Archives of Control Sciences
Volume 12(XLVIII), 2002
No. 4, *pages 351–377*

Effect of fuzzy discretization in fuzzy rule-based systems for classification problems with continuous attributes

HISAO ISHIBUCHI and TAKASHI YAMAMOTO

Continuous attributes are usually discretized into intervals in machine learning and data mining. Our knowledge representation is, however, not always based on such discretization. For example, we usually use linguistic terms (e.g., *young*, *middle-aged*, and *old*) for dividing our ages into some categories with fuzzy boundaries. In this paper, we examine the effect of fuzzy discretization on the classification performance of fuzzy rule-based systems through computer simulations on simple numerical examples and real-world pattern classification problems. For executing such computer simulations, we introduce a control parameter that specifies the overlap grade between adjacent antecedent fuzzy sets (i.e., linguistic terms) in fuzzy discretization. Interval discretization can be viewed as a special case of fuzzy discretization with no overlap. Computer simulations are performed using fuzzy discretization with various specifications of the overlap grade. Simulation results show that fuzzy rules have high generalization ability even when the domain interval of each continuous attribute is homogeneously partitioned into linguistic terms. On the other hand, generalization ability of rule-based systems strongly depends on the choice of threshold values in the case of interval discretization.

Key words: pattern classification, discretization of continuous attributes, rule extraction, data mining, fuzzy rules, rule weights

1. Introduction

When our knowledge extraction task involves numerical data with continuous attributes, each attribute is usually discretized into intervals [3,4]. The discretization into intervals is used in many machine learning techniques such as decision trees [13]. In some situations, human knowledge exactly corresponds to such discretization of continuous attributes. For example, the domain of our ages is divided into two intervals by the threshold age 20 in the following knowledge: *People under 20 are not allowed to smoke*. In other situations, the discretization into intervals is not appropriate for describing human knowledge. For example, we may have the following knowledge: *Tall people are*

The Authors are with Department of Industrial Engineering, Osaka Prefecture University 1-1 Gakuen-cho, Sakai, Osaka 599-8531, Japan. The Author to whom correspondence should be addressed is Professor Ishibuchi, e-mail: hisaoi@ie.osakafu-u.ac.jp tel.: +81-72-254-9350 fax: +81-72-254-9915

Received 26.10.2001, revised 15.03.2002.

not comfortable in small cars. We cannot appropriately represent this knowledge using the discretization of the domain of our height into intervals. This is because the linguistic term *tall* cannot be appropriately represented by an interval. A mathematical framework for representing linguistic terms is fuzzy logic. Fuzzy logic has been recognized as a convenient tool for handling continuous attributes by rule-based systems in a human understandable manner [14]. This recognition is supported by many successful applications of fuzzy control methods [11]. An example of the membership function of *tall* is shown in Fig. 1. In this paper, we discuss fuzzy rule extraction for pattern classification problems with continuous attributes. We use fuzzy rules of the following type for pattern classification problems with n attributes:

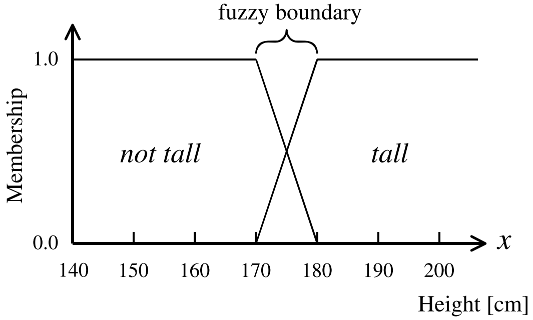


Figure 1. An example of fuzzy discretization for height

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

where $\boldsymbol{x} = (x_1, \ldots, x_n)$ is an n-dimensional pattern vector, A_{qi} is an antecedent fuzzy set for the i-th attribute, C_q is a class label, and CF_q is a certainty grade (i.e., rule weight). The effect of rule weights on the performance of fuzzy rule-based systems was examined in Ishibuchi, Nakashima [6]. The above-mentioned linguistic knowledge on the comfortableness in small cars can be represented in the form of (1) as „If x is *tall* then Class 2” where x is the height and Class 2 is the class label corresponding to *not comfortable*. This fuzzy rule may be obtained from experimental results where a number of examinees are asked whether they feel comfortable or not in a small car. Suppose that we have responses in Table 1 from ten examinees on the comfortableness in a small car. From Fig. 1 and Table 1, we can extract two linguistic rules „If x is *not tall* then Class 1 (i.e., *comfortable*)” and „If x is *tall* then Class 2 (i.e., *not comfortable*)”. These rules are much more intuitive than interval representation rules such as „If $x \leqslant 175$ then Class 1” and „If $x > 175$ then Class 2”.

Table 1. Responses from ten examinees (artificial data for illustration purpose).

Examinee (p)	1	2	3	4	5	6	7	8	9	10
Height (x_p)	150	158	161	173	174	176	177	178	185	191
Comfortableness	Yes	Yes	Yes	No	Yes	No	Yes	No	No	No

The fuzzy rule in (1) can be viewed as a fuzzy association rule $\boldsymbol{A}_q \Rightarrow C_q$ where $\boldsymbol{A}_q = (A_{q1}, \ldots, A_{qn})$. The confidence and the support of the fuzzy association rule $\boldsymbol{A}_q \Rightarrow C_q$ are defined by extending their standard definitions [1] to the case of fuzzy discretization of each continuous attribute [5,10]. In this paper, we first describe fuzzy rule generation from numerical data using these two concepts in data mining: confidence and support. We also explain pattern classification by fuzzy rules. Next we illustrate some effects of fuzzy discretization in the fuzzy rule generation phase and the pattern classification phase using simple numerical examples. Then we examine each effect by computer simulations on well-known and frequently used pattern classification problems in the literature: iris data and wine data. In our computer simulations, we examine the classification performance of fuzzy rule-based systems generated from fuzzy discretization with various overlap grades between adjacent antecedent fuzzy sets. Interval discretization can be viewed as a special case of fuzzy discretization with no overlap. Simulation results show that fuzzy rule-based systems have high generalization ability even when the domain interval of each continuous attribute is homogeneously partitioned into antecedent fuzzy sets. On the other hand, generalization ability of rule-based systems strongly depends on the choice of threshold values in the case of interval discretization.

2. Fuzzy rules for pattern classification

2.1. Two measures for fuzzy rule evaluation

Let us assume that we have m labelled patterns $\boldsymbol{x}_p = (x_{p1}, \ldots, x_{pn})$, $p = 1, 2, \ldots, m$ from M classes. We denote the set of these training patterns as $D = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}$. The cardinality of D is m (i.e., $|D| = m$). As in our former studies on fuzzy rule-based classification systems [6-10], we define the compatibility grade of each training pattern $\boldsymbol{x}_p$ with the antecedent $\boldsymbol{A}_q$ using the product operator as

$$\mu_{A_q}(\boldsymbol{x}_p) = \mu_{A_{q1}}(x_{p1}) \times \cdots \times \mu_{A_{qn}}(x_{pm}), \tag{2}$$

where $\mu_{A_qi}(\cdot)$ is the membership function of the antecedent fuzzy set A_{qi}, and $\times$ means the multiplication of real numbers. Of course, the minimum operator $\wedge$ can be used instead of the product operator $\times$ in (2). Let $D(\boldsymbol{A}_q)$ be the fuzzy set of compatible training patterns with the antecedent $\boldsymbol{A}_q$. Then the total compatibility grade with the antecedent $\boldsymbol{A}_q$ is calculated as

$$|D(\boldsymbol{A}_q)| = \sum_{p=1}^{M} \mu_{A_q}(\boldsymbol{x}_p). \tag{3}$$

In this formulation, $|D(\boldsymbol{A}_q)|$ is the cardinality of the fuzzy set $D(\boldsymbol{A}_q)$. The cardinality of a fuzzy set is often called the sigma count in the literature. Note that $|D(\boldsymbol{A}_q)|$ is the number of compatible training patterns with $\boldsymbol{A}_q$ in the case of interval discretization (i.e., the cardinality of the non-fuzzy set $D(\boldsymbol{A}_q)$ in the case where A_{qi} is an interval).

Using (2), we define the compatibility grade of each training pattern $\boldsymbol{x}_p$ with the fuzzy rule R_p (i.e., with both the antecedent $\boldsymbol{A}_q$ and the consequent C_q) as

$$\mu_{R_q}(\boldsymbol{x}_p) = \begin{cases} \mu_{A_p}(\boldsymbol{x}_p), & \text{if} \quad p \in \text{Class } C_q \\ 0, & \text{if} \quad p \notin \text{Class } C_q \end{cases} \tag{4}$$

Let $D(\boldsymbol{A}_q) \cap D(C_q)$ be the fuzzy set of compatible training patterns with both $\boldsymbol{A}_p$ and C_q. Then the total compatibility grade with both $\boldsymbol{A}_p$ and C_q is calculated as

$$|D(\boldsymbol{A}_q) \cap D(C_q)| = \sum_{p=1}^{m} \mu_{R_q}(\boldsymbol{x}_p) = \sum_{p \in \text{Class } C} \mu_{A_q}(\boldsymbol{x}_p). \tag{5}$$

Note that $|D(\boldsymbol{A}_q) \bigcap D(C_q)|$ is the number of compatible training patterns with both $\boldsymbol{A}_p$ and C_q in the case of interval discretization.

In the field of data mining, two measures are often used for evaluating each association rule [1]. They are *confidence* and *support*. The confidence $c(\boldsymbol{A}_q) \Rightarrow C_q)$ of the association rule $\boldsymbol{A}_q \Rightarrow C_q$ is defined as follows [1,5,10]:

$$c(\boldsymbol{A}_q \Rightarrow C_q) = \frac{|D(\boldsymbol{A}_q) \bigcap D(C_q)|}{|D(\boldsymbol{A}_q)|} = \frac{\sum_{p \in \text{Class } C_q} \mu_{A_q}(\boldsymbol{x}_p)}{\sum_{p=1}^{m} \mu_{A_q}(\boldsymbol{x}_p)}. \tag{6}$$

The confidence $c(\boldsymbol{A}_q \Rightarrow C_q)$ is the ratio of compatible patterns with both the antecedent $\boldsymbol{A}_q$ and the consequent C_q to compatible patterns with the antecedent $\boldsymbol{A}_q$. The confidence $c(\boldsymbol{A}_q \Rightarrow C_q)$ measures the validity of the association rule $\boldsymbol{A}_q \Rightarrow C_q$. Note that the definition of the confidence $c(\boldsymbol{A}_q \Rightarrow C_q)$ in (6) can be used for fuzzy discretization as well as interval discretization.

On the other hand, the support $s(\boldsymbol{A}_q \Rightarrow C_q)$ of the association rule $\boldsymbol{A}_q \Rightarrow C_q$ is defined as follows [1,5,10]:

$$s(\boldsymbol{A}_q \Rightarrow C_q) = \frac{|D(\boldsymbol{A}_q) \cap D(C_q)|}{|D|} = \frac{\sum_{p \in \text{Class } C_q} \mu_{A_q}(\boldsymbol{x}_p)}{m}. \tag{7}$$

The support $s(\boldsymbol{A}_q \Rightarrow C_q)$ is the ratio of compatible patterns with both the antecedent $\boldsymbol{A}_q$ and the consequent C_q to the given m training patterns. The support $s(\boldsymbol{A}_q \Rightarrow C_q)$ measures the coverage of training patterns by the association rule $\boldsymbol{A}_q \Rightarrow C_q$. The definition of the support $s(\boldsymbol{A}_q \Rightarrow C_q)$ in (7) can be used for fuzzy discretization as well as interval discretization.

As an example, let us calculate $c(tall \Rightarrow \text{Class } 2)$ and $s(tall \Rightarrow \text{Class } 2)$ from the ten training patterns in Table 1 where Class 2 corresponds to *not comfortable* (i.e., „No” in Table 1). The membership function of the linguistic term *tall* in Fig. 1 is written as

$$\mu_{tall}(x) = \begin{cases} 0, & \text{if} \quad x \leqslant 170, \\ (x - 170)/10 & \text{if} \quad 170 < x < 180, \\ 1, & \text{if} \quad 180 \leqslant x. \end{cases} \tag{8}$$

The fuzzy set of examinees compatible with *tall* in Table 1 is explicitly written as

$$D(tall) = \left\{ \frac{0.0}{150}, \frac{0.0}{158}, \frac{0.0}{161}, \frac{0.3}{173}, \frac{0.4}{174}, \frac{0.6}{176}, \frac{0.7}{177}, \frac{0.8}{178}, \frac{1.0}{185}, \frac{1.0}{191} \right\} \tag{9}$$

where the denominator shows the height x_p of each examinee and the numerator shows its membership value $\mu_{tall}(x_p)$. Each element in (9) should not be viewed as a fraction number but a pair of x_p and $\mu_{tall}(x_p)$. The total compatibility grade with *tall* is calculated from (9) as

$$|D(tall)| = 0.0 + 0.0 + 0.0 + 0.3 + \cdots + 1.0 = 4.8. \tag{10}$$

From Table 1, the total compatibility grade $|D(tall) \cap D(\text{Class 2})|$ with both *tall* and Class 2 is calculated as

$$|D(tall) \cap D(\text{Class 2})| = 0.3 + 0.6 + 0.8 + 1.0 + 1.0 = 3.7 \tag{11}$$

Thus the confidence and the support are calculated as

$$c(tall) \Rightarrow \text{Class 2}) = \frac{3.7}{4.8} = 0.77, \tag{12}$$

$$s(tall) \Rightarrow \text{Class 2}) = \frac{3.7}{10} = 0.37. \tag{13}$$

In the same manner, the confidence and the support of the fuzzy rule „*tall* ⇒ Class 1" are calculated as

$$c(tall) \Rightarrow \text{Class 1}) = \frac{1.1}{4.8} = 0.23, \tag{14}$$

$$s(tall) \Rightarrow \text{Class 2}) = \frac{1.1}{10} = 0.11. \tag{15}$$

Thus we choose the linguistic association rule „*tall* ⇒ Class 2" rather than „*tall* ⇒ Class 1". We can also generate another linguistic association rule „*not tall* ⇒ Class 1" The confidence and the support of this linguistic rule are calculated as

$$c(not\ tall) \Rightarrow \text{Class 1}) = \frac{3.9}{5.2} = 0.75, \tag{16}$$

$$s(not\ tall) \Rightarrow \text{Class 1}) = \frac{3.9}{10} = 0.39. \tag{17}$$

In Fig. 1, the sum of the membership functions of the two fuzzy sets *tall* and *not tall* is always unity for any values of the height x. Thus $|D(tall)| + |D(not\ tall)|$ is equal to the total number of the examinees in Table 1 (i.e., ten). When $\mu_{tall}(x) + \mu_{not\ tall}(x) = 1$ does not hold, $|D(tall)| + |D(not\ tall) = 10|$ does not hold either. Even in this case, we can use the definitions of the confidence in (6) and the support in (7) with no modification.

2.2. Fuzzy rule generation

As shown in the previous subsection, it is natural to choose the consequent C_q with the maximum confidence for the antecedent $\boldsymbol{A}_q$:

$$c(\boldsymbol{A}_q \Rightarrow C_q) = max\{c(\boldsymbol{A}_q \Rightarrow \text{Class } h) | h = 1, 2, \ldots, M\}. \tag{18}$$

Note that the same consequent C_q is obtained if we use the support $s(\cdot)$ instead of the confidence $c(\cdot)$ in (18). When multiple classes have the maximum confidence (i.e., when C_q cannot be uniquely specified), we do not generate any fuzzy rule with the antecedent $\boldsymbol{A}_q$. Similar heuristic rule generation methods were proposed for function approximation in [12,15].

The antecedent $\boldsymbol{A}_q$ is constructed by combining antecedent fuzzy sets for n attributes. For low-dimensional pattern classification problems, it is possible to generate fuzzy rules corresponding to all combinations of antecedent fuzzy sets. Let us consider a two-dimensional pattern classification problem in Fig. 2. When each axis of the pattern space is divided into three linguistic terms (i.e., three antecedent fuzzy sets) as shown in Fig. 2, nine fuzzy rules are generated because a single fuzzy rule is generated in each fuzzy subspace of the pattern space. In general, let K_i be the number of antecedent fuzzy sets on the i-th attribute. Then the total number of combinations of antecedent fuzzy sets for n attributes is $K_1 \times K_2 \times \cdots \times K_n$. For high-dimensional problems, it is impractical to examine all the combinations for generating fuzzy rules. In our computer simulations in this paper, we only generate short fuzzy rules with a small number of antecedent conditions. In this paper, the number of antecedent conditions is referred to as the rule length. Thus the number of antecedent conditions in short fuzzy rules is small. Short fuzzy rules can be viewed as having many *don't care* conditions. Fuzzy rule-based systems can be applied to high-dimensional problems when we use only short fuzzy rules with a few antecedent conditions. This is because the number of short fuzzy rules is much smaller than that of long fuzzy rules with many antecedent conditions.

The confidence $c(\boldsymbol{A}_q \Rightarrow C_q)$ can be used as the certainty grade CF_q of the fuzzy rule $\boldsymbol{A}_q \Rightarrow C_q$ as in Cordon et al.[2]. It was shown in [10] that better results were obtained from the following definition than the direct use of the confidence $c(\boldsymbol{A}_q \Rightarrow C_q)$.

$$CF_q = c(\boldsymbol{A}_q \Rightarrow C_q) - \bar{c}, \tag{19}$$

where $\bar{c}$ is the average confidence over association rules with the same antecedent $\boldsymbol{A}_q$ but different consequent classes:

$$\bar{c} = \frac{1}{M-1} \sum_{\substack{h=1 \\ h \neq C_q}}^{M} c(\boldsymbol{A}_q \Rightarrow \text{Class } h). \tag{20}$$

This definition of CF_q can be easily understood if we consider the case of $M = 2$ (i.e., two-class pattern classification problems). In this case, CF_q is calculated as

$$CF_q = c(\boldsymbol{A}_q \Rightarrow \text{Class 1}) - c(\boldsymbol{A}_q \Rightarrow \text{Class 2}), \tag{21}$$

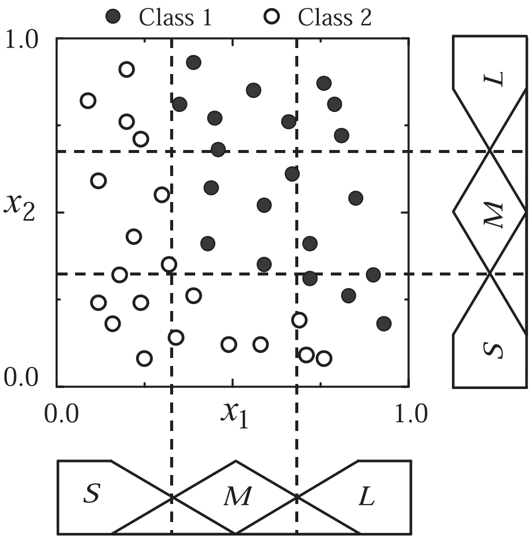


Figure 2. Training patterns and three antecedent fuzzy sets (S: *small*, M: *medium*, and L: *large*)

when

$$c(\boldsymbol{A}_q \Rightarrow \text{Class 1}) > c(\boldsymbol{A}_q \Rightarrow \text{Class 2}). \tag{22}$$

In Fig. 3, we show the consequent class and the certainty grade of each fuzzy rule generated from training patterns and fuzzy discretization in Fig. 2. As shown in (18), the majority class in each fuzzy subspace is chosen as the consequent class of the corresponding fuzzy rule. The certainty grade CF_q shows the purity of compatible training patterns in each fuzzy subspace. If a fuzzy subspace includes training patterns from only a single class, the corresponding fuzzy rule has the maximum certainty grade (i.e., $CF_q = 1$). On the other hand, when a fuzzy subspace includes training patterns from different classes, the certainty grade is less than its maximum value (i.e., $CF_q < 1$). For example, the certainty grade is very small as the right-bottom fuzzy rule in Fig. 3 when the number of training patterns from the majority class in a fuzzy subspace is almost the same as that from the minority class in two-class problems.

2.3. Pattern classification using fuzzy rules

Let S be a set of fuzzy rules of the form in (1). The rule set S can be viewed as a fuzzy rule-based classification system. We use a single winner rule method [7] for classifying new patterns by the rule set S. See [2,7] for other fuzzy reasoning methods for pattern classification. The single winner rule R_w is determined for a new pattern $\boldsymbol{x}_p = (x_{p1}, \ldots, x_{pn})$ as

$$\mu_{\boldsymbol{A}_w}(\boldsymbol{x}_p) \times CF_w = Max\{\mu_{\boldsymbol{A}_q}(\boldsymbol{x}_p) \times CF_q | R_q \in S\} \tag{23}$$

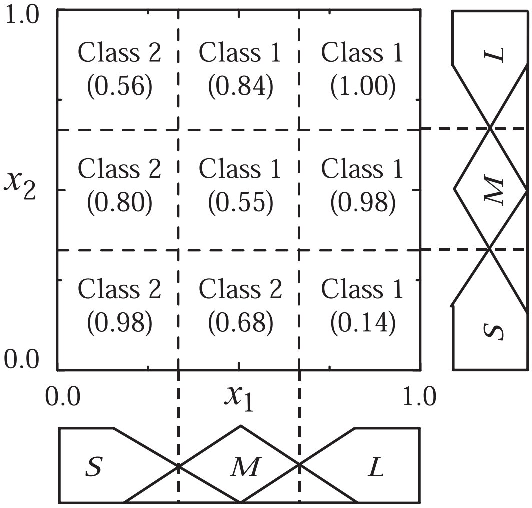


Figure 3. Nine fuzzy rules generated from Fig. 2

That is, the winner rule has the maximum product of the compatibility grade and the certainty grade. If multiple fuzzy rules with different consequent classes have the same maximum product for the new pattern $\boldsymbol{x}_p$, the classification of $\boldsymbol{x}_p$ is rejected. The classification is also rejected if no fuzzy rule is compatible with the new pattern $\boldsymbol{x}_p$.

In Fig. 4, we show classification results by the nine fuzzy rules in Fig. 3. Fig. 4 shows the decision region of each fuzzy rule, which was depicted by examining 501×501 points in the two-dimensional input space. In general, the larger the certainty grade of a fuzzy rule is, the larger its decision region is. The boundary between decision regions of adjacent fuzzy rules with different consequent classes corresponds to the classification boundary between two classes, which is depicted by bold lines in Fig. 4. Note that $\mu_{\boldsymbol{A}_q}(\boldsymbol{x}) \times CF_q = \mu_{\boldsymbol{A}_r}(\boldsymbol{x}) \times CF_r$ holds from (23) on the boundary between the decision regions of the adjacent two fuzzy rules R_q and R_r. As we can see from Fig. 4, the classification boundary is not always parallel to each axis of the pattern space. When all fuzzy rules have the same certainty grade in Fig. 4, the classification boundary is always parallel to each axis. This is because the decision region of each fuzzy rule is the square, which corresponds to the dashed lines in Fig. 3. Complicated decision boundaries can be obtained only when each fuzzy rule has a different certainty grade. See [6] for further discussions on the effect of certainty grades on the classification performance of fuzzy rule-based systems. In the case of high-dimensional problems, we cannot visually depict classification boundaries. The classification of each new pattern, however, can be performed using the single winner rule R_w in (23) in the same manner as Fig. 4.

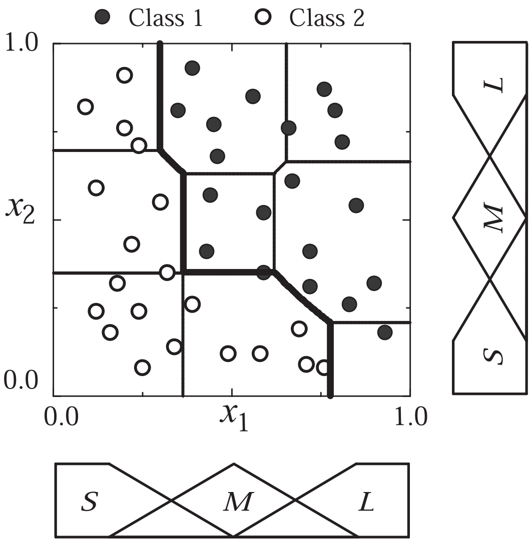


Figure 4. Decision region of each fuzzy rule and the classification boundary

3. Effects of fuzzy discretization

In this section, we illustrate some effects of fuzzy discretization in the rule generation phase and the pattern classification phase using simple numerical examples. The effect of fuzzy discretization on the classification performance of fuzzy rule-based systems will be examined in the next section through computer simulations on real-world pattern classification problems.

3.1. Effects in rule generation phase

The main characteristic feature of fuzzy discretization is the overlap between adjacent antecedent fuzzy sets (i.e., linguistic terms). This means that many fuzzy rules overlap in the pattern space. On the other hand, there is no overlap in the case of interval discretization. This difference is illustrated in Fig. 5. In Fig. 5 (a), an input pattern denoted by the closed circle in the pattern space is covered by four fuzzy rules corresponding to the shaded four squares. In general, an input pattern in the n-dimensional pattern space is covered by 2^n fuzzy rules. On the other hand, it is covered by only a single non-fuzzy rule in the case of interval discretization as shown in Fig. 5 (b).

As shown in Fig. 5 (a), an input pattern in the pattern space is covered by multiple fuzzy rules. This means that each training pattern is involved in the rule generation of multiple fuzzy rules. This effect of fuzzy discretization in the fuzzy rule generation phase is significant in the case of fine partitions and sparse training patterns as shown in Fig. 6. In Fig. 6 (a), the two-dimensional pattern space is divided into 25 fuzzy subspaces by five fuzzy sets on each axis. Almost all the corresponding 25 fuzzy rules can be generated from the given training patterns in Fig. 6 (a). On the other hand, only nine non-fuzzy rules can be generated in the case of interval discretization in Fig. 6 (b).

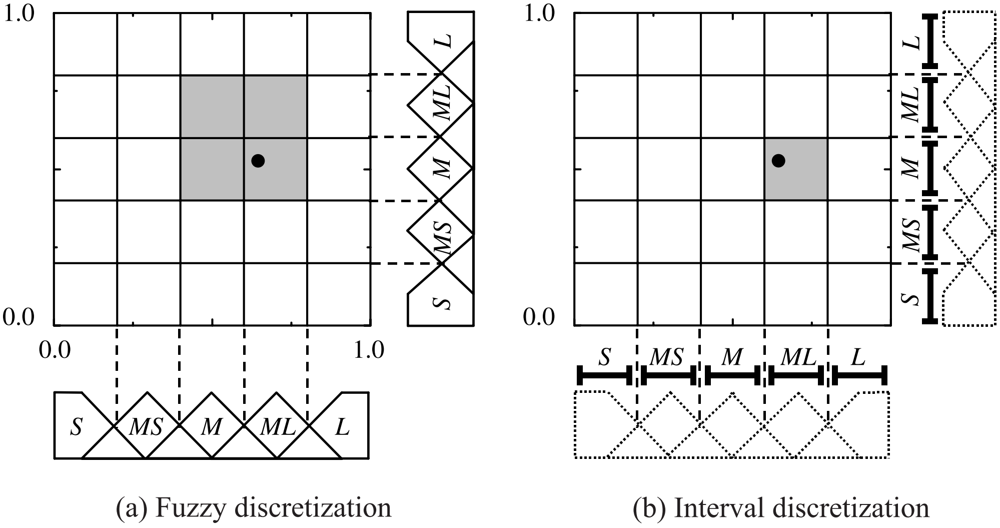


Figure 5. Difference between fuzzy discretization and interval discretization. Each axis is homogeneously divided into five linguistic terms (S: *small*, MS: *medium small*, M: *medium*, ML: *medium large*, and L: *large*) and five intervals

In addition to the above-mentioned fact that each training pattern is involved in the rule generation of multiple fuzzy rules, it should be noted that each training pattern has a different weight depending on its compatibility grade in the rule generation phase. Let us consider the following fuzzy rule at the center of the pattern space in Fig. 6 (a):

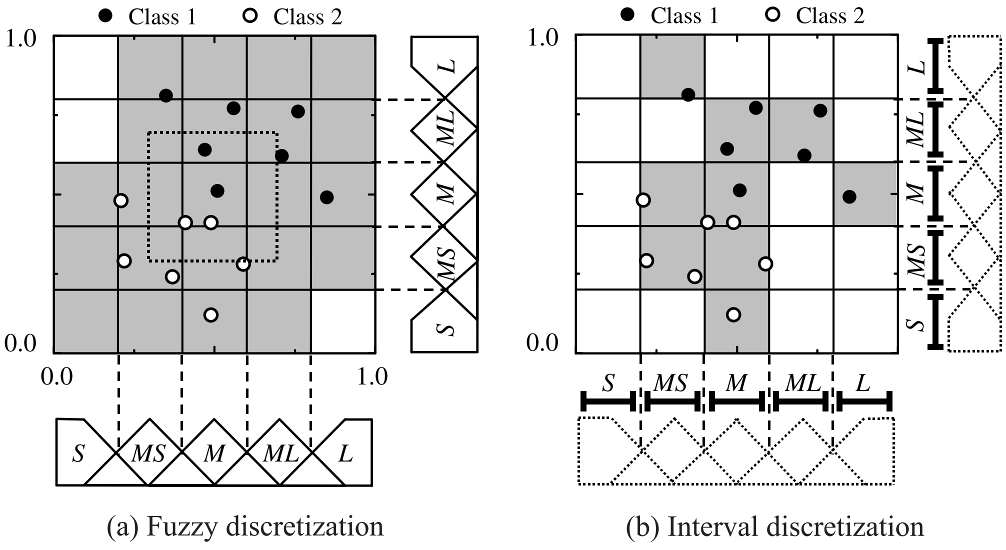
$$\text{If } x_1 \text{ is } \textit{medium} \text{ and } x_2 \text{ is } \textit{medium} \text{ then Class 1 with } CF = 0.17. \quad (24)$$

This fuzzy rule was generated from four training patterns in the large square denoted by dotted lines in Fig. 6 (a). In the rule generation phase of this fuzzy rule, one training pattern from Class 1 near the center of the pattern space has a much larger weight than two training patterns from Class 2 near the sides of the small square at the center of the pattern space. The compatibility grades of the four compatible patterns (i.e., their weights in the rule generation of this fuzzy rule) are 0.90, 0.52, 0.30, and 0.26 (the other pattern from Class 1). The sum of the compatibility grades over the two training patterns from Class 2 is smaller than that of the two Class 1 patterns. As a result, the consequent of the fuzzy rule in (24) is Class 1.

The corresponding non-fuzzy rule was generated in Fig. 6 (b) as

$$\text{If } x_1 \text{ is } \textit{medium} \text{ and } x_2 \text{ is } \textit{medium} \text{ then Class 2 with } CF = 0.33. \quad (25)$$

This non-fuzzy rule was generated from three compatible training patterns that are located in the small square at the center of the pattern space in Fig. 6 (b). Note that the



(a) Fuzzy discretization

(b) Interval discretization

Figure 6. Training patterns and generated fuzzy and non-fuzzy rules

fuzzy rule in (24) and the non-fuzzy rule in (25) have different consequent classes. This is because each compatible training pattern has the same weight in interval discretization while it has a different weight in fuzzy discretization.

3.2. Effects in classification phase

Characteristic features of fuzzy discretization in the classification phase also stem from the fact that multiple fuzzy rules overlap in the pattern space. The classification boundary (i.e., boundary between decision regions of adjacent fuzzy rules with different consequent classes) can be adjusted by changing the certainty grades of those fuzzy rules [6]. That is, the location of the classification boundary is determined by the certainty grade of each fuzzy rule. On the other hand, the location of the classification boundary is determined by the threshold values on each axis in the case of interval discretization as shown in Fig. 7. This is because the decision region of each non-fuzzy rule is uniquely determined by the threshold values on each axis. From these discussions, we expect that good results can be obtained by fuzzy rules even when the fuzzy partition of each axis is not appropriately specified. We also anticipate that an appropriate partition of each axis is necessary for generating non-fuzzy rules with high classification ability in the case of interval discretization.

Each fuzzy rule has a larger decision region than the corresponding non-fuzzy rule when there are no adjacent fuzzy rules. If there are no rules around the fuzzy rule in (24) with the antecedent (*medium*, *medium*), its decision region is the large dotted square in Fig. 6 (a). On the other hand, the decision region of the corresponding non-fuzzy rule is the small square at the center of Fig. 6 (b). In the case of interval discretization, the

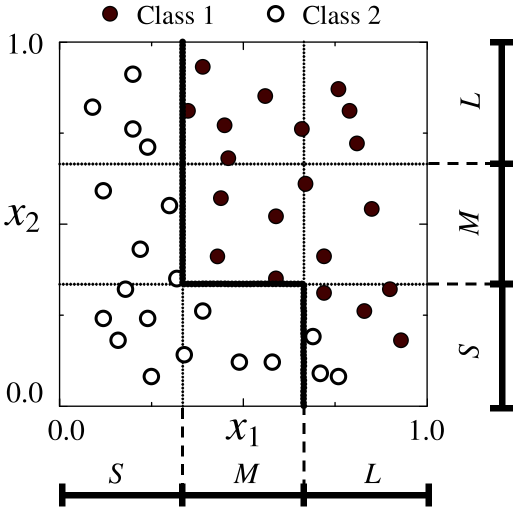


Figure 7. Decision region of each non-fuzzy rule and the classification boundary

size of the decision region is independent of the existence of adjacent rules. The size of the decision region of each fuzzy rule, however, depends on the existence of adjacent rules. Thus a small number of sparsely located fuzzy rules can classify much more patterns than the corresponding non-fuzzy rules. In the case of interval discretization, the classification of many patterns will be rejected if the number of non-fuzzy rules is very small. From these discussions, we expect that a small number of fuzzy rules have higher classification ability than the same number of non-fuzzy rules.

4. Computer simulations using homogeneous discretization

In Section 3, we illustrated some effects of fuzzy discretization in the fuzzy rule generation phase and the pattern classification phase. Main positive effects of fuzzy discretization on the classification performance of fuzzy rule-based systems are summarized as follows:

1. Multiple fuzzy rules can be generated from a single training pattern while only a single rule can be generated in the case of interval discretization. This may lead to better results by fuzzy discretization than interval discretization when the number of training patters is very small (i.e., when training patterns are sparse).

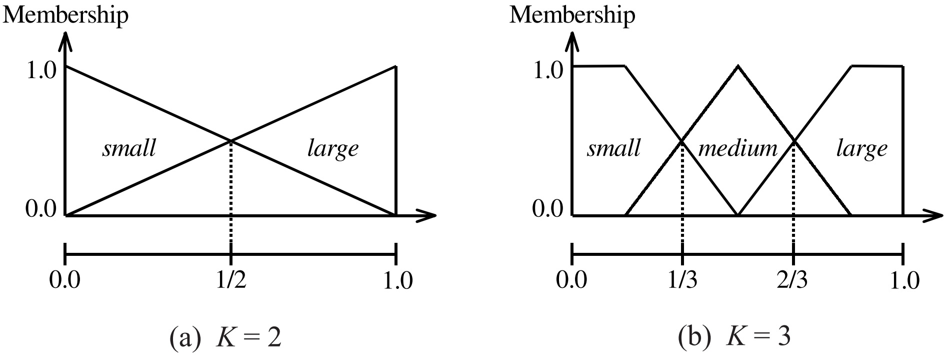
2. The classification boundary can be adjusted by the certainty grade of each fuzzy rule while it is specified by threshold values in the case of interval discretization. This may lead to better results by fuzzy discretization than interval discretization when the fuzzy and interval partitions of each axis are not tuned well.

3. Each fuzzy rule can classify a larger region than the corresponding non-fuzzy rule. This may lead to better results by fuzzy discretization than interval discretization when the number of rules is very small (i.e., when rules are sparse).

We examine these effects through computer simulations on well-known and frequently used pattern classification problems in the literature: iris data and wine data. These data sets are available from the UCI database (http://www.ics.uci.edu/~mlearn/MLSummary.html). In this section, we use homogeneous discretization for examining the above positive effects. Inhomogeneous discretization will be used in the next section for examining negative effects of fuzzy discretization.

4.1. Specification of overlap grade

In our computer simulations, we normalized all attribute values into real numbers in the unit interval $[0, 1]$. This means that the pattern space of each test problem was normalized into the n-dimensional unit hypercube $[0, 1]^n$. We homogeneously divided the domain interval $[0, 1]$ of each attribute into fuzzy sets and intervals as shown in Fig. 5 and Fig. 6. Other examples of homogeneous discretization are shown in Fig. 8 where K denotes the number of fuzzy sets or intervals.



(a) $K = 2$ (b) $K = 3$

Figure 8. Fully fuzzified partitions generated from homogeneous interval partitions

We specified fuzzy partitions from homogeneous interval partitions based on the following constraint conditions:

1. Membership functions are linear (i.e., triangular or trapezoidal).
2. The sum of neighboring membership functions is always 1.
3. Crossing points of neighboring membership functions coincide with the threshold values for interval partitions.
4. The membership value of each intermediate fuzzy set (e.g., *medium* in Fig. 8 (b)) is 1 at the midpoint of the corresponding interval. The membership value of the

smallest fuzzy set is 1 at the smallest attribute value 0 (i.e., at the lower limit of the domain interval $[0,1]$). The membership value of the largest fuzzy set is 1 at the largest attribute value 1 (i.e., at the upper limit of the domain interval $[0,1]$).

It should be noted that a fuzzy partition is not uniquely specified by these constraint conditions from a given homogeneous interval partition. Fig. 8 shows fully fuzzified partitions with the largest fuzziness satisfying these constraint conditions. In Fig. 9, we show partially fuzzified partitions, which also satisfy the above constraint conditions.

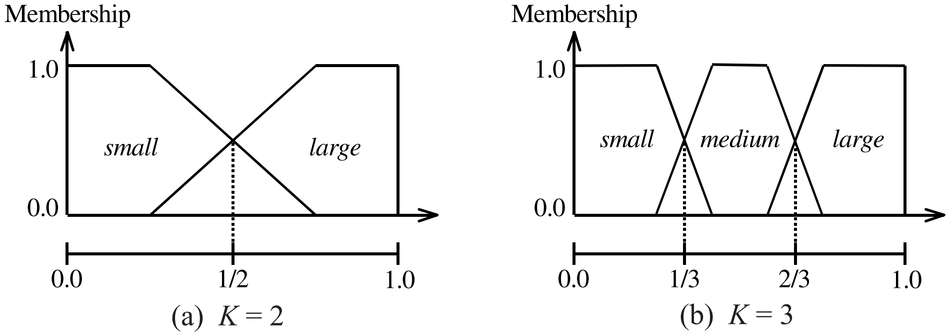


Figure 9. Partially fuzzified fuzzy partitions generated from homogeneous interval partitions

As shown in Fig. 9, the domain interval $[0,1]$ is divided in K trapezoidal antecedent fuzzy sets $A_j = (a_j, b_j, c_j, d_j)$, $j = 1, 2, \dots, K$, where the four parameters denote the four vertexes of the trapezoid $A_j (a_j \leqslant b_j \leqslant c_j \leqslant d_j)$. In the case of interval discretization, the first two parameters denote the lower limit (i.e., $a_j = b_j$) of the interval A_j and the last two parameters denote its upper limit (i.e., $c_j = d_j$). When A_j is a triangular fuzzy set, the first and last parameters denote its lower and upper limits, respectively. The other two parameters show the remaining vertex of the triangular A_j with the membership value 1 (i.e., $c_j = d_j$). From the constraint condition 2., the following relations hold between the adjacent fuzzy sets A_j and A_{j+1} (see Fig. 10):

$$c_j = a_{j+1}, \quad d_j = b_{j+1}. \tag{26}$$

Let us introduce the grade of fuzzification (i.e., overlap grade) denoted by F. Interval partitions correspond to no overlap (i.e., overlap grade is zero: $F = 0$). On the other hand, fully fuzzified partitions (e.g., Fig. 8) correspond to the full overlap (i.e., overlap grade is one: $F = 1$). Let $A_j^0 = (a_j^0, b_j^0, c_j^0, d_j^0)$ and $A_j^1 = (a_j^1, b_j^1, c_j^1, d_j^1)$ be an interval with no overlap and the corresponding fully fuzzified fuzzy set. Note that A_j^1 is a triangular or trapezoidal fuzzy set while A_j^0 is always an interval. As shown in Fig. 8, the following relations hold:

$$a_j^0 = b_j^0 = (a_j^1 + b_j^1)/2, \tag{27}$$

$$c_j^0 = d_j^0 = (c_j^1 + d_j^1)/2. \tag{28}$$

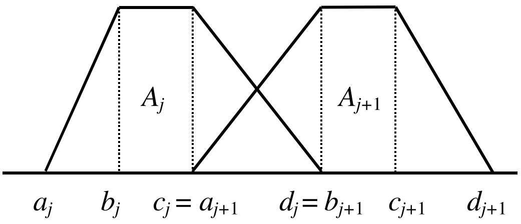


Figure 10. Illustration of the second constraint condition that the sum of adjacent membership functions should be always 1

We can generate a trapezoidal fuzzy set $A_j^F = (a_j^F, b_j^F, c_j^F, d_j^F)$ with an arbitrary overlap grade F from the interval A_j^0 with no overlap and the fully fuzzified fuzzy set A_j^1 as

$$a_j^F = a_j^0 + (a_j^1 - a_j^0)F, \tag{29}$$

$$b_j^F = b_j^0 + (b_j^1 - b_j^0)F, \tag{30}$$

$$c_j^F = c_j^0 + (c_j^1 - c_j^0)F, \tag{31}$$

$$d_j^F = d_j^0 + (d_j^1 - d_j^0)F, \tag{32}$$

Fig. 9 was depicted using these formulations with $F = 0.5$ from Fig. 8.

4.2. Simulation results on iris data

In this subsection, we show simulation results on iris data. The iris data set is a three-class pattern classification problem, which consists of 150 samples with four continuous attributes. In our computer simulations, the iris data set was handled as a three-class problem in the four-dimensional unit hypercube $[0, 1]^4$.

First we examined the classification performance of fuzzy rule-based systems in the case of sparse training patterns. We randomly selected only five samples from each class as training patterns. Thus the total number of training patterns was 15. Those training patterns were used for designing a fuzzy rule-based system. The other 135 samples were used as test patterns for evaluating the generalization ability of the designed fuzzy rule-based system. In our computer simulations, the pattern space was divided into K^4 fuzzy subspaces where K is the number of antecedent fuzzy sets on the domain interval of each attribute (see Fig. 8 and Fig. 9). We tried to generate a single fuzzy rule of the length four in each fuzzy subspace. There were many fuzzy subspaces where fuzzy rules could not be generated because there were no compatible training patterns. We examined four specifications of K: $K = 2, 3, 4, 5$. For each specification of K, we examined 11 specifications of the overlap grade F: $F = 0, 0.1, 0.2, \ldots, 1$. For each specification of K and F, we calculated the average classification rate on test patterns over 500 trials with different partitions of the 150 samples into 15 training patterns and 135 test patterns. Simulation results are summarized in Fig. 11.

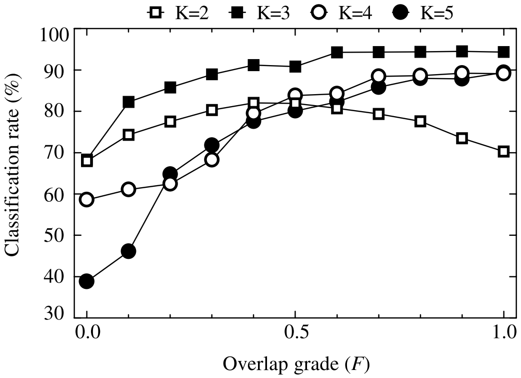


Figure 11. Average classification rates on test patterns. Only 15 samples in the iris data were used as training patterns

From Fig. 11, we can see that fuzzy discretization improved the generalization ability of fuzzy rule-based systems. The main reason for poor generalization ability of interval discretization (i.e., poor results in Fig. 11 in the case of no overlap: $F = 0$) is that only a small number of non-fuzzy rules were generated from sparse training patterns. As a result, the classification of many test patterns was rejected. The average number of generated fuzzy rules was summarized in Table 2. From this table, we can see that much more rules were generated in the case of fuzzy discretization than interval discretization. We also calculated the average rejection rate for each combination of K and F. Simulation results were summarized in Fig. 12. From this figure, we can see that the classification of many test patterns was rejected in the case of a large K (i.e., fine partition) and a small F (i.e., small overlap). The combination of a large K and a small F means a small decision region of each rule. The small decision region has two negative effects on the classification performance in the case of sparse training patterns. One is that the number of generated rules is small. The other is that each rule can classify only a small number of test patterns. As a result, many test patterns cannot be classified in the case of fine partitions and small overlap grades.

Table 2. Number of generated rules

Partition	Interval ($F = 0$)				Fuzzy ($F = 1$)			
Partition	$K = 2$	$K = 3$	$K = 4$	$K = 5$	$K = 2$	$K = 3$	$K = 4$	$K = 5$
# of rules	6.5	9.0	14.4	11.8	16.0	40.3	65.5	87.2

For examining the ability of fuzzy discretization to adjust classification boundaries using certainty grades, we also performed the same computer simulations using fuzzy rules with no certainty grades. This situation was simulated by assigning the same certainty grade to all the generated fuzzy rules (i.e., $CF_q = 1$ for $\forall q$). Simulation results are summarized in Table 3. From this table, we can see that the use of certainty grades improved the generalization ability of fuzzy rule-based systems in the case of $K = 4$.

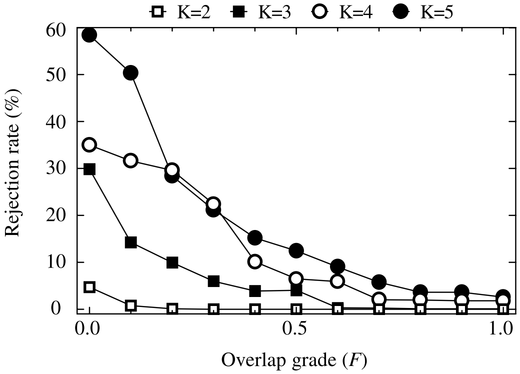


Figure 12. Average rejection rates on test patterns

For further discussions on the use of certainty grades in fuzzy rule-based classification systems, see [6].

Table 3. Average classification rates on test patterns. Fully fuzzified partitions were used

Partition	$K = 2$	$K = 3$	$K = 4$	$K = 5$
With CF_q	70.3	94.3	89.1	89.4
Without CF_q	71.3	94.3	84.3	88.5

Next we examined the classification performance of fuzzy rule-based systems in the case of sparse fuzzy rules. We used the ten-fold cross-validation (10CV) technique [13] for estimating the generalization ability of fuzzy rule-based systems. In the 10CV technique, the 150 samples in the iris data were randomly divided into ten subsets of the same size. Each subset consisted of five samples from each class. Nine subsets were used as training patterns and the remaining subset was used as test patterns. This was iterated ten times so that all subsets were used as test patterns once. The whole 10CV procedure was iterated 50 times using different partitions of the iris data into ten subsets for each specification of the partition K and the overlap grade F. In each trial in the 10CV procedure, generated fuzzy rules were divided into three groups according to their consequent classes. Fuzzy rules in each group were sorted in a descending order of the product of the confidence and the support. For constructing a fuzzy rule-based system, we selected the first N fuzzy rules from each group ($N = 1, 2 \dots$). This means that we used the product of the confidence and the support for choosing a pre-specified number of fuzzy rules. Other rule selection criteria such as the confidence and the support were examined in [10] where the best result was obtained from the product criterion. Simulation results are summarized in Fig. 13 for the case of $N = 1$ and Fig. 14 for the case of $N = 10$. Those figures show the average classification rate on test patterns for each specification for the partition K, the overlap grade F, and the number of fuzzy rules from each class. From Fig. 13, we can see that the effect of fuzzy discretization on the classification performance of fuzzy rule-based systems was significant when the number of fuzzy rules was very small. On the other hand, this effect was not so significant in Fig. 14 where the number of fuzzy rules was large and fuzzy partitions were not fine (e.g.,$K = 3$).

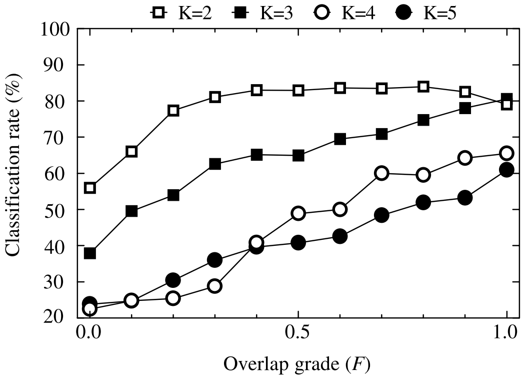


Figure 13. Average classification rates obtained from the 10CV procedure for the iris data using three fuzzy rules (i.e., a single fuzzy rule for each class

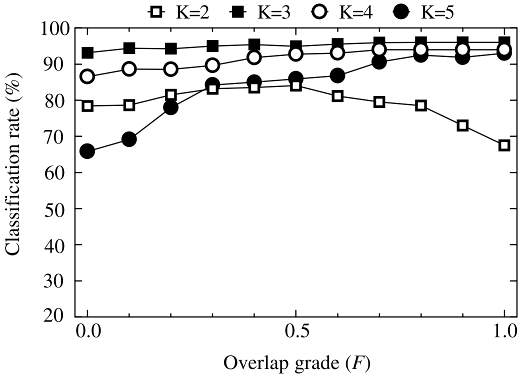


Figure 14. Average classification rates obtained from the 10CV procedure for the iris data using 30 fuzzy rules (i.e., ten fuzzy rules for each class)

4.3. Simulation results on wine data

In this subsection, we show simulation results on wine data. The wine data set is a three-class pattern classification problem, which consists of 178 samples with 13 continuous attributes. As a preprocessing procedure, we normalized each attribute value into a real number in the unit interval $[0, 1]$. Thus the wine data set was handled as a three-class pattern classification problem in the 13-dimensional unit hypercube $[0, 1]^{13}$.

The total number of possible combinations of antecedent fuzzy sets is K^{13} when each domain interval for the 13 attributes is divided into K antecedent fuzzy sets. It is impractical to generate all fuzzy rules from those huge combinations of antecedent fuzzy sets. In our computer simulations on the wine data, we only generated fuzzy rules of the length two or less. Fuzzy rules of the length two are written in the following form:

$$\text{Rule } R_q\text{: If } x_i \text{ is } A_{qi} \text{ and } x_j \text{ is } A_{qj} \text{ then Class } C_q \text{ with } CF_q, \tag{33}$$

where $i, j \in \{1, 2, \ldots, 13\}$.

Using the wine data set and fuzzy rules of the length two or less, we performed almost the same computer simulations as in the previous subsection using the iris data. First we examined the classification performance of fuzzy rule-based systems in the case of sparse training patterns. About 10% training patterns were randomly selected from the wine data as follows: six samples from Class 1 with 59 samples, seven samples from Class 2 with 71 samples, and five samples from Class 3 with 48 samples. Thus the total number of training patterns was 18. Those training patterns were used for designing a fuzzy rule-based system. The other 160 samples were used as test patterns for evaluating the generalization ability of the designed fuzzy rule-based system. For each specification of the partition K and the overlap grade F, we calculated the average classification rates on test patterns over 500 trials with different choices of 18 training patterns. Simulation results are summarized in Fig. 15. From this figure, we can see that fuzzy discretization significantly improved the generalization ability of fuzzy rule-based systems as in Fig. 11 on the iris data.

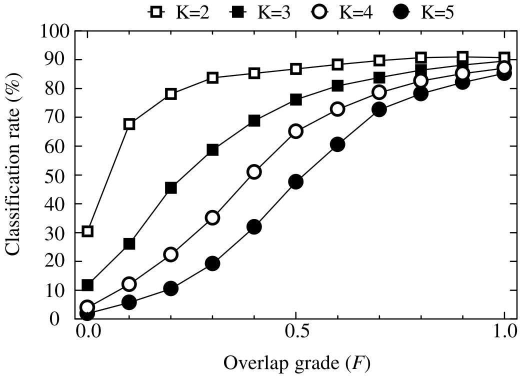


Figure 15. Average classification rates on test patterns. Only 18 samples in the wine data were used as training patterns

Next we examined the classification performance of fuzzy rule-based systems in the case of sparse fuzzy rules. As in the previous subsection, we used the 10CV technique for estimating the generalization ability of fuzzy rule-based systems. The whole 10CV procedure was iterated 50 times using different partitions of the wine data into ten subsets for each specification of the partition K and the overlap grade F. In each trial in the 10CV procedure, we used the product of the confidence and the support for selecting N fuzzy rules for each class. Simulation results are summarized in Fig. 16 for the case of $N = 1$ and Fig. 17 for the case of $N = 10$. From Fig. 16 where the number of fuzzy rules was very small, we can see that the effect of fuzzy discretization on the classification performance of fuzzy rule-based systems was significant. On the other hand, this effect was not so significant in Fig. 17 where the number of fuzzy rules was large. The same observations were obtained from the previous computer simulations on the iris data (i.e., from Fig. 13 and Fig. 14).

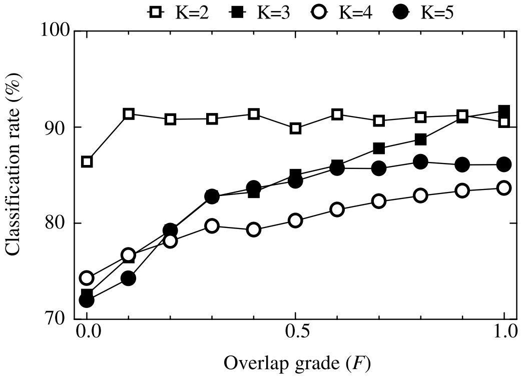


Figure 16. Average classification rates obtained from the 10CV procedure for the wine data using three fuzzy rules (i.e., a single fuzzy rule for each class)

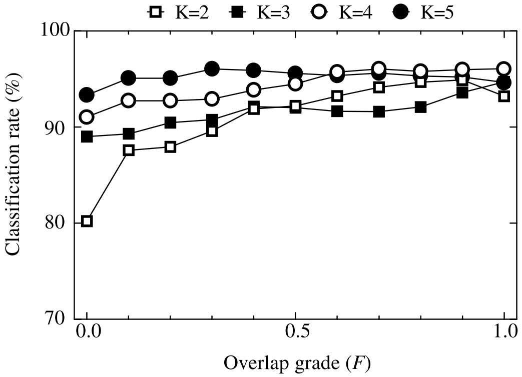


Figure 17. Average classification rates obtained from the 10CV procedure for the iris data using 30 fuzzy rules (i.e., ten fuzzy rules for each class)

5. Computer simulations using inhomogeneous discretization

In the previous computer simulations, we used homogeneous discretization. Since the location of classification boundaries totally depends on threshold values in the case of interval discretization, generalization ability can be improved by carefully choosing threshold values for each attribute. In this section, we specify interval discretization using the entropy measure as in Quinlan [13].

5.1. Rule extraction

In computer simulations in this section, the domain interval of each attribute was discretized independently of the other attributes. When the domain interval of an attribute was discretized into K intervals using $(K-1)$ threshold values, the threshold values were selected from $(m-1)$ candidates. Each candidate was the mid-point of a pair of neighboring attribute values in the given m training patterns. All the${}_{m-1}C_{K-1}$ combinations were examined for selecting $(K-1)$ threshold values from $(m-1)$ candidates. The entropy was calculated for each combination of $(K-1)$ threshold values (i.e., for each discretization). The discretization with the minimum entropy was selected for each attribute. We performed this discretization as a preprocessing procedure before the rule extraction. For comparison, we generated the corresponding fuzzy partition from the specified inhomogeneous interval partition for each attribute as shown in Fig. 18. The generation of the fuzzy partition in Fig. 18 is based on the four constraint conditions in Subsection 4.1. Fig. 18 is the fully fuzzified partition satisfying those constraint conditions. In the same manner as in Subsection 4.1, we can generate a partially fuzzified partition with an arbitrary overlap grade F from an inhomogeneous interval partition and the corresponding fully fuzzified partition.

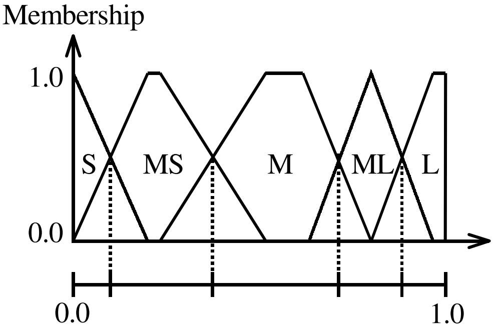


Figure 18. An example of a fully fuzzified partition corresponding to an inhomogeneous interval partition

Computer simulations in this section are for comparing fuzzy discretization with interval discretization in the situation where threshold values are carefully chosen. We do not intend to advocate the above-mentioned generation method of inhomogeneous fuzzy discretization. One problem is that inhomogeneous fuzzy discretization often leads to difficulties in linguistic interpretation of generated fuzzy sets. Another problem is that the examination of ${}_{m-1}C_{K-1}$ combinations of threshold values is time-consuming for large data sets.

5.2. Simulation results on iris data

As in Subsection 4.2, we examined the classification performance of fuzzy rule-based systems in the case of sparse training patterns (i.e., 15 training patterns from the iris data set). Simulation results are summarized in Fig. 19. From this figure, we can see that the classification ability of fuzzy rule-based systems on test patterns was improved by increasing the overlap grade. This improvement was more significant in the case of fine partitions (e.g., $K = 5$) than coarse partitions (e.g., $K = 2$).

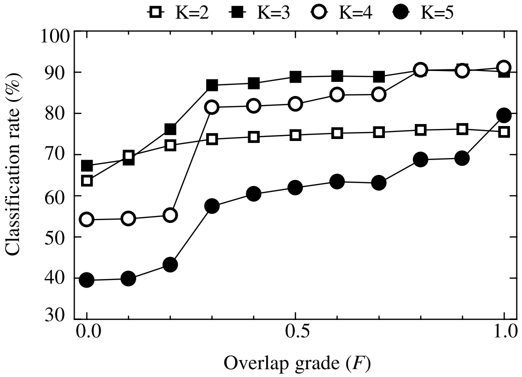


Figure 19. Average classification rates on test patterns in the case of inhomogeneous discretization. Only 15 samples in the iris data were used as training patterns

We also examined the classification performance of fuzzy rule-based systems in the case of sparse fuzzy rules. As in Subsection 4.2, we used the 10CV technique. Simulation results are summarized in Fig. 20 for the case of $N = 1$ and Fig. 21 for the case of $N = 10$. From these figures, we can see that the classification performance of fuzzy rule-based systems on test patterns was improved by increasing the overlap grade. This improvement was more significant in Fig. 20 with only three fuzzy rules than Fig. 21 with 30 fuzzy rules. In the case of $K = 3$ in Fig. 21 (i.e., black squares), we cannot observe clear improvement in the classification rate by the increase of the overlap grade

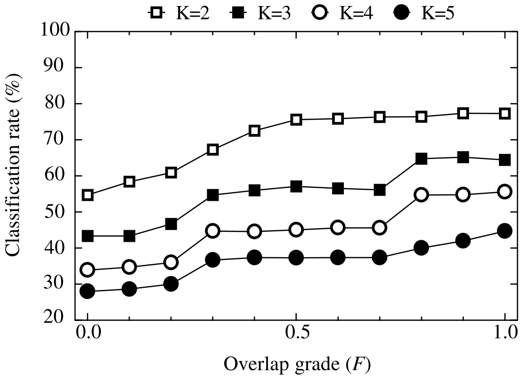


Figure 20. Average classification rates obtained from the 10CV procedure for the iris data using three fuzzy rules (i.e., a single fuzzy rule for each class). Inhomogeneous discretization was used

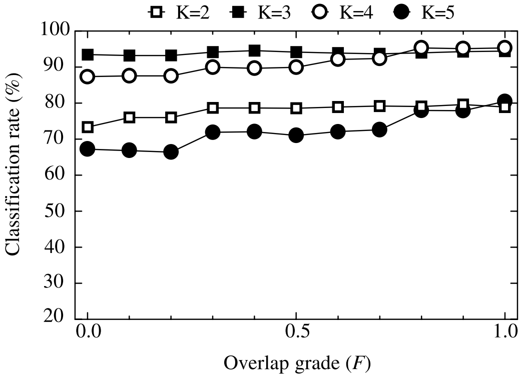


Figure 21. Average classification rates obtained from the 10CV procedure for the iris data using 30 fuzzy rules (i.e., ten fuzzy rules for each class). Inhomogeneous discretization was used

5.3. Simulation results on wine data

In the same manner as in Subsection 4.3, we performed computer simulations on the wine data using inhomogeneous discretization and fuzzy rules of the length two or less. First we examined the classification performance of fuzzy rule-based systems in the case of sparse training patterns (i.e., about 10% training patterns). Simulation results are summarized in Fig. 22. From Fig. 22, we can see that the classification performance of fuzzy rule-based systems was improved by increasing the overlap grade when they were generated from sparse training patterns. The same observation was obtained from the previous computer simulations based on a small number of training patterns.

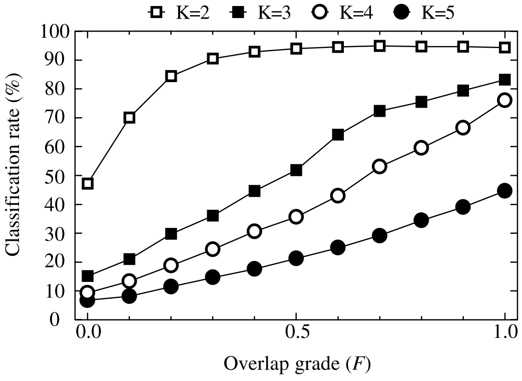


Figure 22. Average classification rates on test patterns using inhomogeneous discretization. Only 18 samples in the wine data were used as training patterns

Next we examined the classification performance of fuzzy rule-based systems in the case of sparse fuzzy rules using the 10CV technique. Simulation results are summarized in Fig. 23 for $N = 1$ and Fig. 24 for $N = 10$ where N is the number of fuzzy rules for each class. From Fig. 23, we can see that the classification performance of fuzzy rule-based systems was improved by increasing the overlap grade when the number of fuzzy rules was very small. The same observation was obtained from the previous computer simulations based on only three fuzzy rules (i.e., Figs. 13, 16, 20). On the other hand, the classification performance was impaired by the fuzzification of interval partitions in Fig. 24. Only this figure among simulation results in this paper shows clear deterioration in the classification performance by fuzzy discretization. Thus we conclude that the fuzzification of interval partitions can have a negative effect on the classification performance of rule-based systems when the following conditions are satisfied: The number of training patterns is not too small, the number of rules is not too small, and threshold values for interval partitions are appropriately specified.

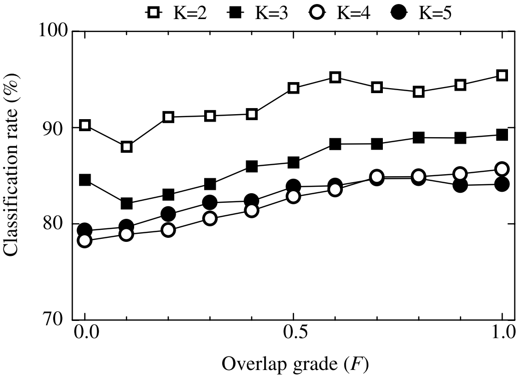


Figure 23. Average classification rates obtained from the 10CV procedure for the wine data using three fuzzy rules (i.e., a single fuzzy rule for each class). Inhomogeneous discretization was used

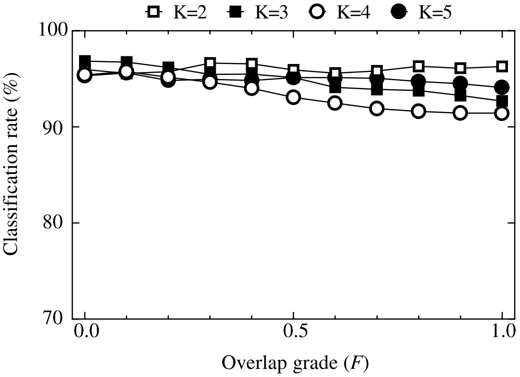


Figure 24. Average classification rates obtained from the 10CV procedure for the wine data using 30 fuzzy rules (i.e., ten fuzzy rules for each class). Inhomogeneous discretization was used

6. Concluding Remarks

In this paper, we first demonstrated some effects of fuzzy discretization in the fuzzy rule generation phase and the pattern classification phase. The main positive effects of fuzzy discretization are summarized as follows:

1. Many fuzzy rules can be generated from a small number of training patterns. This prevents fuzzy rule-based systems from significantly degrading when training patterns are sparse.

2. Classification boundaries can be adjusted by certainty grades. This makes fuzzy rule-based systems less sensitive to the choice of a partition for each attribute than non-fuzzy rule-based systems with interval discretization.

3. Each fuzzy rule has a large decision region when adjacent fuzzy rules are missing. This prevents fuzzy rule-based systems from rejecting the classification of many patterns when fuzzy rules are sparse.

Next these positive effects were examined by computer simulations on the iris data and the wine data. Simulation results showed that fuzzy rule-based systems had high generalization ability even when training patterns were sparse, fuzzy partitions were homogeneous, and the number of fuzzy rules was very small.

For examining the relation between the classification performance of fuzzy rule-based systems and the specification of fuzzy discretization, we introduced the overlap grade F between adjacent antecedent fuzzy sets. By changing the value of F from 0 (i.e., interval discretization with no overlap) to 1 (i.e., fully fuzzified discretization with the maximum overlap), we can examine the classification performance of fuzzy rule-based systems with different fuzzification levels. This scheme for examining the effect of fuzzy discretization on the classification performance of fuzzy rule-based systems can be applied to inhomogeneous discretization as well as homogeneous discretization. In our computer simulations, we examined inhomogeneous discretization based on the entropy measure as well as homogeneous discretization. That is, the domain interval of each attribute was discretized according to the distribution of training patterns. This discretization significantly improved the classification performance of non-fuzzy rule-based systems. Even in this case with carefully specified interval discretization, fuzzification of rule-based systems improved their classification performance when training patterns were sparse. Fuzzification also improved the classification performance when the number of rules was very small. In our computer simulations in this paper, we observed a negative effect of fuzzy discretization on the classification performance of rule-based systems only when training data were not sparse, each domain interval was carefully divided into intervals, and the number of rules was not too small.

References

[1] R. AGRAWAL and R. SRIKANT: Fast algorithms for mining association rules. *Proc. 20th Int. Conf. on Very Large Data Bases*. (1994), 487-499. Expanded version is available as IBM Research Report RJ9839, 1994.

[2] O. CORDON, M. J. DEL JESUS and F. HERRERA: A proposal on reasoning methods in fuzzy rule-based classification systems. *Int. J. Approximate Reasoning*. **20**(1), (1999), 21-45.

[3] J. DOUGHERTY, R. KOHAVI and M. SAHAMI: Supervised and unsupervised discretization of continuous features. *Proc. 12th Int. Conf. on Machine Learning*. (1995), 194-202.

[4] U. M. FAYYAD and K. B. IRANI: Multi-interval discretization of continuous-valued attributes for classification learning. *Proc. 13th Int. Joint Conf. on Artificial Intelligence*. (1993), 1022-1027.

[5] 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. *Int. J. Uncertainty, Fuzziness and Knowledge-Based Systems*. **9**(5), (2001), 587-604.

[6] H. ISHIBUCHI and T. NAKASHIMA: Effect of rule weights in fuzzy rule-based classification systems. *IEEE Trans. on Fuzzy Systems*. **9**(4), (2001), 506-515.

[7] H. ISHIBUCHI, T. NAKASHIMA and T. MORISAWA: Voting in fuzzy rule-based systems for pattern classification problems. *Fuzzy Sets and Systems*. **103**(2), (1999), 223-238.

[8] H. ISHIBUCHI, T. NAKASHIMA and T. MURATA: Three-objective genetics-based machine learning for linguistic rule extraction. *Information Sciences*. **136**(1-4), (2001), 109-133.

[9] H. ISHIBUCHI, K. NOZAKI and H. TANAKA: Distributed representation of fuzzy rules and its application to pattern classification. *Fuzzy Sets and Systems*. **52**(1), (1992), 21-32.

[10] H. ISHIBUCHI, T. YAMAMOTO and T. NAKASHIMA: Fuzzy data mining: Effect of fuzzy discretization. *Proc. 1st IEEE Int. Conf. on Data Mining*. (2001), 241-248.

[11] C. T. LEONDES (ED.): Fuzzy Theory Systems: Techniques and Applications. Academic Press, San Diego, 1999.

[12] K. NOZAKI, H. ISHIBUCHI and H. TANAKA: A simple but powerful heuristic method for generating fuzzy rules from numerical data. *Fuzzy Sets and Systems*, **86**(3), (1997), 251-270.

[13] J. R. QUINLAN: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, 1993.

[14] S. J. RUSSELL and P. NORVIG: Artificial Intelligence: A Modern Approach. Prentice-Hall, Upper Saddle River, 1995.

[15] L. X. WANG and J. M. MENDEL: Generating fuzzy rules by learning from examples. *IEEE Trans. on Systems, Man, and Cybernetics*. **22**(6), (1992), 1414-1427.