection problems in Section 22.1. A number of non-dominated rule sets (i.e., non-dominated fuzzy rule-based classifiers) are obtained by its single run. Some (or all) of the obtained non-dominated fuzzy rule-based classifiers are used to construct an ensemble classifier. The performance of the constructed ensemble classifier is examined by the classification accuracy on test patterns. In this manner, the three formulations of multiobjective fuzzy rule selection are compared with each other in terms of the performance of ensemble classifiers.

22.4 Computational Experiments

22.4.1 Data Sets

We used 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/~mllearn/>). Table 22.1 shows the number of attributes, the number of patterns, and the number of classes in each data set. Some data sets include incomplete patterns with missing values. Those patterns were not used in our computational experiments.

In the last two columns of Table 22.1, we show benchmark results on these data sets. They are error rates reported in [11] where six variants of the C4.5 algorithm [33, 34] were examined. The six variants were different from

Table 22.1. Data sets used in our computer simulations.

Data set	No. of attributes	No. of patterns	No. of classes	C4.5	
				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	297*	5	46.3	47.9
Sonar	60	208	2	24.6	35.8
Wine	13	178	3	5.6	8.8

* Incomplete patterns with missing values are not included.

each other in their discretization methods of continuous attributes. 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., 10×10 -CV) in [11]. We used the same performance evaluation procedure in our computational experiments.

22.4.2 Experimental Conditions

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. 22.1. We generated 300 fuzzy rules for each class as candidate rules in a greedy manner using the SLAVE criterion in (17.10). Thus the total number of candidate rules is $300M$ where M is the number of classes. The upper bound on the length of candidate fuzzy rules is two for the sonar data set with 60 attributes and three for the other data sets with 13 or less attributes.

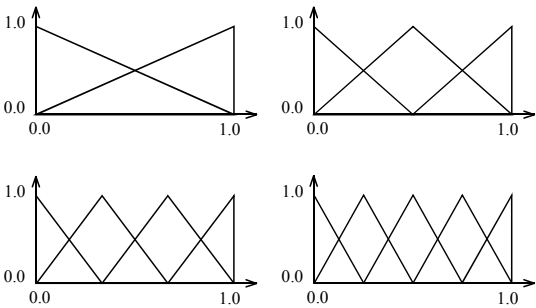


Fig. 22.1. Four fuzzy partitions used in our computer simulations.

The NSGA-II algorithm was employed to find non-dominated rule sets from $300M$ candidate fuzzy rules. We used the following parameter values in the NSGA-II algorithm:

- Population size: 200 strings
- Crossover probability: 0.8
- Biased mutation probabilities: $p_m(0 \rightarrow 1) = 1/300M$ and $p_m(1 \rightarrow 0) = 0.1$
- Stopping condition: 5000 generations

22.4.3 Illustrative Computational Experiments

Before comparing the three formulations of multiobjective fuzzy rule selection by examining the generalization ability of ensemble classifiers on the six data sets by the 10-CV procedure, we demonstrate how our two-stage fuzzy rule selection method works through illustrative computational experiments on the diabetes data set. For illustration purpose, we randomly divided the 768 patterns of this data set into 384 training patterns and 384 test patterns. Using the SLAVE criterion, we generated 600 candidate fuzzy rules (300 rules for each class) from the training patterns. In the heuristic extraction phase of candidate fuzzy rules, we examined fuzzy rules of length 3 or less using “*don’t care*” and 14 antecedent fuzzy sets in Fig. 22.1.

The NSGA-II algorithm was employed for multiobjective fuzzy rule selection from the generated 600 candidate fuzzy rules. We applied the NSGA-II algorithm to **P1** with the two objectives: maximize $f_1(S)$ and minimize $f_2(S)$. After 5000 generations, we obtain 9 non-dominated rule sets (i.e., non-dominated fuzzy rule-based classifiers). It should be noted that these 9 non-dominated rule sets were obtained by a single run of the NSGA-II algorithm. This result clearly demonstrates that EMO algorithms can find a number of non-dominated solutions by their single run. One of the obtained 9 non-dominated rule sets was an empty rule set (i.e., $S = \phi$, $f_1(S) = 0$, $f_2(S) = 0$). Since no patterns are classifiable by an empty rule set, it is always excluded from ensemble classifiers. An ensemble classifier was constructed by combining the other 8 rule sets.

Each pattern was classified by each rule set (i.e., each individual fuzzy rule-based classifier) using the single winner-based fuzzy reasoning method to calculate the error rates on the training patterns and the test patterns. The performance of each rule set is shown in Fig. 22.2 where the rejection of classification is counted as an error (i.e., strictly speaking the vertical axis is the sum of the error rate and the rejection rate). It should be noted that some non-dominated rule sets are overlapping in Fig. 22.2. The second smallest rule set with only a single fuzzy rule is not shown in Fig. 22.2 because its error rates are out of the range of the vertical axis (i.e., 34.0% on the training patterns and 36.8% on the test patterns). Experimental results by the ensemble classifier are shown by the solid line in Fig. 22.2. For comparison, the reported results by the C4.5 algorithm in [11] are also shown in Fig. 22.2 (see Table 22.1).

We can observe a clear tradeoff structure between the error rate on the training patterns and the number of fuzzy rules in Fig. 22.2(a). That is, the interpretability-accuracy tradeoff structure is clearly shown in Fig. 22.2 (a) with respect to the classification accuracy of fuzzy rule-based classifiers on the training patterns. Such a tradeoff structure is not clear in Fig. 22.2(b) with respect to the generalization ability for the test patterns. The generalization ability is somewhat degraded by the increase in the number of fuzzy rules due to the overfitting to the training patterns in Fig. 22.2(b). We can see from Fig. 22.2(b) that the performance of the ensemble classifier is better than almost all the individual fuzzy rule-based classifiers. It should be noted that the computational experiments in this subsection were performed using 50% training patterns and 50% test patterns for illustration purpose. Thus we can not directly compare between the reported results by the C4.5 algorithm based on the 10-CV procedure and our experimental results in Fig. 22.2(b). In the next subsection, we use the 10-CV procedure for the performance evaluation of fuzzy rule-based classifiers and their ensemble classifiers.

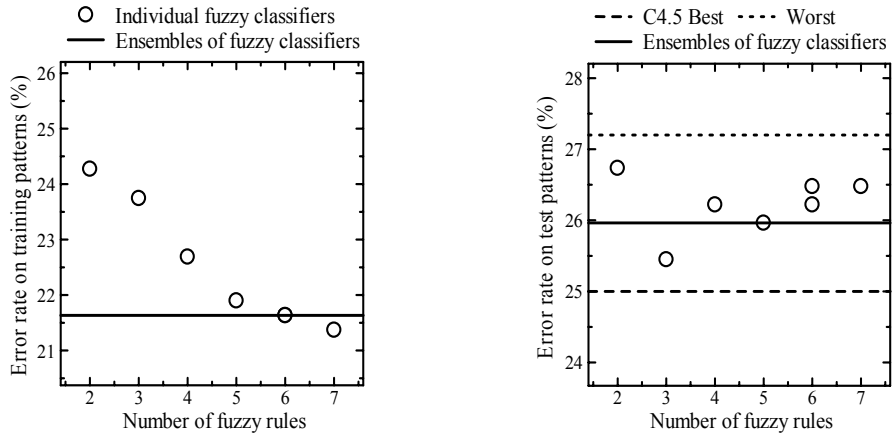


Fig. 22.2. Experimental results of a single run of the NSGA-II algorithm on the diabetes data set. The NSGA-II algorithm was applied to **P1** with 50% training patterns and 50% test patterns. (a) Error rates on training patterns. (b) Error rates on test patterns.

In the same manner as Fig. 22.2, we also applied the NSGA-II algorithm to **P2** with the two objectives: $f_1(S)$ and $f_3(S)$. Experimental results are shown in Fig. 22.3. The horizontal axis in Fig. 22.3 is the total number of antecedent conditions (i.e., $f_3(S)$: the total rule length) while it was the number of fuzzy rules (i.e., $f_2(S)$) in Fig. 22.2. As in Fig. 22.2 (a), we can observe a clear interpretability-accuracy tradeoff structure in Fig. 22.3(a). Such a tradeoff structure is not clear in Fig. 22.3(b) with respect to the accuracy on the test patterns. From the comparison between Fig. 22.2 and Fig. 22.3, we can

see that more non-dominated rule sets were obtained in Fig. 22.3 from **P2** than Fig. 22.2 from **P1**. More specifically, 9 rule sets and 10 rule sets were obtained from **P1** and **P2**, respectively (7 rule sets and 8 rule sets are shown in Fig. 22.2 and Fig. 22.3, respectively). It should be noted again that some rule sets are overlapping in Fig. 22.3. An ensemble classifier was constructed using the obtained 9 non-dominated rule sets (excluding an empty rule set). Error rates by the ensemble classifier are also shown in Fig. 22.3 by the solid line. We can see from Fig. 22.2(b) that the ensemble classifier outperformed all the individual fuzzy rule-based classifiers and the reported results in [11].

The NSGA-II algorithm was also applied to the three-objective fuzzy rule selection problem (i.e., **P3**) in the same manner as Fig. 22.2 and Fig. 22.3. From a single run of the NSGA-II algorithm, 20 non-dominated rule sets were obtained. The classification performance of each rule set is shown in Fig. 22.4 together with that of their ensemble classifier. It should be noted that multiple non-dominated rule sets with the same number of fuzzy rules were obtained in Fig. 22.4. These non-dominated rule sets are different from each other in the total number of antecedent conditions (i.e., the total rule length). More non-dominated rule sets were obtained in Fig. 22.4 from the three-objective formulation than Fig. 22.2 and Fig. 22.3 from the two-objective formulations. In general, the number of obtained non-dominated solutions usually increases with the number of objectives. We can observe a clear interpretability-accuracy tradeoff structure in Fig. 22.4(a) with respect to the classification accuracy of fuzzy rule-based classifiers on the training patterns. This tradeoff structure is not clear in Fig. 22.4(b) with respect to the classification accuracy of fuzzy rule-based classifiers for the test patterns.

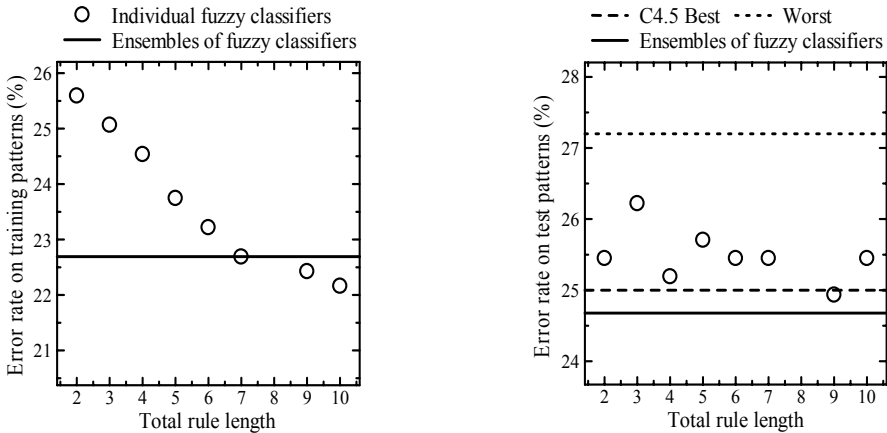


Fig. 22.3. Experimental results on the diabetes data set using **P2**. (a) Error rates on training patterns. (b) Error rates on test patterns.

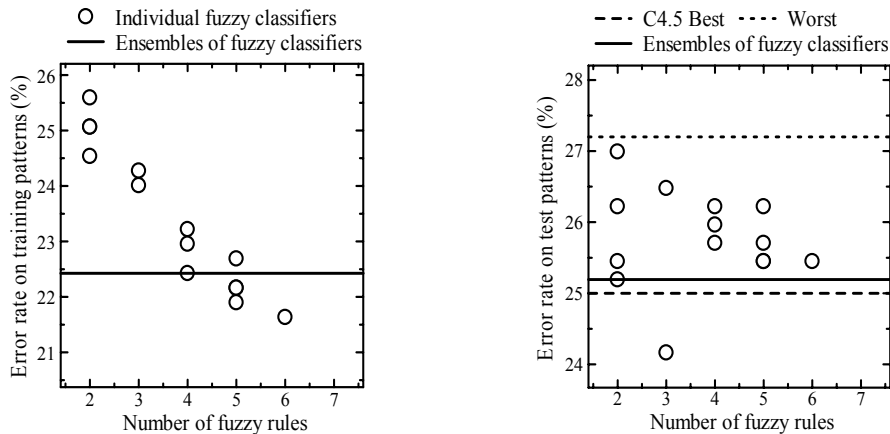


Fig. 22.4. Experimental results on the diabetes data set using **P3**. (a) Error rates on training patterns. (b) Error rates on test patterns.

Examples of non-dominated rule sets with the same number of fuzzy rules are shown in Fig. 22.5 and Fig. 22.6. Fig. 22.5 shows the simplest rule set in Fig. 22.4 with two fuzzy rules, a 25.6% error rate on the training patterns, and a 25.4% error rate on the test patterns. The total number of antecedent conditions (i.e., the total rule length) is 2 in Fig. 22.5. On the other hand, Fig. 22.6 shows the most complicated rule set among those with two fuzzy rules. This rule set corresponds to the left-bottom circle in Fig. 22.4(a) and Fig. 22.4(b), which has a 24.5% error rate on the training patterns and a 25.2% error rate on the test patterns. The total rule length of this rule set is 4 excluding “don’t care” conditions.

	x_2	Consequent
R_1		Class 1 (0.36)
R_2		Class 2 (0.45)

Fig. 22.5. The simplest rule set with two fuzzy rules in Fig. 22.4.





	x_1	x_2	x_8	Consequent
R_1				Class 1 (0.53)
R_2	DC		DC	Class 2 (0.45)

Fig. 22.6. The most complicated rule set with two fuzzy rules in Fig. 22.4.

22.4.4 Comparison of Three Formulations

In this subsection, we compare the three formulations in terms of the classification ability of ensemble classifiers designed by each formulation. The generalization ability of ensemble classifiers was evaluated by ten independent iterations of the whole 10-CV procedure (i.e., 10×10 -CV) in our computational experiments in this subsection. When we constructed ensemble classifiers from non-dominated rule sets, we examined the following three strategies:

(i) The use of all the obtained non-dominated rule sets (excluding an empty rule set).

(ii) The use of a prespecified number of the best non-dominated rule sets in terms of their classification accuracy on training patterns. We examined 10 specifications of the number of classifiers in a single ensemble classifier: 1, 2, ..., 10. When the total number of obtained rule sets was less than the prespecified number, all of them (excluding an empty rule set) were used.

(iii) The use of non-dominated rule sets that have lower error rates on training patterns than a prespecified upper bound. We examined ten specifications of the upper bound on error rates: 5%, 10%, 15%, ..., 50%. In some cases, there were no qualified rule sets that had lower error rates than the prespecified upper bound. In those cases, we could not construct ensemble classifiers. When we could not construct ensemble classifiers more than 50 runs among 100 runs in 10×10 -CV, we do not report the corresponding experimental results.

In the following, we report experimental results using each combination of the three formulations and the three ensemble strategies on each data set.

Wisconsin Breast Cancer Data Set: In Table 22.2, we show the average number of obtained non-dominated rule sets (excluding an empty rule set), the average error rates of those rule sets on training patterns and test patterns. Error rates of ensemble classifiers on test patterns (i.e., generalization ability of ensemble classifiers) are shown in Fig. 22.7. The horizontal axis of Fig. 22.7(a) is the upper bound on the number of rule sets in each ensemble classifier. The right-most label ∞ of the horizontal axis means that there is no upper bound on the number of rule sets (i.e., all the obtained non-dominated rule sets were used in ensemble classifiers). On the other hand, the horizontal axis of Fig. 22.7(b) is the upper bound on error rates on training patterns of rule sets used in ensemble classifiers. The upper bound of 99% (i.e., the right-

Table 22.2. Obtained rule sets by each formulation for the Wisconsin breast cancer data set.

Formulation	No. of rules	Training error (%)	Test error (%)
P1	9.50	6.04	7.84
P2	10.51	5.82	7.40
P3	11.96	6.23	7.84

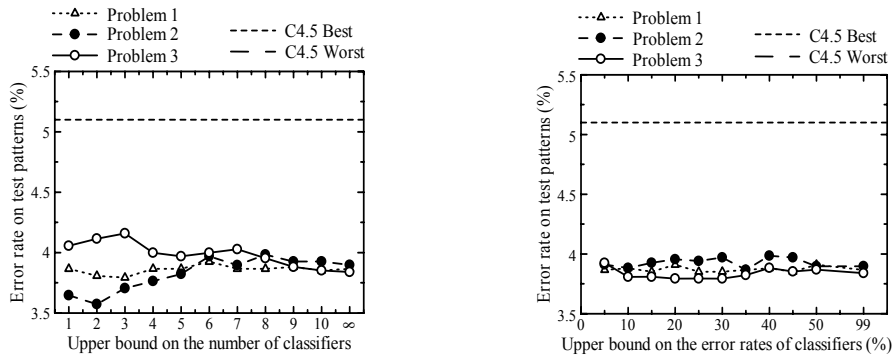


Fig. 22.7. Performance of ensemble classifiers on the Wisconsin breast cancer data set. (a) Second strategy for constructing ensembles. (b) Third strategy for constructing ensembles.

most label of the horizontal axis of Fig. 22.7(b)) means that all the obtained non-dominated rule sets (except for an empty rule set) were used in ensemble classifiers.

We can observe an improvement in error rates by combining more than three fuzzy rule-based classifiers in the case of **P3** in Fig. 22.7(a). We can also see that the performance of individual fuzzy rule-based classifiers and ensemble classifiers in Fig. 22.7 is much better than the best result of the C4.5 algorithm (i.e., the best average error rate: 5.1%) reported in [11].

Diabetes Data Set: In the same manner as the case of the Wisconsin breast cancer data set, we show experimental results on the diabetes data set in Table 22.3 and Fig. 22.8. In Fig. 22.8, the effect of combining multiple fuzzy rule-based classifiers into a single ensemble classifier is not clear. The best results were obtained from **P1** among the three formulations in Fig. 22.8. The performance of individual fuzzy rule-based classifiers and ensemble classifiers in Fig. 22.8 is slightly inferior to the best result of the C4.5 algorithm.

Glass Data Set: Experimental results on the glass data set are shown in Table 22.4 and Fig. 22.9. In Fig. 22.9, the best results were obtained from **P3** among the three formulations. We can observe a slight improvement in error rates by combining multiple fuzzy rule-based classifiers in the case of **P3** in Fig. 22.9(a). The performance of fuzzy rule-based ensemble classifiers in Fig. 22.9 is inferior to even the worst result of the C4.5 algorithm. It may

Table 22.3. Obtained rule sets by each formulation for the diabetes data set.

Formulation	No. of rules	Training error (%)	Test error (%)
P1	8.94	24.03	26.90
P2	13.73	23.93	26.74
P3	16.46	23.90	26.39

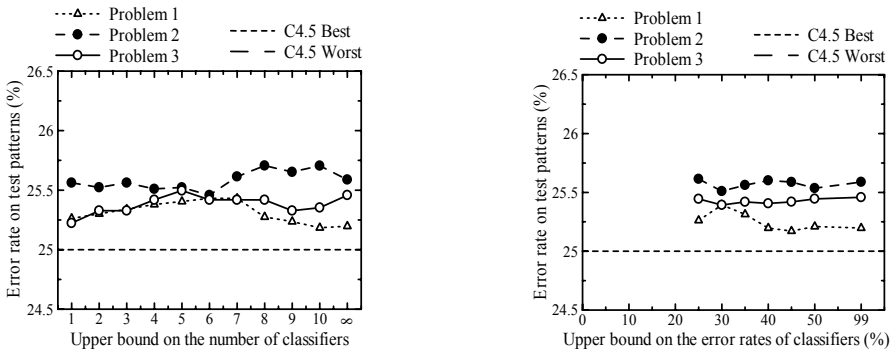


Fig. 22.8. Performance of ensemble classifiers on the diabetes data set. (a) Second strategy for constructing ensembles. (b) Third strategy for constructing ensembles.

Table 22.4. Obtained rule sets by each formulation for the glass data set.

Formulation	No. of rules	Training error (%)	Test error (%)
P1	25.08	26.43	42.19
P2	28.63	32.40	44.89
P3	36.40	33.43	44.50

suggest that the uniform fuzzy partitions of the input space in Fig. 22.1 did not work well on the glass data set. That is, adjustment of antecedent fuzzy sets seems to be necessary to obtain fuzzy rule-based classifiers with high performance on the glass data set.

Cleveland Heart Disease Data Set: Experimental results on the Cleveland heart disease data set are shown in Table 22.5 and Fig. 22.10. In Fig. 22.10, the best results were obtained from **P3** among the three formulations. We can observe a clear positive effect of combining multiple fuzzy rule-based classifiers in Fig. 22.10 for all the three formulations. When an ensemble classifier was constructed from all the non-dominated rule sets obtained from **P3**, its performance was comparable to the best result of the C4.5 algorithm in Fig. 22.10.

Sonar Data Set: Experimental results on the sonar data set are shown in Table 22.6 and Fig. 22.11. The best results were obtained from **P3** in Fig. 22.11. The effect of combining multiple fuzzy rule-based classifiers into a single ensemble classifier is not clear in Fig. 22.11 (a) while it is clear in

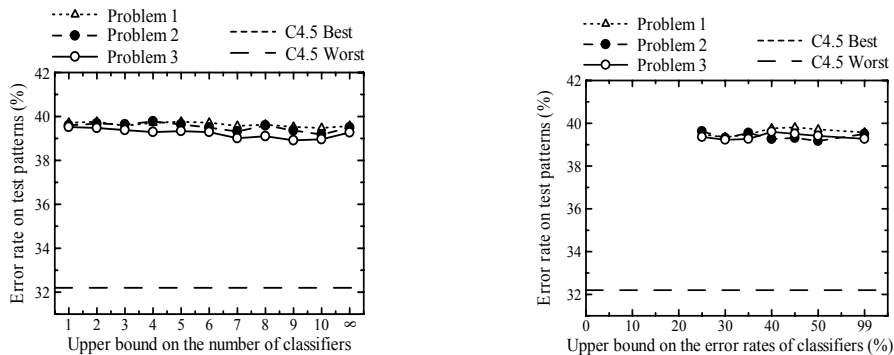


Fig. 22.9. Performance of ensemble classifiers on the glass data set. (a) Second strategy for constructing ensembles. (b) Third strategy for constructing ensembles.

Table 22.5. Obtained rule sets by each formulation for the Cleveland heart disease data set.

Formulation	No. of rules	Training error (%)	Test error (%)
P1	83.33	30.15	47.86
P2	86.50	32.77	47.92
P3	87.04	33.34	47.60

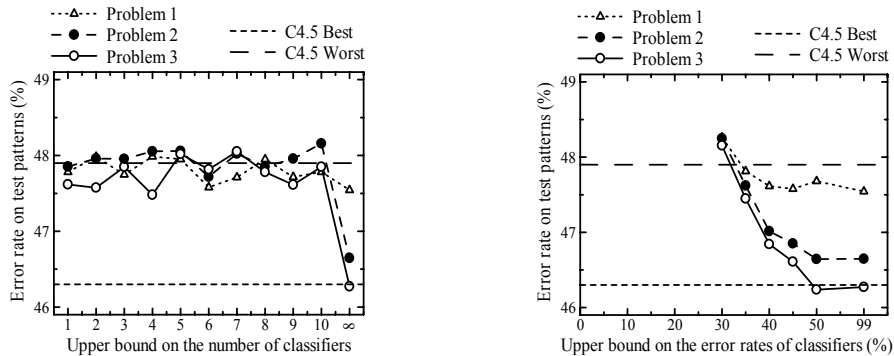


Fig. 22.10. Performance of ensemble classifiers on the Cleveland heart disease data set. (a) Second strategy for constructing ensembles. (b) Third strategy for constructing ensembles.

Fig. 22.11(b). We can also see that the experimental results in Fig. 22.11 by fuzzy rule-based classifiers and ensemble classifiers are better than the best result of the C4.5 algorithm.

Wine Data Set: Experimental results on the wine data set are shown in Table 22.7 and Fig. 22.12. The best results were obtained from **P3** in Fig. 22.12. The improvement in error rates by combining multiple fuzzy rule-based classifiers is clear in Fig. 22.12 in the case of **P3**. Experimental results from **P3** are better than the best result of the C4.5 algorithm (i.e., the best average error rate: 5.6%) in Fig. 22.12.

Table 22.6. Obtained rule sets by each formulation for the sonar data set.

Formulation	No. of rules	Training error (%)	Test error (%)
P1	9.69	16.43	27.66
P2	13.02	16.54	25.96
P3	14.15	17.03	25.83

Table 22.7. Obtained rule sets by each formulation for the wine data set.

Formulation	No. of rules	Training error (%)	Test error (%)
P1	6.23	14.87	19.10
P2	8.49	12.92	17.98
P3	10.94	15.59	18.95

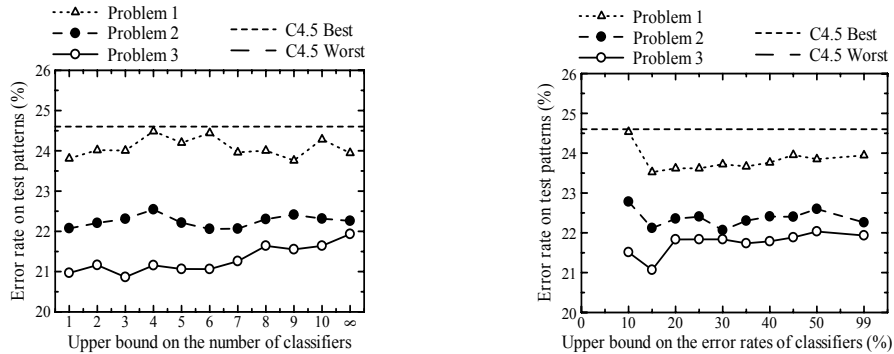


Fig. 22.11. Performance of ensemble classifiers on the sonar data sets. (a) Second strategy for constructing ensembles. (b) Third strategy for constructing ensembles.

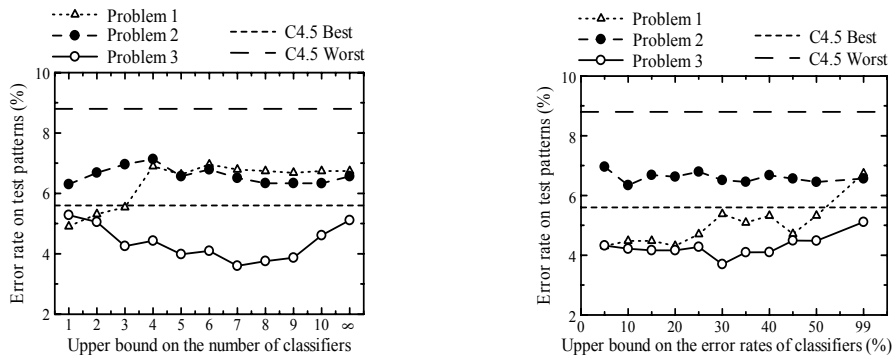


Fig. 22.12. Performance of ensemble classifiers on the wine data set. (a) Second strategy for constructing ensembles. (b) Third strategy for constructing ensembles.

22.4.5 Discussion on Experimental Results

In Table 22.8, we summarized the average number of obtained non-dominated rule sets and the average error rate of their ensemble classifiers with all the obtained non-dominated rule sets on test patterns for each data set. For comparison, we also cite the reported results by the C4.5 algorithm in [11] in the same manner as Table 22.1. The largest average number of obtained non-dominated rule sets and the smallest average error rate for each data set are highlighted by boldface in Table 22.8. In Table 22.8, more non-dominated rule sets were obtained from **P3** with the three objectives than **P1** and **P2** with the two objectives. The use of the total rule length (i.e., $f_3(S)$) in **P2** seems to lead to more non-dominated rule sets than the use of the total number of fuzzy rules (i.e., $f_2(S)$) in **P1**. The best average error rates were obtained from **P3** among the three formulations on average while the difference in average error rates of ensemble classifiers is not large among the three formulations in Table 22.8. We can also see that the performance of ensemble classifiers is comparable to the best results of the C4.5 algorithm except for the case of the glass data set.

In Table 22.9, we compare the generalization ability between the ensemble classifiers of all the obtained non-dominated rule sets and the single best individual classifier (which was chosen from the obtained non-dominated rule sets based on the classification accuracy on training patterns). Table 22.9 shows the average error rates on test patterns calculated by ten independent iterations of the whole 10-CV procedure. Better results between the ensemble classifiers and the single best individual classifier are highlighted by boldface in Table 22.9. From this table, we can see that the effect of combining multiple fuzzy rule-based classifiers depends on data sets and formulations. It improved the generalization ability of fuzzy rule-based classifiers for some data sets while it degraded the generalization ability for other data sets.

Table 22.8. Summary of experimental results.

Data set	Number of non-dominated rules			Error rates of ensembles			C4.5	
	P1	P2	P3	P1	P2	P3	Best	Worst
Breast W	9.50	10.51	11.96	3.87	3.90	3.84	5.1	6.0
Diabetes	8.94	13.73	16.46	25.20	25.59	25.46	25.0	27.2
Glass	25.08	28.63	36.40	39.57	39.50	39.27	27.3	32.2
Heart C	83.33	86.50	87.04	47.54	46.65	46.27	46.3	47.9
Sonar	9.69	13.02	14.15	23.95	22.26	21.93	24.6	35.8
Wine	6.23	8.49	10.94	6.74	6.57	5.11	5.6	8.8
Average	23.80	26.81	29.49	24.48	24.08	23.65	22.32	26.32

Table 22.9. Comparison in the generalization ability between individual and ensemble classifiers. Average error rates were calculated by ten independent iterations of the whole 10-CV procedure.

Data set	P1		P2		P3	
	Single Best	Ensemble	Single Best	Ensemble	Single Best	Ensemble
Breast W	3.87	3.87	3.65	3.90	4.06	3.84
Diabetes	25.26	25.20	25.56	25.59	25.22	25.46
Glass	39.71	39.57	39.59	39.50	39.52	39.27
Heart C	47.78	47.54	47.85	46.65	47.62	46.27
Sonar	23.81	23.95	22.07	22.26	20.97	21.93
Wine	4.91	6.74	6.30	6.57	5.28	5.11
Average	24.22	24.48	24.17	24.08	23.78	23.65

22.5 Concluding Remarks

In this chapter, we first demonstrated that a number of non-dominated rule sets (i.e., non-dominated fuzzy rule-based classifiers) were obtained by a single run of an EMO algorithm for each of the three fuzzy rule selection problems. Then we examined the effect of combining multiple non-dominated fuzzy rule-based classifiers into a single ensemble classifier. Our experimental results showed that the effect of combining multiple non-dominated fuzzy rule-based classifiers depended on data sets and formulations. For some data sets, we observed a clear improvement in average error rates on test patterns by the use of ensemble classifiers. For other data sets, ensemble classifiers did not outperform the single best individual classifier (which was chosen from the obtained non-dominated rule sets based on the classification accuracy on training patterns) independent of the number of classifiers to be combined.

As we have demonstrated in this chapter, non-dominated rule sets obtained by multiobjective fuzzy rule selection had a large diversity with respect

to their classification rates on training patterns, the number of fuzzy rules, and the total rule length. This diversity, however, does not always mean the diversity of fuzzy rules in each rule set. Many non-dominated rule sets may share the same fuzzy rules or the same subset of fuzzy rules. There may not be a large diversity in classification results by non-dominated rule sets obtained by multiobjective fuzzy rule selection. A promising research direction is to combine a diversity-maintenance mechanism of rule sets (or their classification results) into evolutionary multiobjective search as in [4, 5] used for the design of ensemble neural network classifiers. The use of fuzzy genetics-based machine learning (e.g., [26, 35]) instead of fuzzy rule selection is also a promising research direction because no prescreening stage of candidate fuzzy rules is needed.

References

- [1] H.A. Abbass. Pareto neuro-evolution: Constructing ensemble of neural networks using multi-objective optimization. *Proc. of Congress on Evolutionary Computation*, pages 2074–2080, 2003
- [2] E. Bauer, R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36:105–139, 1999
- [3] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996
- [4] A. Chandra, X. Yao. DIVACE: Diverse and accurate ensemble learning algorithm. *Lecture Notes in Computer Science 3177: Intelligent Data Engineering and Automated Learning - IDEAL 2004*. Springer, Berlin, pp 619–625, 2005
- [5] A. Chandra, X. Yao. Evolutionary framework for the construction of diverse hybrid ensemble. *Proc. of the 13th European Symposium on Artificial Neural Networks - ESANN 2005*, pp 253–258, 2005
- [6] O. Cordon, F. Herrera, F. Hoffman, L. Magdalena. *Genetic Fuzzy Systems*, World Scientific, Singapore, 2001
- [7] O. Cordon, M.J.D. Jesus, F. Herrera, L. Magdalena, P. Villar. A multiobjective genetic learning process for joint feature selection and granularity and contexts learning in fuzzy rule-based classification systems. In: J. Casillas, O. Cordon, F. Herrera, L. Magdalena(eds). *Interpretability Issues in Fuzzy Modeling*. Springer, Berlin, pp 79–99, 2003
- [8] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Chichester, 2001
- [9] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. on Evolutionary Computation*, 6:182–197, 2002
- [10] T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40:139–157, 2000
- [11] T. Elomaa, J. Rousu. General and efficient multisplitting of numerical attributes. *Machine Learning*, 36:201–244, 1999
- [12] Y. Freund, R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997

- [13] A. Gonzalez, R. Perez. SLAVE: A genetic learning system based on an iterative approach. *IEEE Trans. on Fuzzy Systems*, 7:176-191, 1999
- [14] H. Ishibuchi, T. Murata, I.B. Turksen. Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems. *Fuzzy Sets and Systems*, 89: 135-150, 1997
- [15] H. Ishibuchi, T. Nakashima, T. Morisawa. Voting in fuzzy rule-based systems for pattern classification problems. *Fuzzy Sets and Systems*, 103: 223-238, 1999
- [16] H. Ishibuchi, T. Nakashima, T. Murata. Three-objective genetics-based machine learning for linguistic rule extraction. *Information Sciences*, 136: 109-133, 2001
- [17] H. Ishibuchi, T. Nakashima, M. Nii. Classification and modeling with linguistic information granules: Advanced approaches to linguistic data mining. Springer, Berlin, 2004
- [18] H. Ishibuchi, S. Namba. Evolutionary multiobjective knowledge extraction for high-dimensional pattern classification problems. *Lecture Notes in Computer Science 3242: Parallel Problem Solving from Nature - PPSN VIII*. Springer, Berlin, pp 1123-1132, 2004
- [19] H. Ishibuchi, K. Nozaki, N. Yamamoto, H. Tanaka. Construction of fuzzy classification systems with rectangular fuzzy rules using genetic algorithms. *Fuzzy Sets and Systems*, 65: 237-253, 1994
- [20] H. Ishibuchi, K. Nozaki, N. Yamamoto, H. Tanaka. Selecting fuzzy if-then rules for classification problems using genetic algorithms. *IEEE Trans. on Fuzzy Systems*, 3: 260-270, 1995
- [21] H. Ishibuchi, T. Yamamoto. Effects of three-objective genetic rule selection on the generalization ability of fuzzy rule-based systems. *Lecture Notes in Computer Science 2632: Evolutionary Multi-Criterion Optimization - EMO 2003*. Springer, Berlin, pp 608-622, 2003
- [22] H. Ishibuchi, T. Yamamoto. Evolutionary multiobjective optimization for generating an ensemble of fuzzy rule-based classifiers. *Lecture Notes in Computer Science 2723: Genetic and Evolutionary Computation - GECCO 2003*. Springer, Berlin, pp 1077-1088, 2003
- [23] H. Ishibuchi, T. Yamamoto. Comparison of heuristic criteria for fuzzy rule selection in classification problems. *Fuzzy Optimization and Decision Making*, 3: 119-139, 2004
- [24] H. Ishibuchi, T. Yamamoto. Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. *Fuzzy Sets and Systems*, 141: 59-88, 2004
- [25] H. Ishibuchi, T. Yamamoto. Rule weight specification in fuzzy rule-based classification systems. *IEEE Trans. on Fuzzy Systems*, 2005 (in press).
- [26] H. Ishibuchi, T. Yamamoto, T. Nakashima. Hybridization of fuzzy GBML approaches for pattern classification problems. *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics* 35: 359-365, 2005
- [27] F. Jimenez, A.F. Gomez-Skarmeta, G. Sanchez, H. Roubos, R. Babuska. Accurate, transparent and compact fuzzy models for function approximation and dynamic modeling through multi-objective evolutionary optimization. *Lecture Notes in Computer Science 1993: Evolutionary Multi-Criterion Optimization - EMO 2001*. Springer, Berlin, pp 653-667.
- [28] F. Jimenez, A.F. Gomez-Skarmeta, G. Sanchez, H. Roubos, R. Babuska. Accurate, transparent and compact fuzzy models by multi-objective evolutionary

- algorithms. In: Casillas J, Cordon O, Herrera F, Magdalena L (eds) *Interpretability Issues in Fuzzy Modeling*. Springer, Berlin, pp 431-451, 2003
- [29] Y. Jin, T. Okabe, B. Sendhoff. Neural network regularization and ensembling using multi-objective evolutionary algorithms. *Proc. of Congress on Evolutionary Computation - CEC 2004*, pp 1-8, 2004
- [30] Y. Jin, T. Okabe, B. Sendhoff. Evolutionary multi-objective optimization approach to constructing neural network ensembles for regression. In: Coello CAC, Lamont GB (eds) *Applications of Multi-Objective Evolutionary Algorithms*. World Scientific, Singapore, pp 653-673, 2004
- [31] L.S. Oliveira, M. Morita, R. Sabourin, F. Bortolozzi. Multi-objective genetic algorithms to create ensemble of classifiers. *Lecture Notes in Computer Science 3410: Evolutionary Multi-Criterion Optimization - EMO 2005*. Springer, Berlin, pp 592-606, 2005
- [32] L.S. Oliveira, R. Sabourin, F. Bortolozzi, C.Y. Suen. Feature selection for ensembles: A hierarchical multi-objective genetic algorithm approach. *Proc. of 7th International Conference on Document Analysis and Recognition - ICDAR 2003*, pp 676-680, 2003
- [33] J.R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo.
- [34] J.R. Quinlan. Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research*, 4: 77-90, 1996
- [35] H. Wang, S. Kwong, Y. Jin, W. Wei, K.F. Man. Agent-based evolutionary approach for interpretable rule-based knowledge extraction. *IEEE Trans. on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 35: 143-155, 2005
- [36] H. Wang, S. Kwong, Y. Jin, W. Wei, K.F. Man. Multi-objective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction. *Fuzzy Sets and Systems* 149: 149-186, 2005