H. Ishibuchi and T. Yamamoto, “Rule weight specification in fuzzy rule-based classification systems,” *IEEE Trans. on Fuzzy Systems*, vol. 13, no. 4, pp. 428-435, August 2005.

<u>Final Version of the Accepted Paper</u>

Rule Weight Specification in Fuzzy Rule-Based Classification Systems

Hisao Ishibuchi*, *Member, IEEE,* and Takashi Yamamoto, *Student Member, IEEE*

Department of Computer Science and Intelligent Systems, Osaka Prefecture University

* Corresponding Author:

Prof. Hisao Ishibuchi
Department of Computer Science and Intelligent Systems
Osaka Prefecture University
1-1 Gakuen-cho, Sakai, Osaka 599-8531, Japan
E-mail: hisaoi@ie.osakafu-u.ac.jp

Rule Weight Specification in Fuzzy Rule-Based Classification Systems

Hisao Ishibuchi*, *Member, IEEE*, and Takashi Yamamoto, *Student Member, IEEE*
Department of Industrial Engineering, Osaka Prefecture University

Abstract: This paper shows how the rule weight of each fuzzy rule can be specified in fuzzy rule-based classification systems. First we propose two heuristic methods for rule weight specification. Next the proposed methods are compared with existing ones through computer simulations on artificial numerical examples and real-world pattern classification problems. Simulation results show that the proposed methods outperform the existing ones in the case of multi-class pattern classification problems with many classes.

Index Term: Fuzzy systems, pattern classification, rule generation, rule selection, data mining.

I. Introduction

We have already demonstrated that rule weights have a significant effect on the classification performance of fuzzy rule-based systems [1]. In this paper, we examine some heuristic methods for rule weight specification. We use fuzzy rules of the following type for an n-dimensional pattern classification problem.

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

where $\mathbf{x} = (x_1, ..., x_n)$ is an n-dimensional pattern vector, A_{qi} is an antecedent fuzzy set for the i-th attribute, C_q is a consequent class, and CF_q is a certainty grade (i.e., rule weight). Various types of fuzzy rules have been used for pattern classification problems. For example, Cordon et al. [2] examined three types of fuzzy rules. One has a single consequent class with no rule weight, another is the same as (1), and the other has multiple consequent classes. The third type with multiple consequent classes is written for M-class pattern classification problems as

$$\text{Rule } R_q : \text{If } x_1 \text{ is } A_{q1} \text{ and ... and } x_n \text{ is } A_{qn} \text{ then Class 1 with } CF_{q1} \text{ and ... and Class } M \text{ with } CF_{qM} , \tag{2}$$

where CF_{qh} is a certainty grade for Class h. In this paper, we mainly use fuzzy rules of the form in (1). Fuzzy rules with multiple consequent classes in (2) are used only when the three types of fuzzy rules are compared with one another. Fuzzy rules of the form in (2) were also examined in [3] from the viewpoint of fuzzy conditional probability.

Many studies on fuzzy rule-based classification systems (e.g., [4]-[7]) did not use rule weights. In most of those studies, antecedent fuzzy sets were generated and adjusted from numerical data. As shown in [8], the learning of rule weights can be replaced by the modification of the membership functions of antecedent fuzzy sets. We do not, however, adjust the membership functions of antecedent fuzzy sets because their adjustment may degrade the comprehensibility of fuzzy rule-based systems. Of course, many approaches have been proposed for finding a good compromise between the accuracy of fuzzy rule-based systems and their comprehensibility. Those approaches often use constraint conditions on the membership functions of antecedent fuzzy sets (e.g., [9]).

This paper is organized as follows. In Section II, we explain two heuristic methods for rule weight specification in [2], [10] using the terminology in data mining: *confidence* and *support* [11]. We also propose different heuristic methods in Section II. Characteristic features of each method are visually demonstrated through computer simulations on artificial numerical examples in Section III. In our computer simulations, we also compare fuzzy rules of the form in (1) with those in (2). Moreover, we compare two fuzzy reasoning methods with each other in Section III: a single winner method and a weighted vote method. In Section IV, we examine the classification performance of fuzzy rule-based systems designed by each heuristic method through computer simulations on real-world pattern classification problems. Finally Section V concludes this paper.

II. HEURISTIC RULE WEIGHT SPECIFICATION

A. Rule Evaluation Measures in Data Mining

In the field of data mining [11], two measures (i.e., *confidence* and *support*) are frequently used to evaluate association rules. The fuzzy rule in (1) can be viewed as a fuzzy association rule $\mathbf{A}_q \Rightarrow C_q$ where $\mathbf{A}_q = (A_{q1}, ..., A_{qn})$. In this subsection, we briefly explain fuzzy versions [12], [13] of the two rule evaluation measures.

Let us assume that *m* labeled patterns $\mathbf{x}_p = (x_{p1}, ..., x_{pn})$, $p = 1,2,...,m$ are given from *M* classes for an *n*-dimensional pattern classification problem. As in our former studies [1], [10], [13], we define the compatibility grade of each training pattern $\mathbf{x}_p$ with the antecedent $\mathbf{A}_q$ by the product operation as

$$\mu_{\mathbf{A}_q}(\mathbf{x}_p) = \mu_{A_{q1}}(x_{p1}) \times \cdots \times \mu_{A_{qn}}(x_{pn}), \qquad (3)$$

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

The confidence of the fuzzy rule $\mathbf{A}_q \Rightarrow C_q$ is written as follows [12], [13]:

$$c(\mathbf{A}_q \Rightarrow C_q) = \frac{\sum_{\mathbf{x}_p \in \text{Class } C_q} \mu_{\mathbf{A}_q}(\mathbf{x}_p)}{\sum_{p=1}^{m} \mu_{\mathbf{A}_q}(\mathbf{x}_p)}. \tag{4}$$

The confidence can be viewed as measuring the validity of $\mathbf{A}_q \Rightarrow C_q$. It can be also viewed as a numerical approximation of the conditional probability [3]. On the other hand, the support of $\mathbf{A}_q \Rightarrow C_q$ is written as follows [12], [13]:

$$s(\mathbf{A}_q \Rightarrow C_q) = \frac{\sum_{\mathbf{x}_p \in \text{Class } C_q} \mu_{\mathbf{A}_q}(\mathbf{x}_p)}{m}. \tag{5}$$

The support can be viewed as measuring the coverage of training patterns by $\mathbf{A}_q \Rightarrow C_q$.

B. Heuristic Methods for Rule Weight Specification

First we explain the determination of the consequent class. Let us assume that a set of antecedent fuzzy sets is given for each attribute. The antecedent part of each fuzzy rule (i.e., $\mathbf{A}_q$) is constructed by combining antecedent fuzzy sets for *n* attributes. The consequent C_q of the fuzzy rule $\mathbf{A}_q \Rightarrow C_q$ in (1) is determined by finding the class with the maximum confidence for the antecedent $\mathbf{A}_q$:

$$c(\mathbf{A}_q \Rightarrow C_q) = \max\{c(\mathbf{A}_q \Rightarrow \text{Class } h) \mid h = 1,2,...,M\}. \tag{6}$$

When the consequent C_q cannot be uniquely determined in (6), we do not generate any fuzzy rule with the antecedent $\mathbf{A}_q$.

The confidence $c(\mathbf{A}_q \Rightarrow C_q)$ can be used as the rule weight CF_q of the fuzzy rule $\mathbf{A}_q \Rightarrow C_q$ as in Cordon et al. [2] and Berg et al. [3]. That is, one definition of the rule weight is

$$CF_q^{\text{I}} = c(\mathbf{A}_q \Rightarrow C_q), \tag{7}$$

where the superscript “I” shows that CF_q^{I} is the first definition of CF_q. In our former studies [1], [10], [13], we used a different definition of the rule weight:

$$CF_q^{\text{II}} = c(\mathbf{A}_q \Rightarrow C_q) - c_{\text{Ave}}, \tag{8}$$

where c_{Ave} is the average confidence over fuzzy rules with the same antecedent $\mathbf{A}_q$ but different consequent classes from C_q:

$$c_{\text{Ave}} = \frac{1}{M-1} \sum_{\substack{h=1 \\ h \neq C_q}}^{M} c(\mathbf{A}_q \Rightarrow \text{Class } h) \ . \tag{9}$$

In this paper, we propose more intuitive definitions of the rule weight. One definition is based on the difference between the largest and the second largest confidence. That is,

$$CF_q^{\text{III}} = c(\mathbf{A}_q \Rightarrow C_q) - c_{\text{2nd}} \,, \tag{10}$$

where c_{2nd} is the second largest confidence for the antecedent $\mathbf{A}_q$:

$$c_{\text{2nd}} = \max\{c(\mathbf{A}_q \Rightarrow \text{Class } h) \mid h = 1,2,\ldots,M;\, h \neq C_q\} \,. \tag{11}$$

We also propose the following definition:

$$CF_q^{\text{IV}} = c(\mathbf{A}_q \Rightarrow C_q) - c_{\text{Sum}} \,, \tag{12}$$

where c_{Sum} is the sum of the confidence over fuzzy rules with the same antecedent $\mathbf{A}_q$ but different consequent classes from C_q:

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

While CF_q is always positive in the first three definitions, CF_q^{IV} can be negative. We do not use fuzzy rules with negative rule weights in fuzzy rule-based systems. Note that the third and fourth definitions in (10) and (12) are the same as the second definition in (8) when our pattern classification problem involves only two classes (i.e., when $M = 2$). This is because $c_{\text{Ave}} = c_{\text{2nd}} = c_{\text{Sum}}$ holds for the case of $M = 2$ in (9), (11) and (13).

For the fuzzy rule with multiple consequent classes in (2), we directly use the confidence as the certainty grade for each class [2], [3]:

$$CF_{qh} = c(\mathbf{A}_q \Rightarrow \text{Class } h) \,,\ h = 1,2,\ldots,M \ . \tag{14}$$

C. Fuzzy Reasoning for Pattern Classification

First we explain fuzzy reasoning for fuzzy rules with a single consequent class. Let *S* be a set of fuzzy rules of the form in (1). We use a single winner method [14] for classifying new patterns by the rule set *S*. The single winner rule R_w is determined for a new pattern $\mathbf{x}_p = (x_{p1}, \ldots, x_{pn})$ as

$$\mu_{\mathbf{A}_w}(\mathbf{x}_p)\cdot CF_w = \max\{\, \mu_{\mathbf{A}_q}(\mathbf{x}_p)\cdot CF_q \mid R_q \in S\,\}. \tag{15}$$

The new pattern $\mathbf{x}_p$ is classified as Class C_w, which is the consequent class of the winner rule R_w. If multiple fuzzy rules have the same maximum value but different consequent classes for the new pattern $\mathbf{x}_p$ in (15), the classification of $\mathbf{x}_p$ is rejected. The classification is also rejected if no fuzzy rule is compatible with the new pattern $\mathbf{x}_p$.

Next let us consider fuzzy reasoning by fuzzy rules with multiple consequent classes. In this case, the single winner method in (15) is rewritten as

$$\mu_{\mathbf{A}_w}(\mathbf{x}_p)\cdot CF_{wh*} = \max\{\, \mu_{\mathbf{A}_q}(\mathbf{x}_p)\cdot CF_{qh} \mid R_q \in S\,,\ h = 1,2,...,M\}. \tag{16}$$

The new pattern $\mathbf{x}_p$ is classified as Class h^* in (16). It can be easily shown that (16) can be reduced to (15) by defining CF_q for each fuzzy rule R_q with multiple consequent classes in (2) as

$$CF_q = \max\{CF_{qh} \mid h = 1,2,...,M\}. \tag{17}$$

This means that multiple consequent classes are not necessary when we use the single winner method.

We also use a weighted vote method [14] for comparison. When we have fuzzy rules with a single consequent class, each fuzzy rule casts a vote for its consequent class. The strength of the vote is defined by the product of the compatibility grade and the certainty grade. Thus the total strength of the vote for each class is calculated as follows:

$$V_{\text{Class } h}(\mathbf{x}_p) = \sum_{\substack{R_q \in S\\ C_q = h}} \mu_{\mathbf{A}_q}(\mathbf{x}_p)\cdot CF_q\,,\quad h = 1,2,...,M\,. \tag{18}$$

The new pattern $\mathbf{x}_p$ is classified as the class with the maximum total strength of the vote.

On the other hand, the total strength of the vote for each class is written in the case of fuzzy rules with multiple consequent classes as

$$V_{\text{Class } h}(\mathbf{x}_p) = \sum_{R_q \in S} \mu_{\mathbf{A}_q}(\mathbf{x}_p)\cdot CF_{qh}\,,\quad h = 1,2,...,M\,. \tag{19}$$

While each fuzzy rule casts a vote for its single consequent class in (18), it votes for all classes in (19).

In the single winner method as well as the weighted vote method, we multiply the compatibility grade $\mu_{\mathbf{A}_q}(\mathbf{x}_p)$ by the rule weight CF_q. Interpretation of this operation is not straightforward. For example, the first definition of the rule weight is a numerical approximation of the conditional probability of the consequent class. In this case, our fuzzy reasoning methods are viewed as combining two different measures: a linguistic compatibility and a probabilistic validity. In the single winner method, we handle the rule weight as the rule strength, which is an expression of the applicability of each fuzzy rule. That is, the rule weight is related to the choice of a single winner rule among compatible rules. Operationally the multiplication of the compatibility grade by the rule weight is equivalent to the modification of the heights of antecedent fuzzy sets [1], [8]. Thus the rule weight can be viewed as adjusting the compatibility grade (i.e., applicability) of each fuzzy rule to the current input vector. When there is exactly one rule per class in neuro-fuzzy naive Bayes classifiers [15], the rule weight and the compatibility grade are explained as being the prior probability of the consequent class and the probability density function, respectively. Under this interpretation, the multiplication of the compatibility grade by the rule weight becomes natural. In the case of multiple rules per class, this interpretation is not directly applicable (i.e., we have to split each class into multiple subclasses).

III. ILLUSTRATION OF THE EFFECT OF RULE WEIGHTS

A. Simulation Results on Single-Dimensional Problems

For visually illustrating characteristic features of each definition of rule weights, let us consider a two-class pattern classification problem on a single-dimensional pattern space [-2, 2] with the homogeneous fuzzy partition by triangular fuzzy sets in Fig. 1. We assume that an infinite number of training patterns are uniformly distributed in the interval [-2, 2]. We also assume that each training pattern x_p belongs to Class 1 or Class 2 depending on its location as shown in Fig. 2. That is, if $x_p \leq \theta$ then x_p belongs to Class 1, otherwise x_p belongs to Class 2.

Using the homogeneous fuzzy partition in Fig. 1 and the uniform distribution of training patterns in Fig. 2, we can generate the following fuzzy rules:

$$R_{q-2}: \text{If } x \text{ is } A_{q-2} \text{ then Class 1 with } CF_{q-2}, \tag{20}$$

$$R_{q-1}: \text{If } x \text{ is } A_{q-1} \text{ then Class 1 with } CF_{q-1}, \tag{21}$$

$$R_{q}: \quad \text{If } x \text{ is } A_{q} \text{ then Class 1 with } CF_{q}, \tag{22}$$

$$R_{q+1}: \text{If } x \text{ is } A_{q+1} \text{ then Class 2 with } CF_{q+1}, \tag{23}$$

$$R_{q+2}: \text{If } x \text{ is } A_{q+2} \text{ then Class 2 with } CF_{q+2}. \tag{24}$$

From Fig. 1 and Fig. 2, we can see that compatible training patterns with A_{q-2} and A_{q-1} always belong to Class 1. Thus the rule weights of R_{q-2} and R_{q-1} are $CF_{q-2} = 1$ and $CF_{q-1} = 1$ independent of the choice of their definition. The rule weight CF_{q+2} of R_{q+2} is also $CF_{q+2} = 1$ because compatible training patterns with A_{q+2} always belong to Class 2.

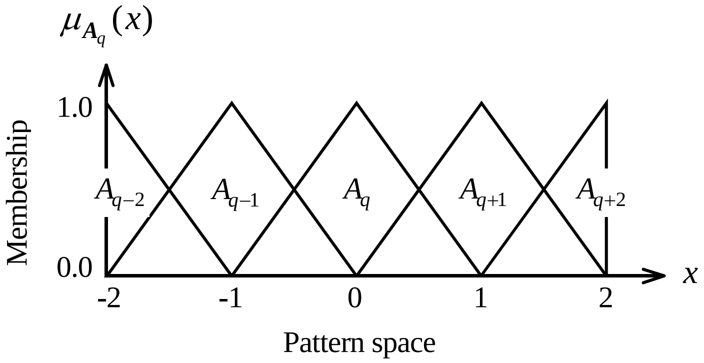


Fig. 1. Homogeneous fuzzy partition by triangular fuzzy sets.

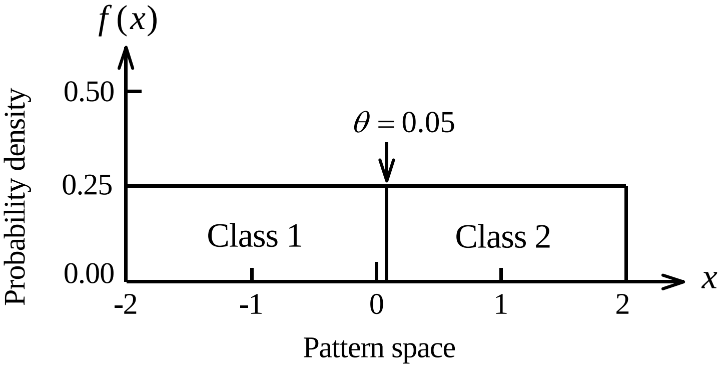


Fig. 2. Distribution of training patterns in our artificial numerical example.

On the other hand, the rule weights of R_q and R_{q+1} are calculated from each definition using the uniform distribution of training patterns in Fig. 2 as

$$CF_q^{\mathrm{I}} = 0.549, \quad CF_{q+1}^{\mathrm{I}} = 0.999, \tag{25}$$

$$CF_q^{\mathrm{II}} = 0.098, \quad CF_{q+1}^{\mathrm{II}} = 0.998. \tag{26}$$

Since our numerical example is a two-class problem (i.e., $M = 2$), the second definition is exactly the same as the third and fourth definitions of rule weights. In (25) and (26), we can observe a large difference in the rule weight CF_q of the fuzzy rule R_q between the first definition and the other definitions. From Fig. 2, the confidence is calculated for "$A_q \Rightarrow$ Class 1" and "$A_q \Rightarrow$ Class 2" as

$$c(A_q \Rightarrow \text{Class 1}) = 0.549, \ \ c(A_q \Rightarrow \text{Class 2}) = 0.451. \tag{27}$$

The difference between these confidence values is small. Thus the rule weight of the fuzzy rule "R_q : $A_q \Rightarrow$ Class 1" is also small in the second definition while it is not so small in the first definition.

Using the five fuzzy rules in (20)-(24), let us estimate the class boundary between Class 1 and

Class 2. When we use the single winner method in (15), the estimated class boundary $\hat{\theta}$ is calculated as $\hat{\theta} = 0.355$ by the first definition and $\hat{\theta} = 0.089$ by the second definition from the equation:

$$\mu_{A_q}(x) \cdot CF_q = \mu_{A_{q+1}}(x) \cdot CF_{q+1} \, . \tag{28}$$

The estimated class boundary $\hat{\theta}$ has a large error in the case of the first definition while it is almost the same as the actual threshold $\theta =$ 0.05 in the second definition. The large error by the first definition is due to the large rule weight CF_q^{I} of the fuzzy rule R_q .

In the same manner, we calculated the estimated class boundary $\hat{\theta}$ for various specifications of the threshold value θ : $\theta = 0.00$, 0.02, ..., 1.00. Simulation results are summarized in Fig. 3 where the desired result $\hat{\theta} = \theta$ is depicted by the diagonal line in Fig. 3. On the other hand, the dotted line corresponding to $\hat{\theta} = 0.5$ shows the estimated class boundary in the case of the fuzzy rules with no rule weights. The estimated class boundary $\hat{\theta}$ has a large error except for the case where the actual class boundary θ is close to 0.5 (i.e., the crossing point of the membership functions A_q and A_{q+1}). This observation suggests that the learning of membership functions is necessary when fuzzy rules have no rule weights.

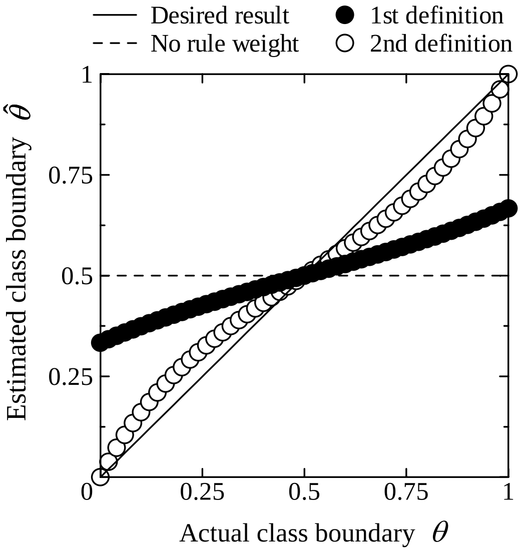


Fig. 3. Comparison of the two definitions of rule weights for our two-class problem.

From Fig. 3, we can see that the estimated class boundary $\hat{\theta}$ is close to the actual threshold θ independent of the value of θ when we use the second definition. On the other hand, the difference between θ and $\hat{\theta}$ is large in the case of the first definition. This observation suggests that the direct use of the confidence as the rule weight (i.e., the first definition) may lead to large classification errors.

Since our numerical example in Fig. 2 involves only two classes, exactly the same results are obtained from the second, third and fourth definitions. Let us extend our numerical example in Fig. 2 to an *M*-class problem ($M > 2$). For simplicity of discussion, we assume that the interval [-2, 2] in Fig. 2 is a part of a larger pattern space (e.g., [-10, 10]). We also assume that training patterns from the other classes (i.e., Class 3, ..., Class *M*) exist far from the interval [-2, 2] in the pattern space (e.g., $x < -5$ or $5 < x$). From these assumptions, we can discuss the specification of rule weights locally in the unit interval [-2, 2]. In this situation, the increase in the number of classes has no effect on the first, third and fourth definitions of rule weights. On the other hand, the second definition depends on the value of M as shown in (9). Thus the second definition is not the same as the third and fourth definitions when pattern classification problems involve more than two classes.

In the same manner as Fig. 3, we calculated the estimated class boundary $\hat{\theta}$ using the second definition for three specifications of *M* (i.e., $M = 2, 5, 10$). Simulation results are summarized in Fig. 4. From this figure, we can see that the difference between the actual class boundary θ and the estimated class boundary $\hat{\theta}$ increases as the value of M increases. When $M = 5$ and $M = 10$, the simulation results by the second definition in Fig. 4 are similar to those by the first definition in Fig. 3.

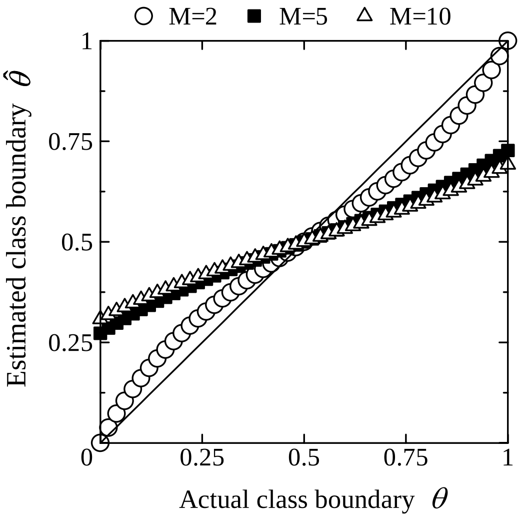


Fig. 4. Simulation results by the second definition of rule weights for *M*-class problems.

For discussing the difference between the third and fourth definitions, let us consider the situation where several minor classes exist in the region of Class 2 in Fig. 2 (i.e., $\theta < x \leq 2$). More specifically, let us assume that Class 2 patterns in Fig. 2 can be further divided into a single major class (say Class 2) and several minor classes (say Class 3, 4, ..., *M*). In this case, the rule weight of " R_q : A_q $\Rightarrow$ Class

1” is calculated by the fourth definition as $CF_q^{\mathrm{IV}} = 0.098$ independent of the subdivision of Class 2 patterns. On the contrary, the rule weight CF_q^{III} by the third definition is increased by the subdivision of Class 2 patterns. Such an increase in the rule weight may lead to the increase in the estimation error similar to Fig. 4. This suggests the possibility that better results are obtained for multi-class pattern classification problems by the fourth definition than the other definitions.

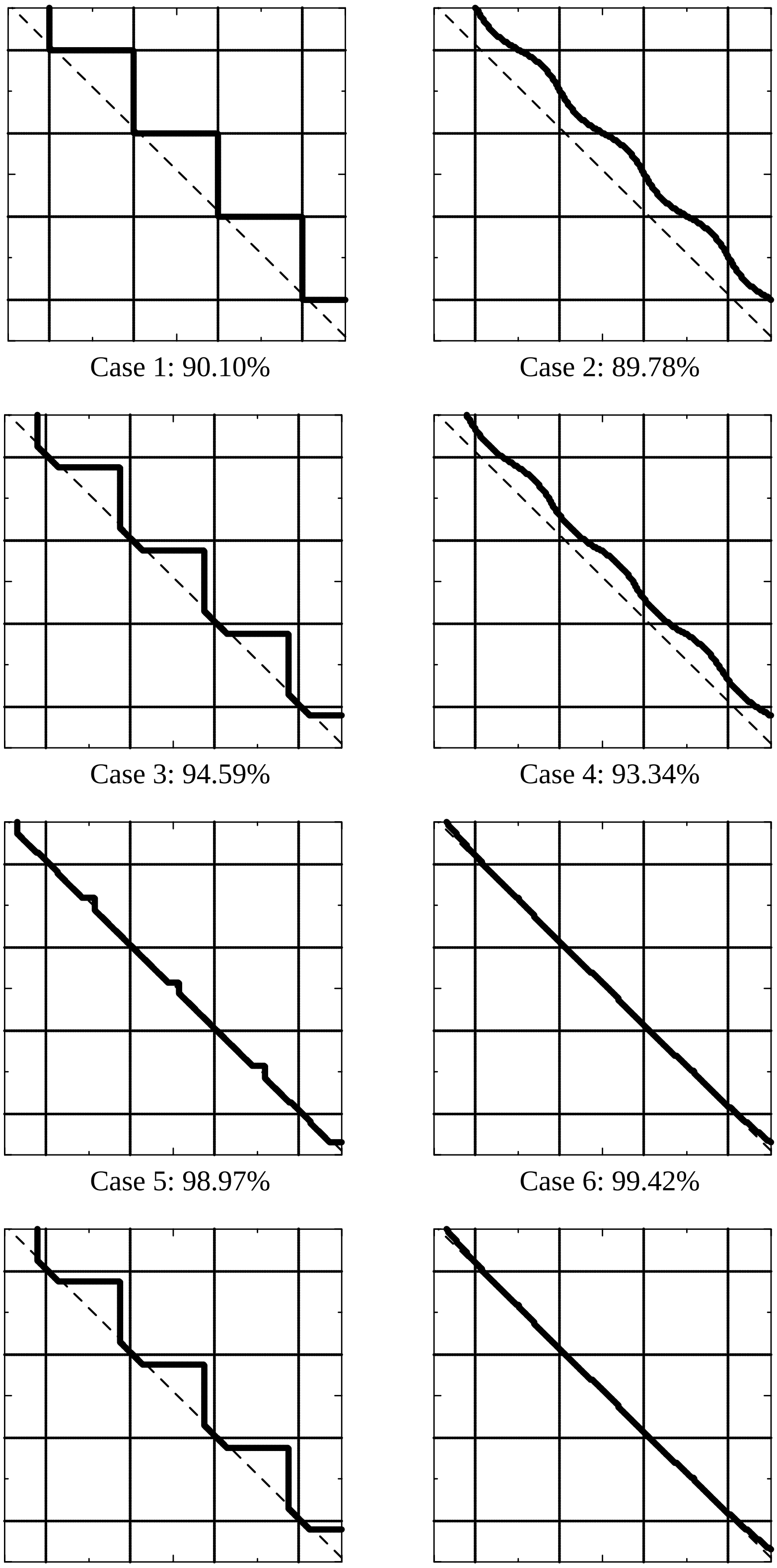
B. Simulation Results on Two-Dimensional Problems

Let us consider a two-class pattern classification problem with a two-dimensional pattern space $[-2, 2]\times[-2, 2]$. We assume that each axis of the pattern space is partitioned into five triangular fuzzy sets in the same manner as Fig. 1. We also assume that training patterns belong to Class 1 only when $x_1 + x_2 \le \theta$. In the two-dimensional pattern space, we generated 160801 training patterns using the 401×401 grid where $x_i = -2.00$, -1.99, ..., 2.00 for $i = 1,2$. Each training pattern is assigned to Class 1 or Class 2 according to its location (i.e., If $x_1 + x_2 \le \theta$ then Class 1 otherwise Class 2). We can generate 25 fuzzy rules from the training patterns using the five antecedent fuzzy sets on each axis of the two-dimensional pattern space.

In our computer simulations, we examined eight cases in Table 1. Simulation results are shown in Fig. 5 where $\theta = 0.05$. In each figure in Fig. 5, the actual class boundary (i.e., $x_1 + x_2 = 0.05$) and the estimated class boundary are shown by a dotted line and a thick line, respectively. The thin lines in each figure show the 5×5 grid by the five antecedent fuzzy sets on each axis. The classification rate on the 160801 training patterns is also shown for each figure.

Table 1. Examined combinations of fuzzy rules and fuzzy reasoning methods.

Case	Consequent	Rule weight	Classification
Case 1	Single class	No rule weight	Single winner
Case 2	Single class	No rule weight	Weighted vote
Case 3	Single class	1st definition	Single winner
Case 4	Single class	1st definition	Weighted vote
Case 5	Single class	2nd definition	Single winner
Case 6	Single class	2nd definition	Weighted vote
Case 7	Multiple classes	1st definition	Single winner
Case 8	Multiple classes	1st definition	Weighted vote



Case 1: 90.10%

Case 2: 89.78%

Case 3: 94.59%

Case 4: 93.34%

Case 5: 98.97%

Case 6: 99.42%

Case 7: 94.59%

Case 8: 99.42%

Fig. 5. Simulation results for the two-dimensional pattern classification problem with $\theta = 0.05$.

In Fig. 5, good results were obtained from the second definition of rule weights (i.e., Case 5 and Case 6). Good results were not obtained from fuzzy rules with no rule weights (i.e., Case 1 and Case

2). This is because fuzzy rules with no rule weights cannot adjust the estimated class boundary. We can also see that simulation results by the first definition of rule weights (i.e., Case 3 and Case 4) are inferior to those by the second definition (i.e., Case 5 and Case 6). On the other hand, when we used the weighted vote method for fuzzy rules with multiple consequent classes, good results were obtained from the first definition (i.e., Case 8 in Fig. 5). These observations suggest that the choice of a weight specification method strongly depends on the type of fuzzy rules and the fuzzy reasoning method.

IV. COMPUTER SIMULATIONS ON REAL-WORLD PROBLEMS

A. Data Sets and Simulation Conditions

In this section, we compare the four definitions of rule weights with one another through computer simulations on wine data and glass data. These data sets are available from the UC Irvine machine learning database (http://www.ics.uci.edu/~mlearn/MLSummary.html). The wine data set is a 13-dimensional problem with 178 samples from three classes. We chose this data set because it involves many continuous attributes. The glass data set is a nine-dimensional problem with 214 samples from six classes. We chose this data set because it involves many classes.

In our computer simulations, we normalized each attribute value into a real number in the unit interval $[0, 1]$. For calculating the average classification rate on test patterns, we used the leaving-one-out (LV1) technique where the given samples were divided in a single test pattern and the other training patterns. A fuzzy rule-based system designed from training patterns was evaluated by a single test pattern. The design-and-test procedure was iterated so that all the given samples were used as test patterns once. Thus the number of iterations of the design-and-test procedure was the same as the number of the given samples in each data set.

In real-world applications of fuzzy rule-based systems, an appropriate fuzzy partition of each continuous attribute is not always given. In our computer simulations, we used four fuzzy partitions in Fig. 6 because we did not know an appropriate granularity of the fuzzy partition for each attribute. In addition to the 14 fuzzy sets in Fig. 6, we also used "*don't care*" as an additional antecedent fuzzy set. The shape of the membership function of *don't care* is the same as the unit interval (i.e., $\mu_{don't\ care}(x) = 1$ for $x \in [0, 1]$). The total number of combinations of antecedent fuzzy sets is 15^n for an *n*-dimensional pattern classification problem.

It is impractical to use all fuzzy rules corresponding to such a huge number of combinations of antecedent fuzzy sets. In our computer simulations, we designed fuzzy rule-based systems in the following manner. First we generated short fuzzy rules of length three or less. Note that the rule length is defined by the number of antecedent conditions (excluding *don't care* conditions). The generated fuzzy rules were divided into *M* groups according to their consequent classes (*M* is three in the wine data, and six in the glass data). Fuzzy rules in each group were sorted in a descending order of a rule selection criterion. We used the product of the confidence $c(\cdot)$ and the support $s(\cdot)$ as the rule selection criterion. When multiple fuzzy rules had the same value of the rule selection criterion, they were randomly sorted (i.e., random tiebreak). We constructed a fuzzy rule-based system by choosing the first N fuzzy rules from each group (i.e., $M \cdot N$ rules in total). Using various values of N (i.e., $N = 1,2,...,5$), we examined the classification performance of fuzzy rule-based systems with different sizes.

We examined the four definitions of rule weights. We also examined the case of no rule weights. This case was simulated by assigning the same rule weight to all the generated fuzzy rules (i.e., $CF_q = 1$ for $\forall q$). For comparison, we also examined the classification performance of fuzzy rules with multiple consequent classes. When we constructed a fuzzy rule-based system from those fuzzy rules, we used the selected $M \cdot N$ fuzzy rules with a single consequent class whose rule weights were specified by the first definition. Each of the selected $M \cdot N$ fuzzy rules with a single consequent class was extended to a fuzzy rule with multiple consequent classes by assigning a certainty grade to each of the M classes in the consequent part. In our computer simulations, we used the single winner method and the weighted vote method.

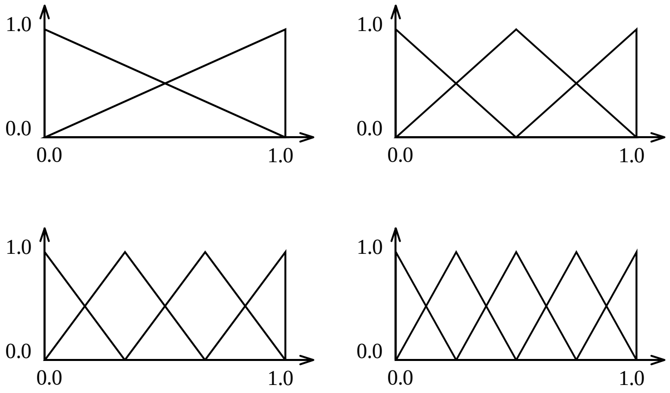


Fig. 6. Four fuzzy partitions used in our computer simulations.

B. Simulation Results on Wine Data

In Table 2, we show average classification rates on test patterns of the wine data set when the single winner method was used. The best result in each row (i.e., each specification of the number of fuzzy rules) is shown by boldface. The best result in the table is indicated by "*". From this table, we can see that the classification performance of fuzzy rule-based systems with no rule weights was improved in some cases by the use of rule weights. We can also see that the difference in the classification performance among the four definitions is small. On the other hand, simulation results by the weighted vote method are shown in Table 3. In this case, good results were obtained independent of the choice of a rule weight specification method. From the comparison between Table

2 and Table 3, we can see that the classification ability of fuzzy rule-based systems was improved by the use of the weighted vote method independent of the choice of a rule weight specification method.

Table 2. Classification rates on test patterns of the wine data set by the single winner method.

Number of rules	No rule weights	First def.	Second def.	Third def.	Fourth def.	Rules in (2)
3	**89.89**	**89.89**	**89.89**	89.33	89.33	**89.89**
6	80.34	83.15	**85.96**	84.83	85.39	83.15
9	88.76	91.57	92.13	**93.26**	**93.26**	91.57
12	**93.26**	**93.26**	92.70	**93.26**	**93.26**	**93.26**
15	88.76	91.57	91.57	**94.38***	93.26	91.57

Table 3. Classification rates on test patterns of the wine data set by the weighted vote method.

Number of rules	No rule weights	First def.	Second def.	Third def.	Fourth def.	Rules in (2)
3	**89.89**	**89.89**	**89.89**	89.33	89.33	**89.89**
6	87.08	87.64	88.76	89.33	88.76	**89.89**
9	93.82	93.26	93.26	**94.38**	93.82	**94.38**
12	94.38	**94.94**	94.38	94.38	93.26	94.38
15	**95.51***	**95.51***	94.38	94.38	93.82	94.94

C. Simulation Results on Glass Data

Average classification rates on test patterns of the glass data set are shown in Table 4 for the single winner method and Table 5 for the weighted vote method. The classification performance of fuzzy rule-based systems strongly depends on the choice of a rule weight definition in these tables. Good results were obtained only when we used the fourth definition of rule weights. Since the glass data set involves six classes (i.e., $M = 6$), the difference among the four definitions is large. As a result, we obtained very different simulation results from the four definitions.

Table 4. Classification rates on test patterns of the glass data set by the single winner method.

Number of rules	No rule weights	First def.	Second def.	Third def.	Fourth def.	Rules in (2)
6	45.79	49.53	45.79	39.25	**58.88**	49.53
12	45.33	48.60	45.79	39.72	**67.76***	48.60
18	45.33	48.60	45.79	39.72	**66.82**	48.60
24	45.33	48.60	45.33	40.19	**65.89**	48.60
30	39.72	48.13	45.33	40.19	**54.21**	48.13

Table 5. Classification rates on test patterns of the glass data set by the weighted vote method.

Number of rules	No rule weights	First Def.	Second Def.	Third Def.	Fourth Def.	Rules in (2)
6	45.79	49.53	45.79	39.25	**58.88**	42.99
12	45.33	48.60	46.26	39.25	**67.76**	43.93
18	45.33	47.20	47.20	40.19	**68.22***	43.93
24	45.33	47.20	48.60	40.19	**68.22***	43.93
30	45.79	46.73	47.20	42.06	**66.36**	44.86

D. Discussions on Simulation Results

Simulation results in Tables 2-5 show that the effect of rule weights is problem-dependent. In computer simulations on the wine data set, the choice of a rule weight specification method did not have a large effect on the classification performance of fuzzy rule-based systems. One reason for this insensitivity is that the wine data set does not have large overlap regions between different classes in the pattern space. On the other hand, the performance of fuzzy rule-based systems strongly depended on the choice of a rule weight specification method in computer simulations on the glass data set.

It should be noted that good results were obtained in Table 4 and Table 5 only from the fourth definition. One may think that the difference in classification rates between the fourth definition and the other definitions is suspiciously large in those tables. We examined why good results were obtained only from the fourth definition. Then we found that different fuzzy rules were used in the case of the fourth definition. As we have already mentioned, CF_q^{IV} by the fourth definition can be negative while rule weights by the other definitions are always positive. More specifically CF_q^{IV} becomes negative when CF_q^{I} is smaller than 0.5. Since we did not use fuzzy rules with negative rule weights, fuzzy rules with $CF_q^{\mathrm{I}} < 0.5$ were not used in the case of the fourth definition while they were used in the other definitions. In computer simulations in Table 4, $CF_q^{\mathrm{I}} < 0.5$ holds for 2800 rules among $30 \times 214 = 6420$ rules generated by the first definition during the execution of the LV1 procedure. This means that the used fuzzy rules were different between the fourth definition and the other definitions in computer simulations on the glass data set. On the other hand, $CF_q^{\mathrm{I}} < 0.5$ holds for no rule among $15 \times 178 = 2670$ rules generated by the first definition in Table 2 for the wine data set.

For evaluating the pure effect of rule weights on the classification performance of fuzzy rule-based systems for the glass data set, we executed additional computer simulations. We first generated a pre-specified number of fuzzy rules using the fourth definition as in Table 4. Then only the rule weight of each fuzzy rule was modified using each definition. In this manner, we generated the same fuzzy rules (with different rule weights) for all definitions. Simulation results are summarized in Table 6. We can see from this table that the best results were still obtained from the fourth definition while the difference in classification rates among the four definitions was reduced from Table 4.

Table 6. Classification rates on test patterns of the glass data set when the single winner method was used. Same fuzzy rules (except for rule weights) were used in all the six cases in this table.

Number of rules	No rule weights	First Def.	Second Def.	Third Def.	Fourth Def.	Rules in (2)
6	**60.75**	**60.75**	60.28	60.28	58.88	**60.75**
12	61.22	61.68	61.22	56.54	**67.76***	61.68
18	60.75	61.68	61.68	56.54	**66.82**	61.68
24	60.75	61.68	61.68	56.07	**65.89**	61.68
30	58.88	**59.81**	59.35	55.61	54.21	**59.81**

V. CONCLUSION

In this paper, we compared four heuristic specification methods of rule weights with one another. We also compared two fuzzy reasoning methods with each other: the single winner method and the weighted vote method. We demonstrated through computer simulations on artificial test problems that the first definition of rule weights (i.e., the direct use of the fuzzy conditional probability) is not appropriate for the single winner method while it is appropriate for the weighted vote method. We also demonstrated that the second definition of rule weights, which was used in our former studies [10], [13], [14], is not appropriate for the case of multi-class problems with many classes. In this case, the fourth definition worked well. While the differences in classification rates among the four definitions were not large in their application to the wine data set with three classes, the superiority of the fourth definition was clear in their application to the glass data set with six classes. Simulation results also showed that better results were obtained by the weighted vote method than the single winner method in many cases. The single winner method, however, is preferable from the viewpoint of the comprehensibility of the classification result of each pattern. The use of the single winner method also makes it easy to decrease the number of fuzzy rules because only a single winner rule is responsible for the classification of each pattern.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] H. Ishibuchi and T. Nakashima, “Effect of rule weights in fuzzy rule-based classification systems,” *IEEE Trans. on Fuzzy Systems*, vol. 9, no. 4, pp. 506-515, August 2001.

[2] O. Cordon, M. J. del Jesus, and F. Herrera, "A proposal on reasoning methods in fuzzy rule-based classification systems," *International Journal of Approximate Reasoning*, vol. 20, no. 1, pp. 21-45, January 1999.

[3] J. van den Berg, U. Kaymak, and W. -M. van den Bergh, "Fuzzy classification using probability based rule weighting," *Proc. of 11th IEEE International Conference on Fuzzy Systems*, pp. 991-996, Honolulu, May 2002.

[4] D. Nauck and R. Kruse, "A neuro-fuzzy method to learn fuzzy classification rules from data," *Fuzzy Sets and Systems*, vol. 89, no. 3, pp. 277-288, August 1997.

[5] S. Abe and R. Thawonmas, "A fuzzy classifier with ellipsoidal regions," *IEEE Trans. on Fuzzy Systems*, vol. 5, no. 3, pp. 358-368, August 1997.

[6] M. Setnes and H. Roubos, "GA-based modeling and classification: Complexity and performance," *IEEE Trans. on Fuzzy Systems*, vol. 8, no. 5, pp. 509-522, October 2000.

[7] L. Castillo, A. Gonzalez, and R. Perez, "Including a simplicity criterion in the selection of the best rule in a genetic fuzzy learning algorithm," *Fuzzy Sets and Systems*, vol. 120, no. 2, pp. 309-321, June 2001.

[8] D. Nauck and R. Kruse, "How the learning of rule weights affects the interpretability of fuzzy systems," *Proc. of 7th IEEE International Conference on Fuzzy Systems*, pp. 1235-1240, Anchorage, May 1998.

[9] F. Klawonn and A. Keller, "Fuzzy clustering and fuzzy rules," *Proc. of 7th International Fuzzy Systems Association World Congress*, vol. I, pp. 193-198, Prague, June 1997.

[10] H. Ishibuchi, K. Nozaki, and H. Tanaka, "Distributed representation of fuzzy rules and its application to pattern classification," *Fuzzy Sets and Systems*, vol. 52, no. 1, pp. 21-32, November 1992.

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

[12] T. -P. Hong, C. -S. Kuo, and S. -C. Chi, "Trade-off between computation time and number of rules for fuzzy mining from quantitative data," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 9, no. 5, pp. 587-604, October 2001.

[13] H. Ishibuchi, T. Yamamoto, and T. Nakashima, "Fuzzy data mining: Effect of fuzzy discretization," *Proc. of 1st IEEE International Conference on Data Mining*, pp. 241-248, San Jose, November 2001.

[14] H. Ishibuchi, T. Nakashima, and T. Morisawa, "Voting in fuzzy rule-based systems for pattern classification problems," *Fuzzy Sets and Systems*, vol. 103, no. 2, pp. 223-238, April 1999.

[15] A. Nürnberger, C. Borgelt, and A. Klose, "Improving naive Bayes classifiers using neuro-fuzzy learning," *Proc. of International Conference on Neural Information Processing*, pp. 154-159, Perth, November 1999.

Hisao Ishibuchi (M’93) received the B.S. and M.S. Degrees in precision mechanics from Kyoto University, Kyoto, Japan, in 1985 and 1987, respectively, and the Ph.D. degree from Osaka Prefecture University, Osaka, Japan, in 1992.

Since 1987, he has been with Department of Industrial Engineering, Osaka Prefecture University, where he is currently a Professor. He was a Visiting Research Associate at the University of Toronto, Toronto, ON, Canada, from August 1994 to March 1995 and from July 1997 to March 1998. His research interests include fuzzy rule-based classification systems, evolutionary multiobjective optimization, and data mining.

Dr. Ishibuchi received the best paper award at the GECCO 2004 conference. He is currently an Associate Editor for *IEEE Trans. on Systems, Man, and Cybernetics - Part B* and for *Mathware and Soft Computing*. He is also a fuzzy technical committee member of the IEEE Computational Intelligence Society.

Takashi Yamamoto (SM’01) received the B.S., M.S. and Ph.D. Degrees in industrial engineering from Osaka Prefecture University, Osaka, Japan, in 2001, 2003 and 2004, respectively. Since 2004, he has been with NTT DoCoMo, Inc., Tokyo, Japan.

Dr. Yamamoto received the best student paper award at the 4th International Conference on Intelligent Technologies (InTech 2004, Chang Mai, Thailand).