Fuzzy Optimization and Decision Making, **Vol.3, No.2, pp.119-139, June 2004.**

Final Version of Accepted Paper: Special Issue on Hybrid Approaches for Intelligent Knowledge Management

Comparison of Heuristic Criteria for Fuzzy Rule Selection in Classification Problems

Hisao ISHIBUCHI
hisaoi@ie.osakafu-u.ac.jp
Department of Industrial Engineering, Osaka Prefecture University, Sakai, Osaka 599-8531, Japan

Takashi YAMAMOTO
yama@ie.osakafu-u.ac.jp
Department of Industrial Engineering, Osaka Prefecture University, Sakai, Osaka 599-8531, Japan

Corresponding Authors:

Prof. Hisao Ishibuchi
Department of Industrial Engineering, Osaka Prefecture University
1-1 Gakuen-cho, Sakai, Osaka 599-8531, Japan
E-mail: hisaoi@ie.osakafu-u.ac.jp
Fax: +81-72-254-9915
Phone: +81-72-254-9350

Abstract. This paper compares heuristic criteria used for extracting a pre-specified number of fuzzy classification rules from numerical data. We examine the performance of each heuristic criterion through computational experiments on well-known test problems. Experimental results show that better results are obtained from composite criteria of confidence and support measures than their individual use. It is also shown that genetic algorithm-based rule selection can improve the classification ability of extracted fuzzy rules by searching for good rule combinations. This observation suggests the importance of taking into account the combinatorial effect of fuzzy rules (i.e., the interaction among them).

Keywords: rule extraction, rule selection, fuzzy rules, pattern classification, data mining, genetic algorithm

1. Introduction

In the design of fuzzy rule-based systems, there exist two conflicting objectives: error minimization and comprehensibility maximization. The error minimization has been used in many applications of fuzzy rule-based systems in the literature (e.g., fuzzy control, fuzzy modeling, and fuzzy classification). While the comprehensibility was not usually taken into account in those applications, recently the tradeoff between these two objectives has been discussed in some studies (e.g., see Casillas et al. (2003a, 2003b)).

When fuzzy rule-based systems are used for two-dimensional problems, fuzzy rules can be represented in a tabular form. Figure 1 shows an example of a fuzzy rule table for a two-dimensional pattern classification problem. In this figure, we have four fuzzy rules:

If x_1 is *small* and x_2 is *small* then Class 1, (1)

If x_1 is *small* and x_2 is *large* then Class 2, (2)

If x_1 is *large* and x_2 is *small* then Class 3, (3)

If x_1 is *large* and x_2 is *large* then Class 4, (4)

where *small* and *large* are linguistic values defined by triangular membership functions. As shown in Figure 1, fuzzy rules for two-dimensional problems can be written in a human understandable manner using the tabular form representation. When fuzzy rule-based systems are applied to high-dimensional problems, their comprehensibility is significantly degraded due to the two difficulties: the increase in the number of fuzzy rules and the increase in the number of antecedent conditions of each fuzzy rule.

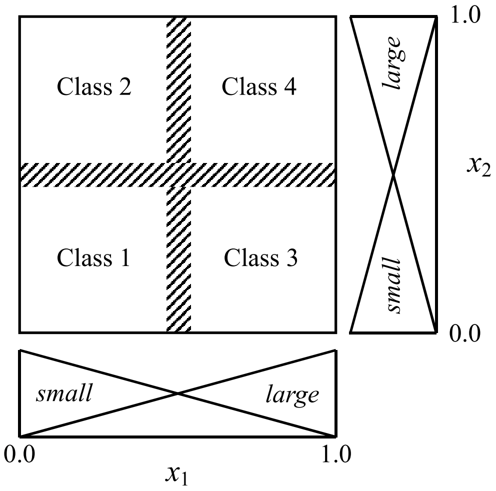

Figure 1. Four fuzzy rules in the two-dimensional pattern space $[0, 1]\times[0, 1]$.

In the field of knowledge discovery and data mining (Fayyad et al. (1996)), emphasis is placed on the comprehensibility of extracted rules. Fuzzy rule-based systems have an inherent advantage with respect to their comprehensibility over other nonlinear systems (e.g., neural networks). This is because fuzzy rules are linguistically interpretable. Such an inherent advantage, however, is significantly degraded due to the above-mentioned two difficulties when fuzzy rule-based systems are applied to high-dimensional problems. For finding comprehensible fuzzy rule-based systems for high-dimensional classification problems, fuzzy rule extraction was formulated as a three-objective optimization problem in Ishibuchi, Nakashima and Murata (2001) where the classification performance was maximized, the number of fuzzy rules was minimized, and the number of antecedent conditions was minimized. Three-objective GBML (genetics-based machine learning) algorithms were used for finding non-dominated rule sets with respect to the three objectives. Because the number of possible fuzzy rules exponentially increases with the number of attributes (i.e., with the dimensionality of problems), the search space for finding good rule sets also exponentially increases. As a result, Pittsburgh-style fuzzy GBML algorithms where an entire rule set is represented by a string require long CPU time and large memory storage in the case of high-dimensional problems. On the other hand, Michigan-style fuzzy GBML algorithms where a single fuzzy rule is represented by a string cannot directly optimize rule sets while they require much less CPU time and memory storage. An alternative approach called iterative fuzzy GBML algorithms has been proposed for efficiently extracting fuzzy rules using heuristic rule selection criteria (e.g., Gonzalez and Perez (1999), Castillo et al. (2001), Castro et al. (2001)). Such an iterative fuzzy GBML algorithm searches for fuzzy rules using a given heuristic rule selection criterion. When a fuzzy rule is found, training patterns covered by that rule are removed from the training data set. Then another fuzzy rule is found using the modified training data set. In this manner, the interaction among fuzzy rules is taken into account in the rule generation process. Iterative fuzzy GBML algorithms require much less memory storage and computation time than Pittsburgh-style algorithms where an entire rule set is represented by a string. On the other hand, iterative algorithms often lead to higher classification performance than Michigan-style algorithms where each fuzzy rule is represented by a string and a population corresponds to a fuzzy rule set. For further discussions on these three classes of fuzzy GBML algorithms, see Cordon et al. (2001).

The aim of this paper is to compare several heuristic rule selection criteria used for fuzzy rule extraction from numerical data. In our computational experiments, we extract a pre-specified number of fuzzy rules using each heuristic criterion. The performance of extracted fuzzy rules is examined on well-known data sets with many continuous attributes available from the UCI ML repository. Experimental results show that better results are obtained from composite criteria of confidence and support measures than their individual use. Experimental results also show that any heuristic criteria do not always generate fuzzy rules with high classification performance when we use a simple greedy method for rule extraction (i.e., when we simply extract a pre-specified number of the best fuzzy rules with respect to a given heuristic criterion without taking into account the combinatorial effect of extracted fuzzy rules). Finally we show that genetic algorithm-based rule selection can improve the classification ability of extracted fuzzy rules. This means that heuristic rule selection criteria can be used as a pre-screening tool of candidate fuzzy rules in genetic algorithm-based rule selection (Ishibuchi and Yamamoto (2002a, 2003a, 2003b)).

2. Fuzzy Rules for Classification Problems

For classification problems with n attributes, we use fuzzy rules of the following form:

$$\text{Rule } R_q : \text{If } x_1 \text{ is } A_{q1} \text{ and } \dots \text{ and } x_n \text{ is } A_{qn} \text{ then Class } C_q \text{ with } CF_q, \tag{5}$$

where R_q is the label of the q-th rule, $\mathbf{x} = (x_1,...,x_n)$ is an n-dimensional pattern vector, A_{qi} is an antecedent fuzzy set (i.e., linguistic value such as *small* and *large* in Figure 1), C_q is a class label, and CF_q is a rule weight. It should be noted that the consequent part of our fuzzy rule for classification problems in (5) is totally different from standard fuzzy rules for function approximation problems. The consequent of our fuzzy rule is a non-fuzzy class label (i.e., Class C_q such as Class 1 and Class 2). Moreover the rule weight CF_q, which is a real number in the unit interval [0, 1], is assigned to each fuzzy rule. The rule weight works as the strength of each fuzzy rule when a new pattern is classified by a set of fuzzy rules (for details, see Ishibuchi & Nakashima (2001)). For other types of fuzzy rules for pattern classification problems, see Cordon et al. (1999).

First we explain how the consequent class C_q and the rule weight CF_q of the fuzzy rule R_q in (5) are specified from numerical data. Let us assume that we have m labeled patterns $\mathbf{x}_p = (x_{p1},...,x_{pn})$, $p = 1,2,...,m$ from M classes (i.e., we have an n-dimensional M-class problem). We define the compatibility grade of each training pattern $\mathbf{x}_p$ with the antecedent part $\mathbf{A}_q = (A_{q1}, ..., A_{qn})$ of the fuzzy rule R_q using the product operator as

$$\mu_{\mathbf{A}_q}(\mathbf{x}_p) = \mu_{A_{q1}}(x_{p1}) \cdot \mu_{A_{q2}}(x_{p2}) \cdot \ \dots \ \cdot \mu_{A_{qn}}(x_{pn}), \tag{6}$$

where $\mu_{A_{qi}}(\cdot)$ is the membership function of the antecedent fuzzy set A_{qi}. The fuzzy conditional probability $\Pr(\text{Class } h \,|\, \mathbf{A}_q)$ of Class h ($h = 1,2,...,M$) for the antecedent part $\mathbf{A}_q$ is numerically approximated as follows (van den Berg et al. (2002)):

$$\Pr(\text{Class } h \mid \mathbf{A}_q) \cong \sum_{\mathbf{x}_p \in \text{Class } h} \mu_{\mathbf{A}_q}(\mathbf{x}_p) \bigg/ \sum_{p=1}^{m} \mu_{\mathbf{A}_q}(\mathbf{x}_p)\,. \tag{7}$$

The right-hand side of (7) is often referred to as the confidence of the fuzzy association rule "$\mathbf{A}_q \Rightarrow \text{Class } h$" in the field of fuzzy data mining (Hong et al. (2001), Ishibuchi, Yamamoto and Nakashima (2001)). This definition of the confidence is a natural extension of its non-fuzzy version (Agrawal and Srikant (1994), Agrawal et al. (1996)). That is, the confidence of the fuzzy association rule "$\mathbf{A}_q \Rightarrow \text{Class } h$" is defined as follows:

$$c(\mathbf{A}_q \Rightarrow \text{Class } h) = \sum_{\mathbf{x}_p \in \text{Class } h} \mu_{\mathbf{A}_q}(\mathbf{x}_p) \bigg/ \sum_{p=1}^{m} \mu_{\mathbf{A}_q}(\mathbf{x}_p)\,. \tag{8}$$

The consequent class C_q of the fuzzy rule R_q is specified by identifying the class with the maximum fuzzy conditional probability (i.e., the maximum confidence). That is, we choose the consequent class C_q so that the following relation holds:

$$c(\mathbf{A}_q \Rightarrow \text{Class } C_q) = \max\{c(\mathbf{A}_q \Rightarrow \text{Class } h) \mid h = 1,2,...,M\}\,. \tag{9}$$

On the other hand, there exist several alternative methods for specifying the rule weight CF_q (Ishibuchi and Yamamoto (2002b)). The choice of an appropriate specification depends on a fuzzy reasoning method used for pattern classification (van den Berg et al. (2002)). The specification of the rule weight of each fuzzy rule has a large effect on the classification performance of fuzzy rule-based systems (Ishibuchi and Nakashima (2001)).

In this paper, we use a single winner-based method (Ishibuchi, Nakashima and Morisawa (1999)). Let S be the set of fuzzy rules in our fuzzy rule-based system. A single winner rule R_w is chosen from the rule set S for an input pattern $\mathbf{x}_p$ as

$$\mu_{\mathbf{A}_w}(\mathbf{x}_p) \cdot CF_w = \max\{\mu_{\mathbf{A}_q}(\mathbf{x}_p) \cdot CF_q \mid R_q \in S\}\,. \tag{10}$$

That is, the winner rule has the maximum product of the compatibility grade and the rule weight in the fuzzy rule-based system. For other fuzzy reasoning methods in fuzzy rule-based classification systems, see Cordon et al. (1999), Ishibuchi, Nakashima and Morisawa (1999), and van den Berg et al. (2002).

When we use the single winner-based method, the following definition of the rule weight CF_q is appropriate for two-class problems (Ishibuchi and Nakashima (2001), Ishibuchi and Yamamoto (2002b)):

$$CF_q = \begin{cases} c(\mathbf{A}_q \Rightarrow \text{Class 1}) - c(\mathbf{A}_q \Rightarrow \text{Class 2}), & \text{if } C_q = \text{Class 1}, \\ c(\mathbf{A}_q \Rightarrow \text{Class 2}) - c(\mathbf{A}_q \Rightarrow \text{Class 1}), & \text{if } C_q = \text{Class 2}. \end{cases} \tag{11}$$

The point is the extension of this formulation to the case of multi-class problems. In this paper, we use the following definition (which is the fourth definition in Ishibuchi and Yamamoto (2002b)) because good results were obtained from this definition in our preliminary computational experiments.

$$CF_q = c(\mathbf{A}_q \Rightarrow \text{Class } C_q) - c(\mathbf{A}_q \Rightarrow \text{Class } \overline{C_q}), \tag{12}$$

where

$$c(\mathbf{A}_q \Rightarrow \text{Class } \overline{C_q}) = \sum_{\substack{h=1 \\ h \neq C_q}}^{M} c(\mathbf{A}_q \Rightarrow \text{Class } h). \tag{13}$$

In this definition, our M-class pattern classification problem is virtually handled as a two-class problem where classification is performed between Class C_q and a merged class including all the other classes (i.e., $\overline{C_q} = \{1, 2, ..., M\} - \{C_q\}$). When CF_q is negative in (12), we do not generate any fuzzy rule with the antecedent part $\mathbf{A}_q$.

Other possible definitions examined in Ishibuchi and Yamamoto (2002b) are as follows:

$$CF_q = c(\mathbf{A}_q \Rightarrow \text{Class } C_q), \tag{14}$$

$$CF_q = c(\mathbf{A}_q \Rightarrow \text{Class } C_q) - c(\mathbf{A}_q \Rightarrow \text{Class } \overline{C_q})/(M-1), \tag{15}$$

$$CF_q = c(\mathbf{A}_q \Rightarrow \text{Class } C_q) - c(\mathbf{A}_q \Rightarrow \text{Class } C_{\text{2nd}}), \tag{16}$$

where Class C_{2nd} is the class with the second largest confidence for the antecedent part $\mathbf{A}_q$. The definition in (15) has been used in many fuzzy rule-based classification systems in our former studies (e.g., Ishibuchi, Nakashima and Murata (1999), and Ishibuchi and Nakashima (1999)) since Ishibuchi et al. (1992). On the other hand, the definition in (16) has been used in some recent studies (e.g., Ishibuchi and Yamamoto (2003b)).

As shown in Ishibuchi and Nakashima (2001), fuzzy rule-based systems can generate various classification boundaries by adjusting the rule weight of each fuzzy rule even when we use fixed membership functions. In Figure 2, we show some examples of classification boundaries generated by the four fuzzy rules in Figure 1 using different rule weights. It should be noted that the membership function of each antecedent fuzzy set in Figure 1 is not modified in Figure 2. A real number in each decision area in Figure 2 shows the rule weight of the corresponding fuzzy rule. As we can see from this figure, classification boundaries are not always parallel to the axes of the pattern space. This is a characteristic feature of fuzzy rule-based classification. For detailed comparison between fuzzy and non-fuzzy rule-based classification, see Ishibuchi and Yamamoto (2002c).

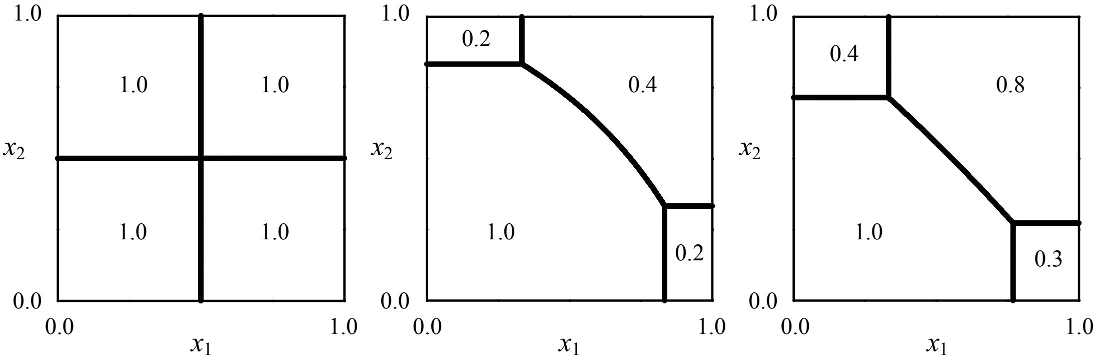

Figure 2. Some examples of classification boundaries generated by the four fuzzy rules in Figure 1.

3. Rule Selection Criteria

In the field of data mining, the confidence and the support of association rules have been often used as rule selection criteria (Agrawal and Srikant (1994), Agrawal et al. (1996)). In (8), we have already shown an extension of the confidence to the case of fuzzy rules. In the same manner, the support is defined for the fuzzy association rule "$\mathbf{A}_q \Rightarrow \text{Class } h$" as follows (Hong et al. (2001), Ishibuchi, Yamamoto and Nakashima (2001)):

$$s(\mathbf{A}_q \Rightarrow \text{Class } h) = \frac{1}{m} \sum_{\mathbf{x}_p \in \text{Class } h} \mu_{\mathbf{A}_q}(\mathbf{x}_p) , \tag{17}$$

where m is the number of given training patterns. In our former studies (Ishibuchi, Yamamoto and Nakashima (2001), Ishibuchi and Yamamoto (2003b)), we used the confidence, the support and their product as rule selection criteria for extracting a pre-specified number of fuzzy rules from numerical data. Experimental results in those studies showed that the product criterion of the confidence and the support outperformed their individual use. In this paper, we examine two composite criteria of the confidence and the support in addition to their product. One is the support criterion with the minimum confidence level where the rule selection is performed using the support criterion from fuzzy rules whose confidence values are larger than or equal to a pre-specified minimum confidence level. The other is the confidence criterion with the minimum support level.

In an iterative fuzzy GBML algorithm called SLAVE in Gonzalez and Perez (1999) and Castillo et al. (2001), a heuristic rule selection criterion was used for extracting fuzzy rules from numerical data. While a somewhat complicated general formulation was shown in those studies, the rule selection criterion in their computational experiments was very simple: $n^{+}(R)-n^{-}(R)$ where $n^{+}(R)$ and $n^{-}(R)$ are the number of positive and negative examples, respectively. This measure can be fuzzified as

$$f_{\text{SLAVE}}(R_q)=n^{+}(R_q)-n^{-}(R_q)=\sum_{\mathbf{x}_p\in\text{Class }C_q}\mu_{\mathbf{A}_q}(\mathbf{x}_p)-\sum_{\mathbf{x}_p\notin\text{Class }C_q}\mu_{\mathbf{A}_q}(\mathbf{x}_p). \quad (18)$$

This formulation can be equivalently rewritten by dividing the right-hand side by m as

$$f_{\text{SLAVE}}(R_q)=s(\mathbf{A}_q\Rightarrow\text{Class }C_q)-s(\mathbf{A}_q\Rightarrow\text{Class }\overline{C_q}), \quad (19)$$

where

$$s(\mathbf{A}_q\Rightarrow\text{Class }\overline{C_q})=\sum_{\substack{h=1\\h\neq C_q}}^{M}s(\mathbf{A}_q\Rightarrow\text{Class }h). \quad (20)$$

On the other hand, the following measure is used in an iterative fuzzy GBML algorithm for the learning of maximal structure fuzzy rules in Castro et al. (2001):

$$f_{\text{Castro}}(R_q)=\frac{n^{+}(R_q)}{|\text{Class }C_q|}\times\frac{|\text{Class }\overline{C_q}|-n^{-}(R_q)}{|\text{Class }\overline{C_q}|}, \quad (21)$$

where $|\text{Class }C_q|$ and $|\text{Class }\overline{C_q}|$ are the number of training patterns in Class C_q and the other classes, respectively. This formulation can be fuzzified and equivalently rewritten by dividing the right-hand side by m^2 as

$$f_{\text{Castro}}(R_q)=\frac{s(\mathbf{A}_q\Rightarrow\text{Class }C_q)}{|\text{Class }C_q|\times|\text{Class }\overline{C_q}|}\times\left(\frac{|\text{Class }\overline{C_q}|}{m}-s(\mathbf{A}_q\Rightarrow\text{Class }\overline{C_q})\right). \quad (22)$$

4. Computational Experiments

In our computational experiments, we used four data sets in Table 1 available from the UCI ML repository. For comparison, some reported results by fuzzy rule-based systems are also included in Table 1 where the average error rates on test data and the average number of fuzzy rules are cited from the literature. We also show some reported results by the C4.5 algorithm (Quinlan (1993)) in the literature. Quinlan (1996) proposed a modified version (Rel 8 in Table 2) of his C4.5 algorithm (Rel 7 in Table 2) for appropriately handling continuous attributes. He evaluated the performance of each version by the 10-CV (ten-fold cross-validation) technique. The whole 10-CV procedure was iterated ten times (i.e., 10×10-CV) using different partitions of each data set into ten subsets in Quinlan (1996). Elomaa and Rousu (1999) proposed an optimal discretization method of continuous attributes into multiple intervals. They examined the performance of six variants of the C4.5 algorithm, which were implemented using three discretization methods (i.e., binary discretization, greedy multisplitting, and optimal multisplitting) and two evaluation functions (i.e., gain ratio and balanced gain). The performance of each variant was examined in Elomaa and Rousu (1999) by ten iterations of the whole 10-CV procedure as in Quinlan (1996). We show in the last two columns in Table 2 the worst and best results among the six variants reported in Elomaa and Rousu (1999) for each data set.

In the Wisconsin breast cancer data, 16 samples with missing values (among 699 samples in total) were not used in our computational experiments. All attribute values of the four data sets were normalized into real numbers in the unit interval [0, 1] before extracting fuzzy rules.

Table 1. Data sets used in this paper and some reported results by fuzzy rule-based systems.

Data set	Number of attributes	Number of samples	Number of classes	Reported results by fuzzy rule-based systems		
				Reference	Error rate	# of rules
Glass	9	214	6	Sanchez et al. (2001)	42.1	8.5
Wisconsin	9	683	2	Sanchez et al. (2001)	4.65	5.1
Wine	13	178	3	Castillo et al. (2001)	3.24	5.2
Sonar	60	208	2	-	-	-

Table 2. Reported error rates by some variants of the C4.5 algorithm.

Data set	Quinlan (1996)		Elomaa & Rousu (1999)	
	Rel 7	Rel 8	Best	Worst
Glass	32.1	32.5	27.3	32.2
Wisconsin	5.29	5.26	5.1	6.0
Wine	-	-	5.6	8.8
Sonar	28.4	25.6	24.6	35.8

Since we did not know an appropriate fuzzy partition for each attribute of each test problem, we simultaneously used four different fuzzy partitions in Figure 3. One of the 14 triangular fuzzy sets was used as an antecedent fuzzy set. For generating simple fuzzy rules (i.e., short fuzzy rules with a small number of antecedent conditions), we also used "*don't care*" as an antecedent fuzzy set. The membership function of "*don't care*" is defined as $\mu_{don't\ care}(x) = 1$ for $\forall x$ because "*don't care*" is compatible with any input values. Since each antecedent fuzzy set may assume one of the 14 triangular fuzzy sets in Figure 3 or "*don't care*", the total number of combinations of antecedent fuzzy sets is 15^n for an n-dimensional problem. Our task is to find a small number of comprehensible fuzzy rules with high classification ability from 15^n possible rules. In our computational experiments on the sonar data with 60 attributes (i.e., $n = 60$), we only examined fuzzy rules with two or less antecedent conditions (i.e., with $(n-2)$ or more "*don't care*" conditions). For the other data sets, we examined fuzzy rules with three or less antecedent conditions. The restriction on the number of antecedent conditions is for finding short (i.e., comprehensible) fuzzy rules as well as for decreasing the CPU time.

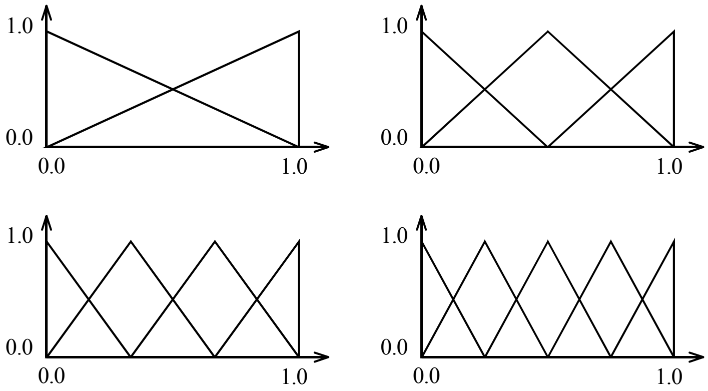

Figure 3. Four fuzzy partitions used in our computational experiments.

We extracted *N* fuzzy rules ($N = 1,2,...$) for each class using one of the seven rule selection criteria described in the previous section: the confidence, the support, their product, the confidence with the minimum support level, the support with the minimum confidence level, the SLAVE criterion, and the Castro criterion. Several values of the minimum support and confidence levels were examined for each data set. The rule extraction was performed for each class in a simple greedy manner. First, the best fuzzy rule with respect to a given rule selection criterion was found for each class. Next, the second best fuzzy rule was found. In this manner, the best *N* fuzzy rules were found for each class. There were many cases where multiple fuzzy rules had the same best value of a given rule selection criterion. In those cases, the tiebreak was performed by applying the following two-step procedure to the multiple fuzzy rules with the same best value of the primary criterion (i.e., one of the seven rule selection criteria). The first tiebreak criterion was the number of antecedent conditions. The fuzzy rule with the least antecedent conditions was chosen. This tiebreak criterion is to favor simpler fuzzy rules. When a single fuzzy rule could not be chosen by the first tiebreak criterion, we used the total area of the antecedent fuzzy sets of each fuzzy rule as the second tiebreak criterion. The total area was calculated by simply summing up the area of the triangular membership function of each antecedent fuzzy set in each fuzzy rule. The fuzzy rule with the largest total area was chosen from the competitive fuzzy rules with the same value of the primary criterion and the same number of antecedent conditions. The second tiebreak criterion is to favor more general fuzzy rules that cover larger subspaces of the pattern space. When multiple fuzzy rules still had the same best evaluation with respect to all the three criteria (i.e., the primary criterion and the two tiebreak criteria), a single rule was randomly selected from those best rules. This two-step tiebreak process was used together with each of the seven primary rule selection criteria. It should be noted that our computational experiments in this section are designed for comparing various rule selection criteria with each other (not for finding optimal fuzzy rule-based classification systems). Thus we use simple experimental settings where a pre-specified number of fuzzy rules are selected for each class. The number of fuzzy rules for each class is adjustable in our genetic algorithm-based rule selection method discussed in Section 5.

In order to evaluate the classification performance on test data (i.e., generalization ability), we used the 10-CV technique for all the four data sets. Since the classification rate estimated by the 10-CV technique usually depends on the data partition into ten subsets, we executed the whole 10-CV procedure five times using different data partitions. Average error rates (including rejection rates) on test data estimated by the 10-CV technique are summarized in Tables 3-6. Note that each error rate in those tables includes the rejection rate (i.e., the error rate was calculated as $(1-r_c)\times 100\,\%$ where r_c is the correct classification rate). On the other hand, average rejection rates are shown in Tables 7-10. Moreover, the average rule length over extracted fuzzy rules is shown in Tables 11-14. It should be noted that the number of actually extracted fuzzy rules was not always the same as the number of fuzzy rules specified in the first column of each table. For example, when we used the minimum support (or confidence) level in the rule extraction, there were several cases where a pre-specified number of fuzzy rules could not be generated for some minority classes because many fuzzy rules did not satisfy the given minimum level. In those cases, the number of actually extracted fuzzy rules was smaller than the specified number in the first column of each table.

From Tables 3-6, we can see that better results were obtained in many cases from the three composite criteria based on both the confidence and the support than their individual use (i.e., the second and third columns). In general, the confidence criterion tends to choose fuzzy rules that cover only a small number of patterns from the same class. Thus the classification of many patterns is likely to be rejected (see Tables 7-10). On the other hand, the support criterion tends to choose fuzzy rules that cover many patterns from multiple classes. Thus some patterns are likely to be misclassified while rejection rates are not high (see Tables 7-10). The point is to find a good balance between these two tendencies by combining the confidence and the support into a single rule selection criterion. The SLAVE and Castro criteria can be viewed as attempts for finding such a good balance.

Table 3. Average error rates including rejection rates on test data of the glass data set.

Number of rules	c	s	Product	c with minimum s-level			s with minimum c-level			SLAVE	Castro
				s: 0.1	s: 0.2	s: 0.3	c: 0.6	c: 0.7	c: 0.8		
6	94.95	51.40	47.85	66.26	73.55	74.81	49.58	42.24	70.28	46.12	48.88
12	94.11	48.46	46.36	59.44	66.82	68.60	48.83	41.12	68.69	45.56	47.90
18	93.36	48.18	47.34	58.27	65.19	67.24	48.08	41.03	67.57	45.98	47.52
24	92.85	47.76	46.92	54.95	62.20	64.21	47.99	40.89	66.73	46.07	48.41
30	92.15	48.04	46.36	54.58	61.68	63.74	47.57	40.84	65.79	46.54	48.08
60	89.63	45.98	46.07	54.49	60.51	62.06	46.54	41.31	63.64	46.17	47.29

Table 4. Average error rates including rejection rates on test data of the Wisconsin breast cancer data set.

Number of rules	c	s	Product	c with minimum s-level			s with minimum c-level			SLAVE	Castro
				s: 0.1	s: 0.2	s: 0.3	c: 0.6	c: 0.7	c: 0.8		
2	90.76	9.78	6.59	51.23	51.42	33.09	9.84	9.08	7.35	6.28	5.70
4	87.07	10.34	7.01	42.46	42.46	23.35	10.28	8.30	8.13	6.85	5.90
6	84.85	8.95	7.07	37.92	37.73	18.67	9.11	5.62	7.09	6.98	4.71
8	83.37	6.18	5.94	33.03	32.88	14.96	5.99	5.46	6.87	6.21	4.58
10	82.46	5.20	5.42	28.57	28.96	12.53	5.20	5.46	6.62	5.21	4.48
60	76.65	4.25	3.97	15.58	16.34	4.44	4.19	4.47	4.55	4.00	4.01

Table 5. Average error rates including rejection rates on test data of the wine data set.

Number of rules	c	s	Product	c with minimum s-level			s with minimum c-level			SLAVE	Castro
				s: 0.1	s: 0.2	s: 0.3	c: 0.6	c: 0.7	c: 0.8		
3	73.60	30.22	11.46	72.70	74.33	74.21	15.39	12.87	10.96	11.35	16.29
6	66.01	22.92	10.84	66.07	66.57	66.46	13.71	11.40	7.25	7.53	12.53
9	52.75	15.11	7.92	35.79	38.88	39.10	14.78	9.72	6.52	7.13	10.11
12	49.89	15.45	6.52	30.17	33.20	33.71	14.49	8.88	6.91	7.25	8.09
15	47.08	15.79	6.57	27.81	25.39	26.18	13.54	7.47	6.69	7.47	7.53
60	44.61	11.18	6.91	5.56	5.84	5.11	6.52	6.69	6.80	6.18	7.13

Table 6. Average error rates including rejection rates on test data of the sonar data set.

Number of rules	*c*	*s*	Product	*c* with minimum *s*-level			*s* with minimum *c*-level			SLAVE	Castro
				s: 0.1	*s*: 0.2	*s*: 0.3	*c*: 0.6	*c*: 0.7	*c*: 0.8		
2	98.37	46.63	46.83	97.84	98.03	98.41	26.68	27.55	43.89	27.55	27.60
4	96.49	46.63	47.69	94.38	96.59	96.54	26.78	27.84	38.99	27.40	27.50
6	95.10	46.78	47.45	78.89	93.46	92.50	27.50	27.55	36.59	26.78	27.50
8	94.52	46.78	47.12	68.94	88.89	88.56	27.98	26.54	34.76	26.39	27.79
10	94.09	47.07	45.48	64.47	85.05	83.99	27.98	26.35	33.85	26.35	27.40
60	87.55	46.35	43.03	51.68	47.84	48.17	27.16	25.19	26.35	23.75	25.10

Table 7. Average rejection rates on test data of the glass data set.

Number of rules	*c*	*s*	Product	*c* with minimum *s*-level			*s* with minimum *c*-level			SLAVE	Castro
				s: 0.1	*s*: 0.2	*s*: 0.3	*c*: 0.6	*c*: 0.7	*c*: 0.8		
6	91.87	0.56	1.40	27.57	51.22	56.45	1.96	7.06	45.65	4.95	0.42
12	89.44	0.42	1.26	17.80	37.06	47.48	1.21	5.37	42.52	3.74	0.28
18	87.94	0.28	1.21	15.28	29.02	43.74	0.79	4.72	40.33	2.90	0.23
24	86.64	0.23	1.07	10.89	25.09	38.93	0.70	3.88	38.83	2.38	0.19
30	85.61	0.23	1.03	9.02	23.36	36.73	0.56	2.80	37.29	2.29	0.14
60	82.10	0.09	0.79	0.93	16.31	26.40	0.42	0.93	32.34	1.36	0.09

Table 8. Average rejection rates on test data of the Wisconsin breast cancer data set.

Number of rules	*c*	*s*	Product	*c* with minimum *s*-level			*s* with minimum *c*-level			SLAVE	Castro
				s: 0.1	*s*: 0.2	*s*: 0.3	*c*: 0.6	*c*: 0.7	*c*: 0.8		
2	89.66	0.00	1.35	50.86	50.94	32.05	0.00	0.40	0.89	1.29	1.76
4	85.93	0.00	1.13	41.70	41.54	21.71	0.00	0.32	0.32	1.13	1.43
6	83.73	0.00	0.92	36.65	36.49	16.91	0.00	0.13	0.04	0.97	0.38
8	82.24	0.00	0.38	31.23	31.00	12.97	0.00	0.00	0.00	0.57	0.18
10	81.33	0.00	0.13	26.33	26.91	10.42	0.00	0.00	0.00	0.16	0.07
60	74.57	0.00	0.00	12.21	10.50	0.98	0.00	0.00	0.00	0.00	0.00

Table 9. Average rejection rates on test data of the wine data set.

Number of rules	*c*	*s*	Product	*c* with minimum *s*-level			*s* with minimum *c*-level			SLAVE	Castro
				s: 0.1	*s*: 0.2	*s*: 0.3	*c*: 0.6	*c*: 0.7	*c*: 0.8		
3	73.20	0.06	4.16	72.30	73.82	73.76	0.00	0.00	4.72	6.18	1.35
6	65.22	0.00	0.51	65.56	65.84	65.84	0.00	0.00	0.51	1.40	0.00
9	51.69	0.00	0.22	31.46	34.94	35.17	0.00	0.00	0.00	1.12	0.00
12	48.82	0.00	0.00	23.54	28.48	29.16	0.00	0.00	0.00	0.28	0.00
15	45.90	0.00	0.00	19.21	18.71	19.49	0.00	0.00	0.00	0.22	0.00
60	42.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 10. Average rejection rates on test data of the sonar data set.

Number of rules	c	s	Product	c with minimum s-level			s with minimum c-level			SLAVE	Castro
				s: 0.1	s: 0.2	s: 0.3	c: 0.6	c: 0.7	c: 0.8		
2	96.59	0.00	0.00	96.35	96.83	96.63	0.19	1.59	23.37	0.48	0.48
4	93.17	0.00	0.00	91.39	94.28	93.46	0.00	0.53	15.29	0.00	0.19
6	89.09	0.00	0.00	68.80	88.46	87.12	0.00	0.29	11.11	0.00	0.10
8	87.69	0.00	0.00	55.38	81.97	80.82	0.00	0.10	8.65	0.00	0.00
10	86.92	0.00	0.00	48.65	76.25	74.81	0.00	0.00	6.44	0.00	0.00
60	78.22	0.00	0.00	27.84	3.17	3.32	0.00	0.00	0.05	0.00	0.00

Table 11. Average rule length over extracted rules for the glass data set.

Number of rules	c	s	Product	c with minimum s-level			s with minimum c-level			SLAVE	Castro
				s: 0.1	s: 0.2	s: 0.3	c: 0.6	c: 0.7	c: 0.8		
6	1.50	2.49	2.79	1.40	1.04	1.00	2.69	2.71	2.72	2.85	2.57
12	1.56	2.51	2.81	1.40	1.08	1.00	2.73	2.76	2.76	2.85	2.59
18	1.60	2.56	2.84	1.40	1.11	1.00	2.76	2.78	2.79	2.84	2.59
24	1.66	2.61	2.84	1.42	1.14	1.00	2.78	2.81	2.80	2.82	2.62
30	1.73	2.64	2.85	1.43	1.15	1.00	2.80	2.83	2.78	2.83	2.65
60	1.93	2.71	2.86	1.52	1.13	1.00	2.85	2.87	2.76	2.82	2.72

Table 12. Average rule length over extracted rules for the Wisconsin breast cancer data set.

Number of rules	c	s	Product	c with minimum s-level			s with minimum c-level			SLAVE	Castro
				s: 0.1	s: 0.2	s: 0.3	c: 0.6	c: 0.7	c: 0.8		
2	1.80	0.50	1.99	2.00	2.00	2.00	0.50	1.00	1.08	1.99	1.97
4	2.04	0.75	1.96	2.07	2.00	2.00	0.75	1.00	1.18	1.93	1.90
6	2.16	0.84	1.95	2.07	2.00	2.00	0.84	1.02	1.22	1.94	1.92
8	2.24	0.89	1.95	2.08	2.00	2.00	0.89	1.03	1.26	1.93	1.95
10	2.29	0.92	1.95	2.09	2.00	2.00	0.92	1.12	1.38	1.94	1.98
60	2.46	1.55	2.22	2.36	2.00	2.00	1.56	1.64	1.76	2.24	2.23

Table 13. Average rule length over extracted rules for the wine data set.

Number of rules	c	s	Product	c with minimum s-level			s with minimum c-level			SLAVE	Castro
				s: 0.1	s: 0.2	s: 0.3	c: 0.6	c: 0.7	c: 0.8		
3	1.00	1.23	1.00	1.00	1.00	1.00	1.17	1.30	1.02	1.00	1.00
6	1.00	1.42	1.05	1.00	1.00	1.00	1.13	1.42	1.21	1.18	1.00
9	1.19	1.38	1.13	1.13	1.00	1.00	1.18	1.55	1.47	1.34	1.01
12	1.30	1.35	1.23	1.22	1.00	1.00	1.31	1.62	1.59	1.45	1.17
15	1.37	1.31	1.31	1.28	1.00	1.00	1.40	1.62	1.68	1.53	1.27
60	1.82	1.70	1.75	2.21	1.04	1.00	1.81	1.98	2.00	1.89	1.75

Table 14. Average rule length over extracted rules for the sonar data set.

Number of rules	c	s	Product	c with minimum s-level s: 0.1	 s: 0.2	 s: 0.3	s with minimum c-level c: 0.6	 c: 0.7	 c: 0.8	SLAVE	Castro
2	1.00	0.52	0.52	1.00	1.00	1.00	1.06	1.55	1.78	1.00	1.23
4	1.02	0.78	0.78	1.02	1.00	1.00	1.26	1.75	1.82	1.15	1.37
6	1.15	0.88	0.87	1.11	1.00	1.00	1.33	1.81	1.87	1.34	1.46
8	1.23	0.94	0.92	1.17	1.00	1.00	1.39	1.84	1.89	1.48	1.54
10	1.28	0.99	0.95	1.22	1.00	1.00	1.44	1.86	1.90	1.56	1.60
60	1.48	1.40	1.33	1.44	1.10	1.00	1.77	1.94	1.95	1.85	1.87

Let us examine our experimental results in detail. First we compare our results with the reported results by fuzzy rule-based systems in Table 1. For the glass data set in Table 3, a 42.24% average error rate was obtained by six fuzzy rules in the case of the support criterion with the minimum confidence level 0.7. This average result is almost the same as a 42.1% average error rate in Table 1 by 8.5 fuzzy rules in Sanchez et al. (2001). For the Wisconsin breast cancer data set in Table 4, a 4.71% average error rate was obtained by six fuzzy rules in the case of the Castro criterion. This average result is almost the same as a 4.65% error rate in Table 1 by 5.1 fuzzy rules in Sanchez et al. (2001). These observations show that good rule sets can be obtained by our simple greedy method for the glass data set and the Wisconsin breast cancer data set if we appropriately choose a rule selection criterion. On the other hand, our results in Table 5 (e.g., a 7.25 average error rate by six fuzzy rules) on the wine data are inferior to the reported result (i.e., a 3.24% average error rate by 5.2 fuzzy rules; see Table 1) in Castillo et al. (2001). This observation shows that good rule sets are not always obtained by our simple greedy method.

Next we compare our results with the reported results by the C4.5 algorithm in Table 2. While the average error rates by fuzzy rules in Table 3 for the glass data are much inferior to those by the C4.5 algorithm in Table 2, the performance of fuzzy rules is comparable to the C4.5 algorithm on the other data sets. For example, the best average error rate by fuzzy rules in Table 4 on the Wisconsin breast cancer data was 3.97% while the best result by the C4.5 algorithm in Table 2 was 5.1%. A 5.11% average error rate was obtained for the wine data by fuzzy rules while the best result by the C4.5 algorithm was 5.6% in Table 2. The best average error rate on the sonar data by fuzzy rules was 23.75% in Table 6, which is almost the same as the best result by the C4.5 algorithm in Table 2 (i.e., 24.6%). It should be noted that threshold values were carefully specified in a sophisticated manner for each continuous attribute in each variant of the C4.5 algorithm in Table 2. That is, interval partitions were appropriately specified based on the distribution of training patterns. On the other hand, we simply used homogeneous fuzzy partitions independent of the distribution of training patterns. High performance of fuzzy rules was realized by their ability to adjust classification boundaries through rule weights as shown in Fig. 2.

In Tables 3-6, it is very interesting to observe that the classification performance does not monotonically increase with the number of fuzzy rules in some cases (e.g., see the last column of Table 3). This is because the interaction among selected fuzzy rules was not taken into account in our simple greedy method. This observation suggests the possibility that better results can be obtained by searching for appropriate combinations of a smaller number of fuzzy rules. In the next section, we will examine this possibility using genetic algorithms. It should be noted that the above-mentioned drawback of our simple greedy method was partially resolved in the iterative fuzzy GBML algorithms (Gonzalez and Perez (1999), Castillo et al. (2001), Castro et al. (2001)) by removing training patterns that had already been covered by the previously found rules.

In our computational experiments, we used the restriction on the rule length for generating short fuzzy rules and for decreasing the CPU time. From the point of view of the CPU time, this restriction is not always necessary for all the seven rule selection criteria. In the case of the support criterion, we can utilize the well-known Apriori algorithm, which was proposed for extracting non-fuzzy association rules in the field of data mining (Agrawal et al. (1996)). The following relation holds for fuzzy association rules from the definition of the support in (17) when the inclusion relation $\mathbf{A}_q \subseteq \mathbf{B}_q$ holds:

$$s(\mathbf{A}_q \Rightarrow \text{Class } C_q) \ \le \ s(\mathbf{B}_q \Rightarrow \text{Class } C_q), \tag{23}$$

where the inclusion relation between the two fuzzy vectors $\mathbf{A}_q$ and $\mathbf{B}_q$ (i.e., between the two antecedent parts) is defined by their elements as

$$\mathbf{A}_q \subseteq \mathbf{B}_q \ \Leftrightarrow \ A_{qi} \subseteq B_{qi} \text{ for } \forall i\,. \tag{24}$$

The inequality in (23) means that the support of a fuzzy rule does not increase when we add an additional condition to its antecedent part. For example, the inequality condition in (23) holds for $\mathbf{A}_q =$ (*small*, *small*, *don't care*) and $\mathbf{B}_q =$ (*small*, *don't care*, *don't care*). When the support of the fuzzy rule "$\mathbf{B}_q \Rightarrow \text{Class } C_q$" is not high, we do not have to examine any fuzzy rules "$\mathbf{A}_q \Rightarrow \text{Class } C_q$" satisfying the inclusion relation $\mathbf{A}_q \subseteq \mathbf{B}_q$. Thus we can efficiently perform the fuzzy rule extraction using the support criterion. This search technique can be also utilized for the product criterion, the SLAVE criterion, and the Castro criterion using the following relations:

$$c(R_q)\cdot s(R_q) \le s(\mathbf{A}_q \Rightarrow \text{Class } C_q), \tag{25}$$

$$f_{\text{SLAVE}}(R_q) \le s(\mathbf{A}_q \Rightarrow \text{Class } C_q), \tag{26}$$

$$f_{\text{Castro}}(R_q) \le s(\mathbf{A}_q \Rightarrow \text{Class } C_q) \times \frac{|\text{Class } \overline{C_q}\,|}{m}. \tag{27}$$

On the other hand, the confidence has no monotonicity property similar to (23). The confidence tends to increase as an additional condition is added to the antecedent part (i.e., as the rule length increases). Thus the confidence criterion usually needs the restriction on the rule length. In the case of the support criterion with the minimum confidence level, the necessity of the restriction on the rule length depends on the value of the minimum confidence level. When the minimum confidence level is very low, this criterion is almost the same as the support criterion. Thus the restriction on the rule length is not necessary. On the other hand, this criterion with no restriction on rule length may require long CPU time when the minimum confidence level is high. In this case, we may have to examine a huge number of long fuzzy rules when short fuzzy rules do not satisfy the minimum confidence level. The confidence criterion with the minimum support level also requires the restriction on the rule length when the minimum support level is very low. In this case, the minimum support level is not likely to play an important role for decreasing the search space.

5. Genetic Algorithm-Based Rule Selection

In the above computational experiments, we did not take into account the combinatorial effect of generated fuzzy rules (i.e., the interaction among them). Thus we could not always find good rule sets. We will be able to find better rule sets with higher classification performance by directly searching for rule sets (i.e., combinations of fuzzy rules) as in Pittsburgh-style fuzzy GBML algorithms (e.g., see Cordon et al. (2001)). Such a fuzzy GBML algorithm, however, usually requires long CPU time and large memory storage for finding good rule sets for high-dimensional problems. A promising idea for efficiently finding good rule sets is to search for good subsets of fuzzy rules generated by a heuristic rule selection criterion (Ishibuchi and Yamamoto (2002a, 2003b)). In this section, we demonstrate that good rule sets can be obtained by genetic algorithms as subsets of fuzzy rules generated in the previous section. Of course, the performance of obtained rule sets (i.e., obtained fuzzy rule-based classification systems) totally depends on the choice of candidate fuzzy rules, which are generated by heuristic rule selection in the previous section. When the number of candidate fuzzy rules is very small, we can examine all of their subsets as fuzzy rule-based classification systems. By increasing the number of candidate fuzzy rules, we can increase the chance to find good subsets. At the same time, the size of the search space is exponentially increased. Thus we need a good heuristic rule selection criterion for finding a tractable number of candidate fuzzy rules. It is a very difficult task to find a good rule set without such a heuristic procedure. This is because the search space with possible fuzzy rules is huge for high-dimensional pattern classification problems involving tens of input variables (e.g., the number of possible fuzzy rules was 15^{60} for the sonar data with 60 attributes in the previous section).

In our computational experiments, we used 60 fuzzy rules generated by the product criterion as candidate rules in rule selection. A subset *S* of those 60 fuzzy rules was handled as an individual and represented by a binary string of the length 60 (i.e., $S = s_1 s_2 \ldots s_{60}$). In this coding, $s_q = 1$ and $s_q = 0$ mean the inclusion of the *q*-th fuzzy rule R_q in *S* and the exclusion of R_q from *S*, respectively. A genetic algorithm was used for finding the best subset with respect to the following fitness function:

$$fitness(S) = w_1 \cdot f_1(S) - w_2 \cdot f_2(S) - w_3 \cdot f_3(S)\,, \tag{28}$$

where $f_1(S)$ is the number of correctly classified training patterns by *S*, $f_2(S)$ is the number of fuzzy rules in *S*, $f_3(S)$ is the total number of antecedent conditions (i.e., total rule length) in *S*, and w_i is a positive weight for the *i*-th objective $f_i(S)$. In our computational experiments, the weight values were specified as $w_1 = 100$, $w_2 = 10$ and $w_3 = 1$.

As in the previous section, we executed the whole 10-CV procedure five times for calculating the average error rate on test data for each data set. A genetic algorithm with the following specifications was used for searching for a rule set with the maximum value of the fitness function in (28).

Selection: Standard binary tournament selection,
Crossover: Standard uniform crossover with the crossover rate 0.8,
Mutation: Standard flip-flop mutation with the mutation rate 1/60,
Population size: 500 strings,
Generation update: Standard generation model with a single elite individual,
Stopping condition: 5000 generations.

Experimental results were summarized in Table 15. From this table, we can see that the average error rates were improved by removing unnecessary fuzzy rules (i.e., finding good subsets of fuzzy rules extracted by the simple greedy method) except for the case of the wine data. It should be noted that the number of fuzzy rules was significantly decreased from 60. This means that the interpretability of fuzzy rule-based systems was significantly improved.

Table 15. Genetic algorithm-based rule selection from 60 candidate fuzzy rules.

Data set	Before rule selection		After rule selection	
	Error rate	# of rules	Error rate	# of rules
Glass	46.07	60	41.84	7.72
Wisconsin	3.97	60	3.84	4.58
Wine	6.91	60	6.95	5.44
Sonar	43.03	60	31.25	4.00

Table 16. Genetic algorithm-based rule selection from 600 candidate fuzzy rules.

Data set	Before rule selection		After rule selection	
	Error rate	# of rules	Error rate	# of rules
Glass	44.72	600	40.38	11.88
Wisconsin	3.65	600	3.25	7.76
Wine	4.49	600	6.90	5.68
Sonar	29.28	600	24.06	7.04

We also performed similar computational experiments by extracting much more fuzzy rules (i.e., 600 rules) as candidate rules using our simple greedy method. A genetic algorithm was used for finding good subsets of those 600 fuzzy rules. We used the same parameter specifications as in the previous computational experiments except for the mutation probability. We specified it as 1/600 because the string length was 600. Experimental results were summarized in Table 16. From the comparison between Table 15 and Table 16, we can see that lower error rates were obtained in Table 16 than Table 15. This improvement in the performance of selected fuzzy rules was realized at the cost of the increase in the CPU time and the required memory storage. For example, the average CPU time for a single run of the genetic algorithm for the glass data set was 8.33 minutes in Table 15 with 60 candidate rules while it was 13.75 minutes in Table 16 with 600 rules. Moreover the increase in the number of candidate fuzzy rules significantly increases the difficulty in the search for good rule sets because the size of the search space is 2^Q where Q is the number of candidate fuzzy rules. Thus the number of candidate rules should be appropriately specified. From the comparison of our results in Table 16 with the reported results in the literature in Table 1 and Table 2, we can see that very good results were obtained by the genetic algorithm-based rule selection for the Wisconsin breast cancer data (a 3.25% average error rate) and the sonar data (i.e., a 24.06% average error rate).

For the simplicity of explanation, we used a single-objective genetic algorithm for rule selection in this section. Thus we had to pre-specify the weight values in the fitness function in (28). In our computational experiments, we performed the weight specification in a trial-and-error manner. If we use a three-objective genetic algorithm for rule selection, such a trail-and-error specification is not necessary and better results can be obtained (Ishibuchi and Yamamoto (2003a)).

6. Conclusion

In this paper, we first examined the performance of fuzzy rules extracted from numerical data using heuristic rule selection criteria through computational experiments on four well-known data sets with many continuous attributes (i.e., glass data, Wisconsin breast cancer data, wine data, and sonar data). Experimental results showed that better results were obtained from composite criteria of the confidence and the support than their individual use. It was also shown that the SLAVE and Castro criteria also worked well. In our computational experiments, we obtained good results using a simple greedy rule selection method in comparison with the C4.5 algorithm for the three data sets except for the glass data when we appropriately chose a rule selection criterion and the number of extracted fuzzy rules. Finally we showed that the classification performance of extracted fuzzy rules was improved by searching for their good subsets by genetic algorithms. This suggests that the combinatorial effect of fuzzy rules (i.e., the interaction among them) should be taken into account when we design fuzzy rule-based systems.

Our comparative study in this paper is based on computational experiments for some benchmark data sets. From those experiments, we do not conclude which criterion is the best among the examined ones. As our experimental results show, the choice of a heuristic rule selection criterion seems to be problem-dependent while we can say that better results were obtained from composite criteria of the confidence and the support than their individual use. Theoretical studies as well as further computational experiments may be required for providing guidelines for the choice of a heuristic rule selection criterion for a particular pattern classification problem.

Acknowledgment

The authors would like to thank the financial support from Casio Science Promotion Foundation.

References

Agrawal, R., H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo (1996). "Fast discovery of association rules," in U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (eds.) *Advances in Knowledge Discovery and Data Mining*, 307-328, AAAI Press.

Agrawal, R. and R. Srikant (1994). "Fast algorithms for mining association rules," *Proc. of 20th International Conference on Very Large Data Bases*, 487-499. Expanded version is available as IBM Research Report RJ9839, 1994.

Casillas, J., O. Cordon, F. Herrera, and L. Magdalena (2003a). *Interpretability Issues in Fuzzy Modeling*, Springer-Verlag.

Casillas, J., O. Cordon, F. Herrera, and L. Magdalena (2003b). *Accuracy Improvements in Linguistic Fuzzy Modeling*, Springer-Verlag.

Castillo, L., A. Gonzalez, and R. Perez (2001). "Including a simplicity criterion in the selection of the best rule in a genetic fuzzy learning algorithm," *Fuzzy Sets and Systems* 120 (2), 309-321.

Castro, L., J. J. Castro-Schez, and J. M. Zurita (2001). "Use of a fuzzy machine learning technique in the knowledge acquisition process," *Fuzzy Sets and Systems* 123 (3), 307-320.

Cordon, O., M. J. del Jesus, and F. Herrera (1999). "A proposal on reasoning methods in fuzzy rule-based classification systems," *International Journal of Approximate Reasoning* 20 (1), 21-45.

Cordon, O., F. Herrera, F. Hoffman, and L. Magdalena (2001). *Genetic Fuzzy Systems*, World Scientific.

Elomaa, T. and J. Rousu (1999). "General and efficient multisplitting of numerical attributes," *Machine Learning* 36, 201-244.

Fayyad, U. M., G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (1996). *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press.

Gonzalez, A. and R. Perez (1999). "SLAVE: A genetic learning system based on an iterative approach," *IEEE Trans. on Fuzzy Systems* 7 (2), 176-191.

Hong, T. -P., C. -S. Kuo, and S. -C. Chi (2001). "Trade-off between computation time and number of rules for fuzzy mining from quantitative data," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 9 (5), 587-604.

Ishibuchi, H. and T. Nakashima (1999). "Improving the performance of fuzzy classifier systems for pattern classification problems with continuous attributes," *IEEE Trans. on Industrial Electronics* 46 (6), 157-168.

Ishibuchi, H. and T. Nakashima (2001). "Effect of rule weights in fuzzy rule-based classification systems," *IEEE Trans. on Fuzzy Systems* 9 (4), 506-515.

Ishibuchi, H., T. Nakashima, and T. Morisawa (1999). “Voting in fuzzy rule-based systems for pattern classification problems,” *Fuzzy Sets and Systems* 103 (2), 223-238.

Ishibuchi, H., T. Nakashima, and T. Murata (1999). “Performance evaluation of fuzzy classifier systems for multi-dimensional pattern classification problems,” *IEEE Trans. on Systems, Man, and Cybernetics- Part B: Cybernetics* 29 (5), 601-618.

Ishibuchi, H., T. Nakashima, and T. Murata (2001). “Three-objective genetics-based machine learning for linguistic rule extraction,” *Information Sciences* 136 (1-4), 109-133.

Ishibuchi, H., K. Nozaki, and H. Tanaka (1992). “Distributed representation of fuzzy rules and its application to pattern classification,” *Fuzzy Sets and Systems* 52 (1), 21-32.

Ishibuchi, H. and T. Yamamoto (2002a). “Fuzzy rule selection by data mining criteria and genetic algorithms,” *Proc. of 2002 Genetic and Evolutionary Computation Conference*, 399-406.

Ishibuchi, H. and T. Yamamoto (2002b). “Comparison of heuristic rule weight specification methods,” *Proc. of 11th IEEE International Conference on Fuzzy Systems*, 908-913.

Ishibuchi, H. and T. Yamamoto (2002c). “Effect of fuzzy discretization in fuzzy rule-based systems for classification problems with continuous attributes,” *Archives of Control Sciences* 12 (4), 351-378.

Ishibuchi, H. and T. Yamamoto (2003a). “Effects of three-objective genetic rule selection on the generalization ability of fuzzy rule-based systems,” *Proc. of Second International Conference on Evolutionary Multi-Criterion Optimization*, 608-622.

Ishibuchi, H. and T. Yamamoto (2003b). “Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining,” *Fuzzy Sets and Systems* (in press).

Ishibuchi, H., T. Yamamoto, and T. Nakashima (2001). “Fuzzy data mining: Effect of fuzzy discretization,” *Proc. of 1st IEEE International Conference on Data Mining*, 241-248.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*, Morgan Kaufmann.

Quinlan, J. R. (1996). “Improved use of continuous attributes in C4.5,” *Journal of Artificial Intelligence Research* 4, 77-90.

Sanchez, L., I. Couso, and J. A. Corrales (2001). “Combining GP operators with SA search to evolve fuzzy rule base classifiers,” *Information Sciences* 136 (1-4), 175-191.

van den Berg, J., U. Kaymak, and W. -M. van den Bergh (2002). “Fuzzy classification using probability based rule weighting,” *Proc. of 11th IEEE International Conference on Fuzzy Systems*, 991-996.