

# Multiobjective Fuzzy Genetics-Based Machine Learning with a Reject Option

Yusuke Nojima and Hisao Ishibuchi

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

**Abstract**—Classifier design for a classification problem with  $M$  classes can be viewed as finding an optimal partition of its pattern space into  $M$  disjoint subspaces. However, this is not always a good strategy especially when training patterns from different classes are heavily overlapping in the pattern space. A simple but practically useful idea is the use of a reject option. In this case, the pattern space is partitioned into  $(M+1)$  disjoint subspace where the classification of new patterns is rejected in the  $(M+1)$ th subspace. In this paper, we discuss the design of fuzzy rule-based classifiers with a reject option. The rejection subspace is specified by a threshold value for the difference of a kind of matching degrees between the best matching class and the second best matching class. The important research question is how to specify the threshold value. We examine the following two approaches: One is manual specification after designing a fuzzy rule-based classifier, and the other is simultaneous multi-objective optimization of a threshold value and a fuzzy rule-based classifier. In the latter approach, we use three objectives: maximization of the correct classification, and minimization of the rejection and the complexity of the classifier.

**Keywords**—Fuzzy genetics-based machine learning; reject option; evolutionary multiobjective optimization.

## I. INTRODUCTION

Tens of thousands of classifier design methods have been proposed so far. Each method generates a classifier based on the pattern distribution of training data and tries to approximate the decision boundary among different classes in order to correctly classify unseen test data. There always exists a tradeoff between the accuracy and the complexity of the classifiers. In general, a very accurate classifier often has a very complicated structure. Such a complicated structure tends to easily over-fit to the training data and leads to the deterioration of the generalization ability for the test data. On the other hand, a simple classifier can avoid the over-fitting to the training data. It is, however, difficult to obtain an accurate classifier with low complexity. Thus, we should consider a good balance between the accuracy and the complexity for classifier design. Evolutionary multiobjective data mining is one of the representative approaches for this issue [1]-[3]. It can provide various classifiers with different accuracy-complexity tradeoff by its single run.

From the practical point of view, we should consider another issue which is uncertainty of data. Since most real-world data sets include measurement noise and/or sample noise,

the decision boundary among different classes for those data sets is usually unclear. As a result, some patterns around the decision boundary are sometimes correctly classified by chance and vice versa.

In this paper, we explicitly incorporate a reject option into fuzzy rule-based classifier design. Since the patterns around the decision boundary are expected to include some noise, a classifier should reject the classification for those patterns in order to increase the reliability of the classifier. Those rejected patterns should exceptionally be handled (e.g., additional manual inspection). This idea is not new. We can find some pioneer studies by C. K. Chow [4] and [5]. Many extended studies can also be found in the literature (e.g., [6]-[9]).

One important issue for fuzzy classifier design with a reject option is its definition. In this paper, we employ a single winner-based classification method for fuzzy rule-based classifiers [10]. Each pattern is classified by the winner rule which has the highest matching degree with an input pattern. We introduce a threshold on the difference between the best matching class and the second matching class. If the difference is less than the threshold, the classifier rejects the classification. Another important issue is how to specify the threshold and how to optimize the fuzzy classifier.

We consider two approaches for the above issues. The first approach is to design fuzzy classifiers using multiobjective fuzzy genetics-based machine learning (MoFGBML) [11] which is one of the most promising methods in the framework of evolutionary fuzzy systems [12]-[15]. MoFGBML can provide a number of fuzzy classifiers with different tradeoffs between accuracy and complexity by its single run. A reject option is applied to the obtained fuzzy classifiers. The second approach is to optimize the threshold during the optimization of fuzzy classifiers. The basic part of the optimization process is the same as the first approach. The differences are the codifications of individuals and the used objective functions. The threshold is coded together with antecedent conditions of fuzzy rules as an individual. We use three objectives: the maximization of the correct classification (i.e., minimization of the error rate on training data except for rejected patterns), the minimization of the number of rejected patterns, and the minimization of the number of rules in a classifier. Through computational experiments, we demonstrate our approaches can improve the correct classification ability by rejecting classification around the decision boundary.

This paper is organized as follows. Section II explains a fuzzy rule-based classifier and a reject option. Section III explains multiobjective fuzzy genetics-based machine learning and objective functions. Section IV demonstrates the proposed approaches through some computational experiments. Section V concludes this paper.

## II. FUZZY CLASSIFIERS WITH A REJECT OPTION

### A. A Fuzzy Rule-Based Classifier

Let assume we have an  $M$ -class data set with  $m$  training patterns  $\mathbf{x}_p = (x_{p1}, \dots, x_{pn})$ ,  $p = 1, 2, \dots, m$ .  $n$  is the number of attributes. For simplicity, each attribute is normalized by the minimum and maximum values. In this paper, a fuzzy rule-based classifier consists of the following type of rules [10]:

Rule  $R_q$ : If  $x_{p1}$  is  $A_{q1}$  and ... and  $x_{pn}$  is  $A_{qn}$   
then Class  $C_q$  with  $CF_q$ , (1)

where  $R_q$  is a rule label for the  $q$ th rule.  $A_{qi}$ ,  $C_q$ , and  $CF_q$  are an  $i$ th antecedent condition, a consequent class, and a rule weight, respectively. A number of triangular membership functions with different granularities (Fig. 1) are simultaneously used as antecedent conditions together with the “don’t care” condition. The “don’t care” plays a role in ignoring the attribute.

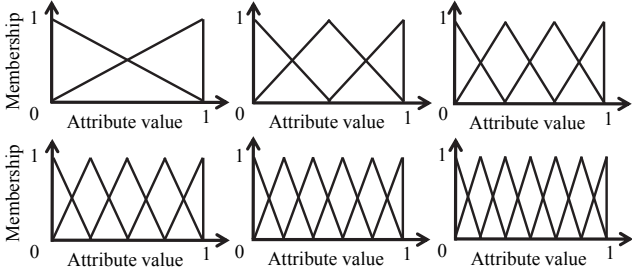


Fig. 1. Fuzzy membership functions with different granularities.

The compatibility grade of  $\mathbf{x}_p$  with the antecedent part  $\mathbf{A}_q$  ( $\mathbf{A}_q = (A_{q1}, \dots, A_{qn})$ ) of each rule  $R_q$  is calculated by the product operation as:

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

The confidence of each class for the antecedent part  $\mathbf{A}_q$  of each rule  $R_q$  is calculated as:

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

Then the consequent class  $C_q$  is specified as the class with the maximum confidence of (3) as:

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

The rule weight  $CF_q$  of each rule  $R_q$  is determined as:

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

As explained above, the consequent part,  $C_q$  and  $CF_q$  of each rule  $R_q$  can be specified in a heuristic manner. Thus, only the combination of antecedent conditions is coded as an individual in a population.

### B. Fuzzy Reasoning

The fuzzy classifier has a number of fuzzy if-then rules explained in the previous subsection. The most appropriate rule for an input pattern is chosen for classification. This is called a single winner-based classification method or a winner-take-all method. When an input pattern  $\mathbf{x}_p$  is classified by a classifier  $S$ , a single winner rule  $R_w$  is chosen from  $S$  as:

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

The input pattern  $\mathbf{x}_p$  is classified as the class  $C_w$  of the winner rule  $R_w$ . When multiple rules with the same maximum value of (6) have different classes, the classification of  $\mathbf{x}_p$  is rejected. If there is no compatible fuzzy rule with  $\mathbf{x}_p$ , its classification is also rejected. We have counted rejected patterns as misclassified patterns in the previous study [11].

### C. Reject Option

We have regarded two cases explained in the last subsection as rejections. One case is multiple rules with the maximum value of (6) have different consequent classes. The other case is that no rule covers a pattern. In this paper, we distinguish between misclassified patterns and rejected patterns. We also extend the definition of the reject option.

When an input pattern  $\mathbf{x}_p$  is classified by  $S$ , one rule with the maximum value of (6) is chosen for each class. Thus, we choose at most  $M$  rules from  $S$  for each pattern. Then, the rule  $R_w$  with the best matching degree and one  $R_v$  with the second best matching degree are chosen. The normalized difference  $\delta$  is calculated as:

$$\delta = \frac{\mu_{\mathbf{A}_w}(\mathbf{x}_p) \cdot CF_w - \mu_{\mathbf{A}_v}(\mathbf{x}_p) \cdot CF_v}{\mu_{\mathbf{A}_w}(\mathbf{x}_p) \cdot CF_w}. \quad (7)$$

If the normalized difference  $\delta$  is less than the threshold  $T_\delta$  (i.e.,  $\delta < T_\delta$ ), the pattern  $\mathbf{x}_p$  is rejected. Thus, the number of rejected patterns strongly depends on the threshold  $T_\delta$ . When the threshold  $T_\delta$  is specified as 0, the classification is the same as the previous one without the reject option.

There are several definitions on threshold-based reject options for fuzzy rule-based classifiers. Some examples can be found in [8].

Next, we explain how the reject option works through a simple example. Let assume that we have a two-class data set with 60 patterns in a two-dimensional decision space shown as in Fig. 2. When we use only three triangular membership functions (i.e., the top-middle membership functions in Fig. 1) per attribute, we obtain the following nine rules.

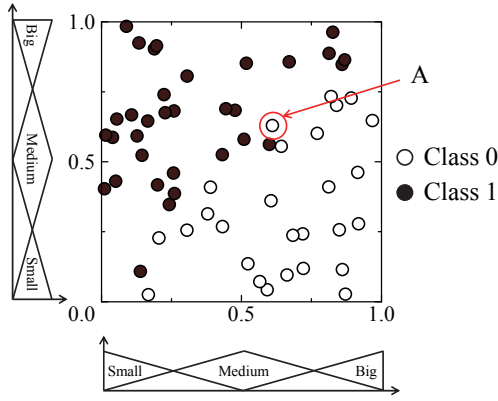


Fig. 2. Example with 60 patterns in a two-dimensional space.

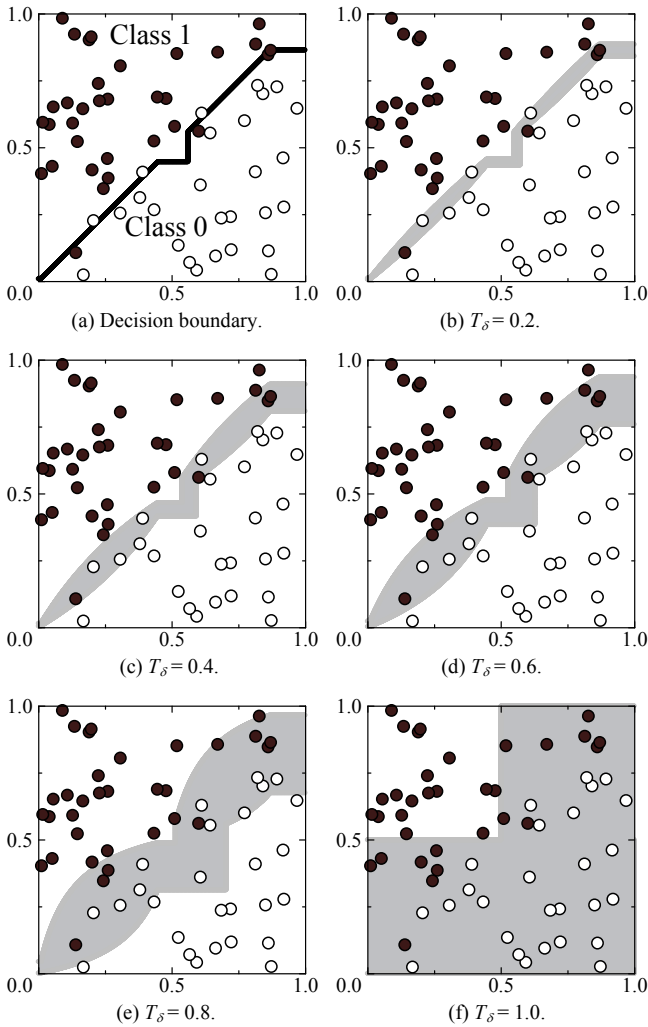


Fig. 3. Decision boundary of the data set in Fig. 2 and the reject regions with different thresholds  $T_\delta$ .

- $R_1$ : If  $x_1$  is *small* and  $x_2$  is *small* then Class 0 with 0.016,  
 $R_2$ : If  $x_1$  is *small* and  $x_2$  is *medium* then Class 1 with 0.828,  
 $R_3$ : If  $x_1$  is *small* and  $x_2$  is *big* then Class 1 with 1.000,

- $R_4$ : If  $x_1$  is *medium* and  $x_2$  is *small* then Class 0 with 0.824,  
 $R_5$ : If  $x_1$  is *medium* and  $x_2$  is *medium* then Class 1 with 0.099,  
 $R_6$ : If  $x_1$  is *medium* and  $x_2$  is *big* then Class 1 with 0.748,  
 $R_7$ : If  $x_1$  is *big* and  $x_2$  is *small* then Class 0 with 1.000,  
 $R_8$ : If  $x_1$  is *big* and  $x_2$  is *medium* then Class 0 with 0.724,  
 $R_9$ : If  $x_1$  is *big* and  $x_2$  is *big* then Class 1 with 0.269.

Let us focus on the pattern A in Fig. 2. The class of the pattern A is 0. Among the above rules, the highest value of (6) is obtained by  $R_6$  (i.e., 0.149). Since the consequent class of  $R_6$  is Class 1, the pattern A is misclassified as Class 1. Among the rules with Class 0 (i.e.,  $R_1$ ,  $R_4$ ,  $R_7$  and  $R_8$ ), the highest value of (6) is obtained by  $R_8$  (i.e., 0.120). Thus, the normalized difference  $\delta$  of (7) is 0.195. When  $T_\delta$  is specified as 0.2, this classifier rejects the classification of the pattern A.

Fig. 3 (a) shows the decision boundary (i.e., black line) using the above rule set (i.e., nine rules). The patterns in the left from the boundary are classified as Class 1. The patterns in the right from the boundary are classified as Class 0. Three patterns with Class 0 and four patterns with Class 1 are misclassified as Class 1 and Class 0, respectively.

In Fig. 3 (b)-(f), the patterns in the gray region are rejected by the proposed approach. When we specify  $T_\delta$  as 0.4, all the patterns outside the reject region are correctly classified. On the other hand, when we specify  $T_\delta$  as 0.6 or more, the reject region seems to be too large. From this observation, we can say that the reject region (i.e., the threshold  $T_\delta$ ) should be optimized.

### III. APPLICATION OF EVOLUTIONARY MULTIOBJECTIVE OPTIMIZATION TO FUZZY CLASSIFIER DESIGN

#### A. Multiobjective Formulations

For fuzzy classifier design with a reject option, we consider two approaches. The first one is a post-processing approach where we use MoFGBML [10] for obtaining fuzzy classifiers without considering a reject option. Then, we specify the threshold  $T_\delta$  on the normalized difference of (7) in order to allow reject classification. In this approach, we use two objective functions like the previous studies:

$$\begin{aligned}
 f_1(S): & \text{Error rates by } S \text{ on all training patterns,} \\
 f_2(S): & \text{Number of rules in a classifier } S, \\
 \text{Minimize } & f_1(S) \text{ and minimize } f_2(S). \tag{8}
 \end{aligned}$$

The second approach is a direct one where first we specify the threshold  $T_\delta$  and then use MoFGBML for obtaining fuzzy classifiers with a reject option. The classifiers are optimized according to the specified threshold  $T_\delta$ . A variant of this direct approach can be considered. In the variant approach, the threshold  $T_\delta$  is also coded as an individual and is optimized together with the combination of the antecedent conditions of each classifier. In this paper, we define two new objectives for the direct approaches.

$$\begin{aligned}
 f_3(S): & \text{Error rates by } S \text{ on training data except for rejected} \\
 & \text{patterns,} \\
 f_4(S): & \text{Number of rejected training patterns by } S.
 \end{aligned}$$

Since we distinguish between misclassified patterns and rejected patterns, the error rate  $f_3(S)$  by  $S$  is calculated by only the number of misclassified patterns except for rejected patterns. If the threshold  $T_\delta$  is 0.0, the value of  $f_3(S)$  is the same as that of  $f_1(S)$ .  $f_4(S)$  should be minimized. The minimization of the number of rejected patterns leads to the reduction of the rejected region.

Since a problem with more than three objectives is regarded as a many-objective problem and is difficult to solve by using dominance-based EMO algorithms [16], we use the following three-objective formulation in this paper:

$$\text{Minimize } f_2(S), \text{ minimize } f_3(S) \text{ and minimize } f_4(S). \quad (9)$$

### B. Multiobjective Fuzzy Genetics-Based Machine Learning

Our MoFGBML is the multiobjective version of hybrid FGBML [17]. The main framework of hybrid FGBML is designed based on the Pittsburgh-approach, where a single classifier is coded as a string (i.e., one individual). A single iteration of the Michigan operation, where a single rule is handled as an individual, is applied as local search to optimize each rule in a classifier. The non-dominated ranking and the crowding distance of NSGA-II [18] are employed in order to obtain a wide range of non-dominated fuzzy classifiers in terms of multiple objectives. The  $(\mu+\mu)$ -ES type generation updating scheme is also employed. The flowchart of our MoFGBML is shown in Fig. 4.

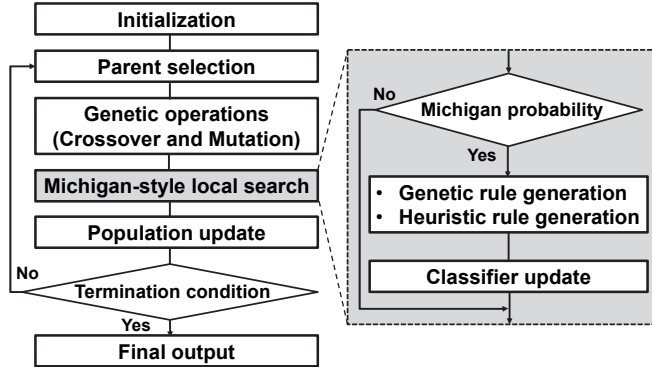


Fig. 4. Hybrid fuzzy genetics-based machine learning.

At the initialization, for each individual,  $N_{\text{rule}}$  base patterns are randomly selected from the training data. This number is the same as the number of initial rules in each individual (i.e., a fuzzy classifier). Then, a prespecified number of support patterns are selected for each base pattern. We employ the recently proposed heuristic rule generation with multiple patterns [19] in order to generate a general rule in an appropriate way. By this operation, “don’t care” conditions are automatically specified according to the base and support patterns. We do not need to specify “don’t care” probability. Instead, we need to specify the number of rules  $H$  for generating a single rule.

In the parent selection, binary tournament selection with replacement is performed based on the non-dominated ranking and the crowding distance. As genetic operations, uniform

crossover and random replacement of antecedent conditions with alternatives are performed to generate offspring.

When the threshold  $T_\delta$  on the normalized difference of (7) is coded in the string (i.e., individual), BLX- $\alpha$  ( $\alpha = 0.5$ ) is used as a crossover operation for the threshold  $T_\delta$ .

After the above operations, a single run of the Michigan-style rule operation is applied with the prespecified probability. In each classifier,  $N_{\text{replace}}$  rules are replaced.  $N_{\text{replace}}$  is specified as  $\lceil 0.2 \times |S| \rceil$ . The procedure is explained as follows:

Step 1: Classify training patterns by  $S$ . The fitness of each fuzzy rule in  $S$  is specified by the product of the number of correctly classified training patterns by that rule and the rule weight  $CF$ . The fitness is penalized by the product of the penalty factor and the number of misclassified patterns.

Step 2: Generate at most half of  $N_{\text{replace}}$  fuzzy rules by heuristic rule generation. Generate the remaining fuzzy rules from the existing rules in  $S$  by genetic operations.

Step 3: Replace the worst  $N_{\text{replace}}$  fuzzy rules in  $S$  with the newly generated  $N_{\text{replace}}$  fuzzy rules.

The heuristic rule generation from multiple patterns in Step 2 is the same as the one in the initialization except for the base pattern selection. It is chosen from the misclassified training patterns and rejected ones. If the number of misclassified patterns is less than  $N_{\text{replace}}/2$ , the genetic operations are performed to generate  $N_{\text{replace}}$  rules in total.

## IV. COMPUTATIONAL EXPERIMENTS

### A. Experimental Setting

To demonstrate the proposed approaches, we use two data sets: Pima data set (768 patterns, 8 attributes, 2 classes) and Heart data set (270 patterns, 13 attributes, 2 classes) available from UCI machine learning repository. Each data set is divided into training data and test data with equal size. The parameter setting is as follows:

- Population size: 100,
- Number of generations: 2,000,
- Number of fuzzy membership functions: 27 (Fig. 1),
- Crossover: Uniform crossover,
- Crossover probability: 0.9,
- Mutation probability:  $1/(n|S|)$  (Pittsburgh),  $1/n$  (Michigan),  
( $n$ : Number of attributes),
- Michigan operation probability: 0.5,
- Number of rules per classifier: 30 (initial), 1 (min), 60 (max),
- Number of patterns  $H$  for heuristic rule generation: 5 patterns,
- Penalty factor in Michigan part: 1.

### B. Results by the Post-Processing Approach

Figs. 5-8 show the results on the Pima data. The black closed circles represent the non-dominated fuzzy classifiers obtained by the two-objective formulation of (8). Fig. 5 shows the error rates except for rejected patterns (i.e.,  $f_3(S)$ ) with the different threshold  $T_\delta$ . We can see a clear tradeoff between the error rate on training data and the number of rules (i.e., closed

black circles). We can also see small improvements on the accuracy by a reject option with the large threshold values ( $T_\delta = 0.8$  or  $1.0$ ).

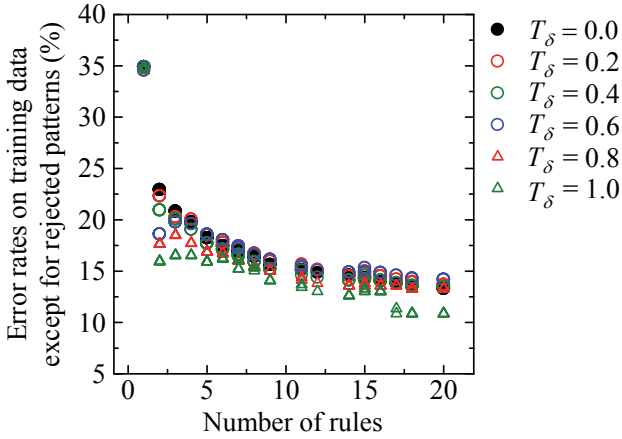


Fig. 5. The results on the Pima data. The error rates on training data and the effects of a reject option with different threshold values ( $T_\delta$ ).

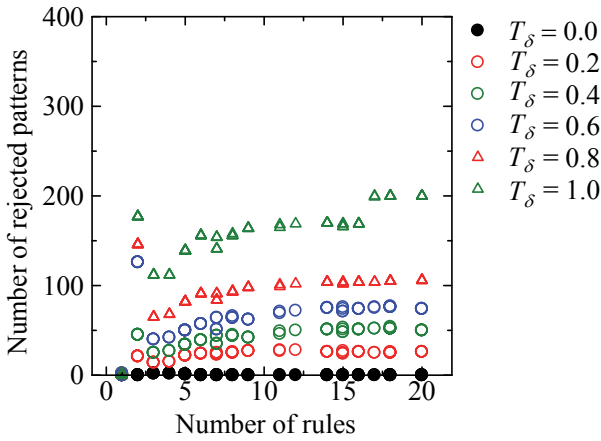


Fig. 6. The results on the Pima data. The number of rejected training patterns by fuzzy classifiers with different threshold values ( $T_\delta$ ).

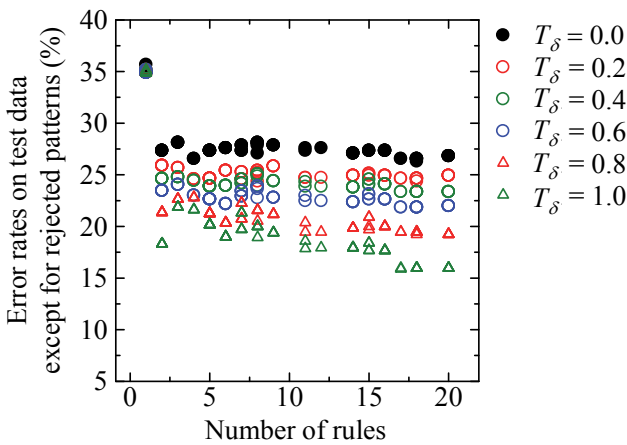


Fig. 7. The results on the Pima data. The error rates on test data and the effects of a reject option with different threshold values ( $T_\delta$ ).

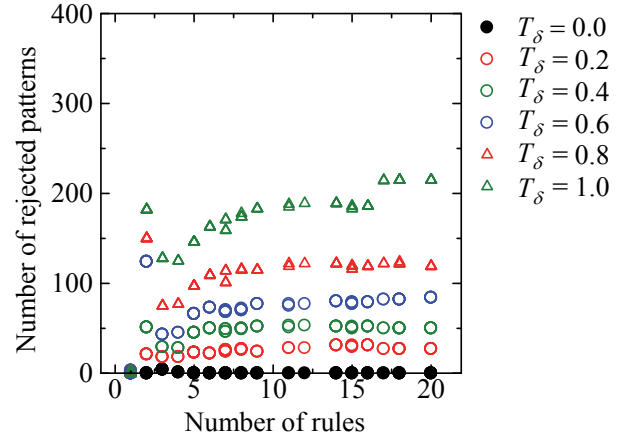


Fig. 8. The results on the Pima data. The number of rejected test patterns by fuzzy classifiers with different threshold values ( $T_\delta$ ).

Fig. 6 shows the number of rejected patterns by fuzzy classifiers with a reject option with different threshold values. The large threshold  $T_\delta$  caused the increase in the number of rejected patterns of the training data. An interesting observation is that fuzzy classifiers with a large number of rules were well affected by the threshold values.

Fig. 7 shows the error rates except for rejected patterns with the different threshold  $T_\delta$ . The error rates of most classifiers with from 2 to 20 rules (i.e., closed black circles) were similar to one another. That is, the classifiers with a large number of rules were over-fitted to the training data. We can also see clear improvements on the error rates except for rejected patterns. The largest improvement was more than 10 %.

On the other hand, the number of rejected test patterns was increased by using the large threshold values  $T_\delta$  (see Fig. 8). If we choose more complicated classifier (i.e., 20 rules) and  $T_\delta = 1.0$ , more than half of the test patterns were rejected for their classifications.

It is almost impossible to specify an appropriate threshold value  $T_\delta$  from the results of Fig. 5. However, we may be able to specify the threshold value  $T_\delta$  from the number of rejected training patterns in Fig. 6, because the similar effects were observed between Fig. 6 and Fig. 8. We can obtain more reliable classifiers only by determining the accepted level on the number of rejected patterns.

Figs. 9-12 show the results on the Heart data. From Fig. 9, no misclassified training pattern was observed by using the reject option with  $T_\delta = 1.0$ . As expected, more than half of the training patterns were rejected (see Fig. 10).

From Fig. 11, we can observe the overfitting phenomena of the obtained classifiers (i.e., black circles). By using the reject option, the overfitting was alleviated. A clear improvement by the reject option was observed for fuzzy classifiers with more than three rules.

As for the Pima data, we can see the similarity of the number of rejected patterns between the training data (Fig. 10) and the test data (Fig. 12). This means that we can predict the



number of rejected test patterns from the result on the training data (Fig. 10). We only need to choose any of classifiers (not too simple ones) and improve its reliability by a reject option.

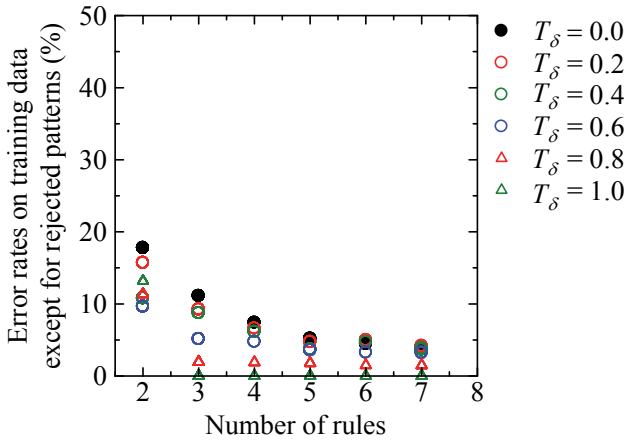


Fig. 9. The results on the Heart data. The error rates on training data and the effects of a reject option with different threshold values ( $T_\delta$ ).

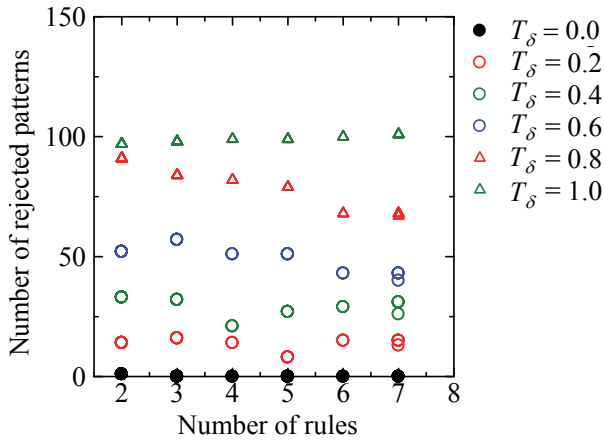


Fig. 10. The results on the Heart data. The number of rejected training patterns by fuzzy classifiers with different threshold values ( $T_\delta$ ).

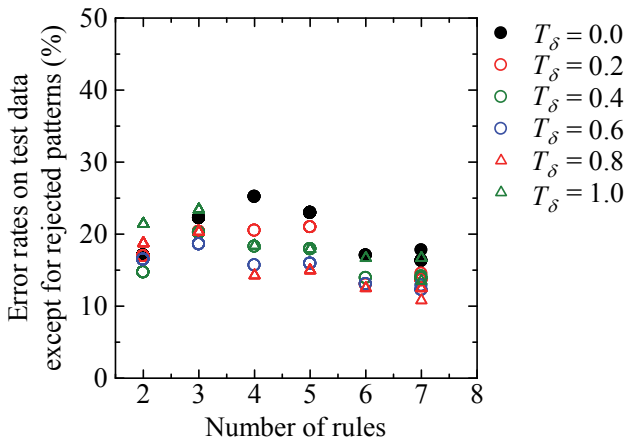


Fig. 11. The results on the Heart data. The error rates on test data and the effects of a reject option with different threshold values ( $T_\delta$ ).

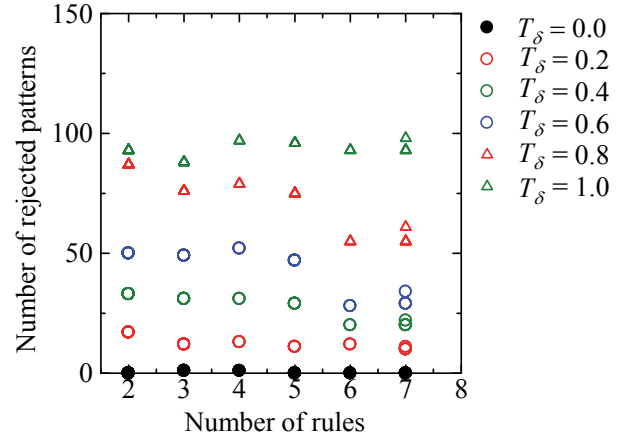


Fig. 12. The results on the Heart data. The number of rejected test patterns by fuzzy classifiers with different threshold values ( $T_\delta$ ).

### C. Results by the Direct Optimization Approaches

Figs. 13-16 show the results on the Pima data by the three-objective formulation of (9). In these figures, the open circles represent the results by using the fixed threshold values for the rejection. The closed circles (i.e., ‘‘Flexible’’) represent the results when the threshold was also coded in an individual.

From Fig. 13, we can see that a large number of non-dominated classifiers were obtained by the three-objective formulation. Some of them were more accurate (i.e., less than 10% error) than the classifiers obtained without the proposed reject option (i.e., black closed circles in Fig. 5).

From Figs. 13 and 14, we can observe, among four settings on the threshold value (i.e.,  $T_\delta = \text{Flexible}$ , 0.1, 0.2, and 0.5), the non-dominated classifiers around the center region of the Pareto front were obtained by the flexible version (see Figs. 13 and 14). The classifiers obtained by the small threshold (i.e.,  $T_\delta = 0.1$ ) rejected a small number of patterns, while those by the large threshold (i.e.,  $T_\delta = 0.5$ ) rejected a large number of patterns.

Regarding the test data accuracy, we can see a wide range of classifiers were obtained by the flexible version with respect to the number of rejected test patterns and the error rates on the test data except for rejected patterns as shown in Fig. 15. The well-improved classifiers (i.e., red and blue circles in Figs. 13 and 14) obtained by the two fixed settings (i.e.,  $T_\delta = 0.1, 0.2$ ) were overfitting to the training data as shown in Figs. 15 and 16. Some of classifiers obtained by the fixed threshold (i.e.,  $T_\delta = 0.5$ ) were very accurate even on the test data. However, those classifiers rejected more than half of the test patterns.

Figs. 16-19 show the results on the Heart data. Similar observations were obtained from the flexible version and the fixed version with  $T_\delta = 0.1$ . Since we do not know an appropriate threshold value, the flexible version would be more useful for the Heart data.

From these results on two data sets, the flexible threshold version seems to be more promising for multiobjective fuzzy classifier design with a reject option.

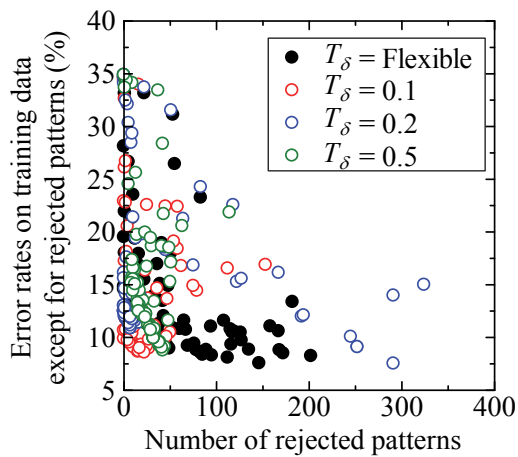


Fig. 13. The results on the Pima data. The error rates on training data and the number of rejected training patterns of the non-dominated classifiers.

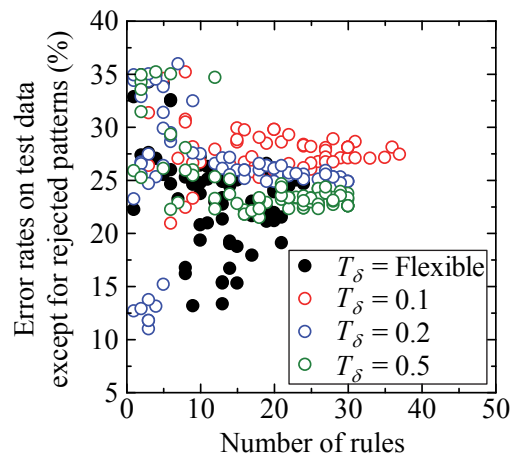


Fig. 16. The results on the Pima data. The error rates on test data and the number of rules of the non-dominated classifiers.

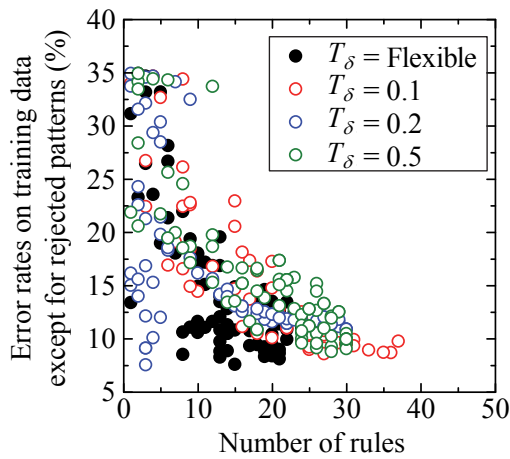


Fig. 14. The results on the Pima data. The error rates on training data and the number of rules of the non-dominated classifiers.

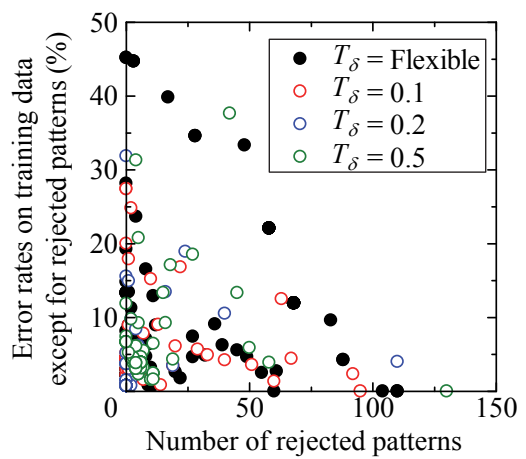


Fig. 16. The results on the Heart data. The error rates on training data and the number of rejected training patterns of the non-dominated classifiers.

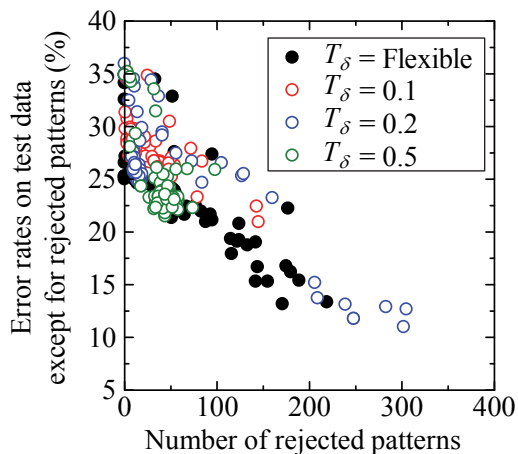


Fig. 15. The results on the Pima data. The error rates on test data and the number of rejected test patterns of the non-dominated classifiers.

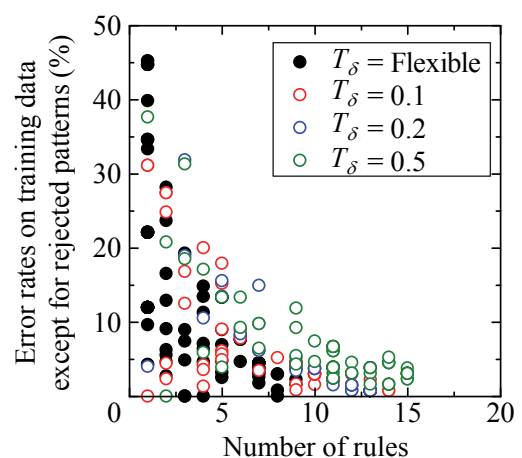


Fig. 17. The results on the Heart data. The error rates on training data and the number of rules of the non-dominated classifiers.

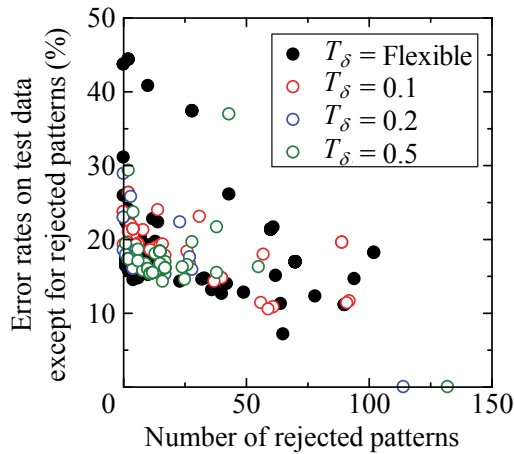


Fig. 18. The results on the Heart data. The error rates on test data and the number of rejected test patterns of the non-dominated classifiers.

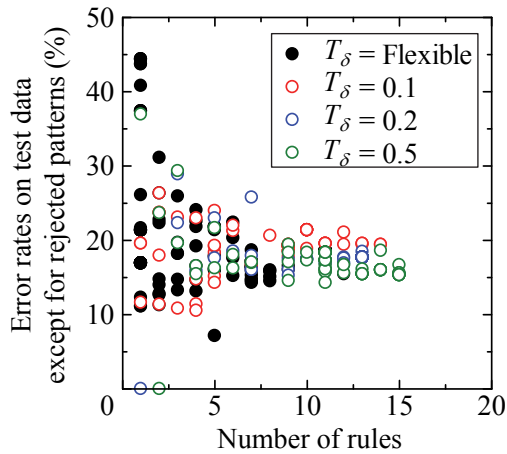


Fig. 19. The results on the Heart data. The error rates on test data and the number of rules of the non-dominated classifiers.

## V. CONCLUSIONS

In this paper, we incorporated a reject option into multiobjective fuzzy classifier design in order to increase the reliability of the obtained fuzzy classifiers. We defined a reject option for a single winner-based classification method where the rejected patterns are specified by a single parameter (i.e., threshold). We proposed two approaches for fuzzy classifier design with a reject option: A post-processing approach and a direct optimization approach. We demonstrated the effect of the proposed approaches through computational experiments in this paper. Experimental results show that better classifiers can be obtained by the flexible version of the direct optimization method with respect to three objectives. The good thing of this method is that we do not need to specify the fixed threshold value for each data. Of course, further experiments with a number of data sets are necessary for deep analysis of the proposed approaches.

As future work, we will incorporate various reject options like [6]-[9] into MoFGBML and compare among them. Data

mining from the rejected patterns would also be an interesting future research topic.

## REFERENCES

- [1] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. Coello Coello, "A survey of multiobjective evolutionary algorithms for data mining: Part I," *IEEE Trans. on Evolutionary Computation*, vol. 18, no. 1, pp. 4-19, 2014.
- [2] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. C. Coello, "Survey of multiobjective evolutionary algorithms for data mining: Part II," *IEEE Trans. on Evolutionary Computation*, vol. 18, no. 1, pp. 20-35, 2014.
- [3] M. Fazzolari, R. Alcalá, Y. Nojima, H. Ishibuchi, and F. Herrera, "A review of the application of multiobjective evolutionary fuzzy systems: Current status and further directions," *IEEE Trans. on Fuzzy Systems*, vol. 21, no. 1, pp. 45-65, February 2013.
- [4] C. K. Chow, "An optimum character recognition system using decision function," *IRE Trans. on Electronic Computers*, vol. 6, no. 4, pp. 247-254, Dec. 1957.
- [5] C. K. Chow, "On optimum recognition error and reject tradeoff," *IEEE Trans. on Information Theory*, vol. 16, no. 1, Jan. 1970.
- [6] B. Dubuisson and M. Masson, "A statistical decision rule with incomplete knowledge about classes," *Pattern Recognition*, vol. 26, no. 1, pp. 155-165, 1993.
- [7] T. M. Ha, "The optimum class-selective rejection rule," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 608-615, June 1997.
- [8] H. Ishibuchi and T. Nakashima, "Fuzzy classification with reject options by fuzzy if-then rules," *Proc. of 7th IEEE International Conference on Fuzzy Systems*, pp. 1452-1457, 1998.
- [9] H. Ishibuchi and M. Nii, "Neural networks for soft decision making," *Fuzzy Sets and Systems*, vol. 115, pp. 121-140, 2000.
- [10] H. Ishibuchi, T. Nakashima, and M. Nii, *Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining*, Springer, Berlin, 2004.
- [11] H. Ishibuchi and Y. Nojima, "Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning," *International Journal of Approximate Reasoning*, vol. 44, no. 1, pp. 4-31, January 2007.
- [12] O. Cordon, F. Herrera, F. Gomide, and F. Hoffman, "Ten years of genetic fuzzy systems: Current framework and new trends," *Fuzzy Sets and Systems*, vol. 14, no. 141, pp. 5-31, January 2004.
- [13] F. Herrera, "Genetic fuzzy systems: Status, critical considerations and future directions," *International Journal of Computational Intelligence Research*, vol. 1, pp. 59-67, December 2005.
- [14] F. Herrera, "Genetic fuzzy systems: Taxonomy, current research trends and prospects," *Evolutionary Intelligence*, vol. 1, pp. 27-46, January 2008.
- [15] A. Fernández, V. López, M. J. del Jesus, and F. Herrera, "Revisiting evolutionary fuzzy systems: Taxonomy, applications, new trends and challenges," *Knowledge-Based Systems*, vol. 80, pp. 109-121, May 2015.
- [16] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," *Proc. of 2008 IEEE Congress on Evolutionary Computation*, pp. 2419-2426, 2008.
- [17] H. Ishibuchi, T. Yamamoto, and T. Nakashima, "Hybridization of fuzzy GBML approaches for pattern classification problems," *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 35, no. 2, pp. 359-365, April 2005.
- [18] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. on Evolutionary Computation*, no. 6, vol. 2, pp. 182-197, April 2002.
- [19] Y. Nojima, K. Watanabe, and H. Ishibuchi, "Effects of heuristic rule generation from multiple patterns in multiobjective fuzzy genetics-based machine learning," *Proc. of 2015 IEEE Congress on Evolutionary Computation*, pp. 2996-3003, 2015.