H. Ishibuchi, T. Yamamoto, and T. Nakashima, "An approach to fuzzy default reasoning for function approximation," *Soft Computing*, Vol. 10, No. 9, pp. 850-864, July 2006.

Final Version: Accepted for Publication in the Special Issue on *"New Trends in the Design of Fuzzy Systems"* in *Soft Computing*

An Approach to Fuzzy Default Reasoning for Function Approximation

HISAO ISHIBUCHI, TAKASHI YAMAMOTO, AND TOMOHARU NAKASHIMA

Department of Industrial Engineering, Osaka Prefecture University, Sakai, Osaka 599-8531, Japan

e-mail: {hisaoi, yama, nakashi}@ie.osakafu-u.ac.jp

Phone: +81-72-254-9350

Fax: +81-72-254-9915

Abstract

This paper discusses fuzzy reasoning for approximately realizing nonlinear functions by a small number of fuzzy if-then rules with different specificity levels. Our fuzzy rule base is a mixture of general and specific rules, which overlap with each other in the input space. General rules work as default rules in our fuzzy rule base. First we briefly describe existing approaches to the handling of default rules in the framework of possibility theory. Next we show that standard interpolation-based fuzzy reasoning leads to counterintuitive results when general rules include specific rules with different consequents. Then we demonstrate that intuitively acceptable results are obtained from a non-standard inclusion-based fuzzy reasoning method. Our approach is based on the preference for more specific rules, which is a commonly used idea in the field of default reasoning. When a general rule includes a specific rule and they are both compatible with an input vector, the weight of the general rule is discounted in fuzzy reasoning. We also discuss the case where general rules do not perfectly but partially include specific rules. Then we propose a genetics-based machine learning (GBML) algorithm for extracting a small number of fuzzy if-then rules with different specificity levels from numerical data using our inclusion-based fuzzy reasoning method. Finally we describe how our approach can be applied to the approximate realization of fuzzy number-valued nonlinear functions.

Keywords: *Fuzzy modeling, Fuzzy reasoning, Default reasoning, Genetics-based machine learning, Fuzzy number-valued function.*

1

Introduction

Fuzzy logic has been recognized as a convenient tool for handling continuous variables in rule-based systems (Russell & Norvig (1995)). This recognition is supported by many successful applications of fuzzy rule-based systems (for example, see Leondes (1999)). Fuzzy reasoning is a basic component in many applications of fuzzy logic to rule-based systems. In this paper, we discuss fuzzy reasoning for approximately realizing nonlinear functions using a small number of fuzzy if-then rules with different specificity levels. For an *n*-input and single-output nonlinear function $y = y(\mathbf{x})$, we use the following fuzzy if-then rules:

$$\text{Rule } R_k: \text{If } x_1 \text{ is } A_{k1} \text{ and ... and } x_n \text{ is } A_{kn} \text{ then } y \text{ is } B_k, \quad k = 1,2,...,N, \qquad (1)$$

where x_i is the *i*-th input variable of an *n*-dimensional input vector $\mathbf{x} = (x_1,..., x_n)$, *y* is an output variable, *k* is a rule index, A_{ki} is an antecedent linguistic value (e.g., *small* and *large*) for x_i, B_k is a consequent linguistic value, and *N* is the total number of fuzzy if-then rules. We also use the same type of fuzzy if-then rules for approximately realizing an *n*-input and single-output fuzzy number-valued nonlinear function $\tilde{y} = \tilde{y}(\mathbf{x})$, which maps an *n*-dimensional non-fuzzy input vector **x** to a fuzzy number $\tilde{y}$. There are many interpretations of fuzzy if-then rules. The most common and widely used interpretation is to consider the fuzzy if-then rule R_k in (1) as a fuzzy point $A_{k1} \times A_{k2} \times \cdots \times A_{kn} \times B_k$ and a collection of fuzzy if-then rules as a fuzzy graph (Dubois & Prade (1996)). Throughout this paper, we use this interpretation unless stated otherwise.

When fuzzy if-then rules are used for describing a nonlinear function with only two input variables, they are concisely written in a tabular form as in many applications of fuzzy rule-based systems to control problems. A two-dimensional fuzzy rule table describes the input-output relation of a nonlinear function in a human understandable manner (e.g., see Fig. 1). Fig. 1 includes 25 fuzzy if-then rules from the bottom-left rule "If x_1 is *small* and x_2 is *small* then y is *large*" to the top-right rule "If x_1 is *large* and x_2 is *large* then y is *small*". From this table, we can imagine a three-dimensional graphic as shown in Fig. 2.

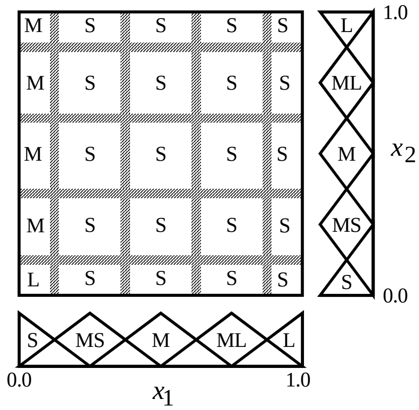


Fig. 1. An example of a two-dimensional fuzzy rule table. Each axis of the input space is divided into five linguistic values (S: *small*, MS: *medium small*, M: *medium*, ML: *medium large*, and L: *large*). The same five linguistic values are also used for describing the output variable.

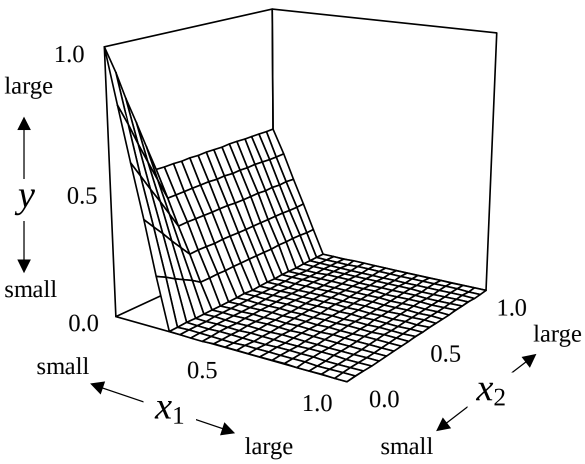


Fig. 2. Three-dimensional graphic corresponding to the fuzzy rule table in Fig. 1. This graphic is drawn using a standard interpolation-based fuzzy reasoning method.

While the tabular form representation of fuzzy if-then rules has been used in many applications to control problems, it cannot scale up to high-dimensional problems because the number of rules exponentially increases with the dimensionality of the input space. A simple trick for avoiding the exponential increase in the number of fuzzy if-then rules is to use general rules with many *don't care* conditions (i.e., many A_{ki} 's in (1) are *don't care*). Since each general rule covers a large portion of the input space, the entire input space can be covered by a small number of general rules. General rules describe rough behavior of a nonlinear function while its accurate behavior is described by specific rules with many antecedent conditions. Thus our fuzzy rule base is a mixture of general and specific rules. For example, the nonlinear function in Fig. 2 can be described by the following three rules:

Rule R_A: y is *small*, (2)

Rule R_B: If x_1 is *small* then y is *medium*, (3)

Rule R_C: If x_1 is *small* and x_2 is *small* then y is *large*. (4)

Since the first rule R_A has no antecedent conditions, it can be viewed as having *don't care* conditions on both x_1 and x_2. The second rule R_B has an antecedent condition on x_1 and a *don't care* condition on x_2. The third rule R_C is the same as the bottom-left rule in Fig. 1. The most general rule R_A is applicable to any input vector in the input space $[0,1]\times[0,1]$ of the nonlinear function in Fig. 2. The most specific rule R_C is applicable only when x_1 is *small* and x_2 is *small*.

Combinations of general and specific information are used for describing our knowledge in many situations. The following example is often used for explaining such a combination in the field of default reasoning (Reiter (1980), Poole (1991), Goldszmidt & Pearl (1996)).

Rule R_{I}: Birds fly, (5)

Rule R_{II}: Penguins are birds, (6)

Rule R_{III}: Penguins do not fly. (7)

Let us consider the reasoning about a penguin *x* using the above three rules. The question is whether *x* flies or not. If we use the first two rules R_{I} and R_{II}, we can conclude that *x* flies. We, however, usually conclude that *x* does not fly using the third rule R_{III}. This is because the third rule R_{III} is more specific than the first rule R_{I}. This example illustrates the preference for specific rules over general ones. Many studies on default reasoning espouse some form of preference for more specific information (Bacchus et al. (1996)). In data mining algorithms for finding exceptions (Suzuki & Kodratoff (1998), Knorr & Ng (1999), Knorr et al. (2000), Suzuki (2000)), specific information (i.e., exception rules) is implicitly assumed to have priority over general rules.

The three fuzzy if-then rules in (2)-(4) can be viewed as an inconsistent rule base. In other words, they are not coherent (Dubois & Prade (1996)). For example, an input vector with a *small* x_1 and a *small* x_2 is applicable to all the three fuzzy if-then rules. Their consequent linguistic values, however, are different from each other. When a specific rule is included in a general rule and they have different consequent linguistic values, they are viewed as being inconsistent with each other. Several approaches have been proposed for finding inconsistent rules in fuzzy rule bases (Yager & Larsen (1991), Bien & Yu (1995), Mees (1999), Viaene et al. (2000)). In those studies, it was implicitly assumed that the inconsistency in fuzzy rule bases should be removed or resolved. On the contrary, we explicitly utilize both specific and general rules even when they are inconsistent with each other. This is for describing nonlinear functions using a small number of fuzzy if-then rules. For example, three fuzzy if-then rules in (2)-(4) can concisely describe the nonlinear function in Fig. 2 depicted by the 25 fuzzy if-then rules in Fig. 1. As we have already explained using the penguin example from default reasoning, inconsistency is included in our knowledge in many cases.

The handling of default rules has been discussed in the framework of possibility theory. Yager (1987a, 1987b) proposed an idea of possibility qualification where a standard statement "*V* is *A*" is modified as "*V* is *A* is possible". The possibility-qualified statement is less restrictive than the standard statement. Let *V* be a variable with a base set *X*. The possibility-qualified statement "*V* is *A* is possible" is handled as a statement "*V* is A^+" where $A^+ = \{B \mid B \subseteq X,\ A \cap B \neq \phi\}$. Using the idea of possibility qualification, Yager (1987a, 1987b) discussed the handling of default rules of the type "If V_1 is A_1 and V_2 is A_2 is possible then *U* is *B*". The handling of default rules of this type was further discussed in Yager (1988a) where a knowledge system (i.e., world) with *n* atoms A_1, A_2, ..., A_n was considered. Each atom is a basic proposition which can be true or false. Thus the world has 2^n interpretations. When we have no information about the world, all the 2^n interpretations are possible. Our knowledge of the world eliminates some possibilities. While standard statements monotonically reduce the possible interpretations, possibility-qualified statements do not. That is, possibility-qualified statements exhibit the property of non-monotonicity. The truth value of an arbitrary statement can be evaluated as true, false or unknown from the current set of possible interpretations. Yager (1988b) proposed a mathematical programming approach to the handling of default rules where the determination of the truth value of a statement was formulated as a 0-1 integer programming problem. In his formulation, inference was performed by restricting the set of possible interpretations. Our knowledge of the world was used as constraint conditions in the 0-1 integer programming problem.

Dubois & Prade (1988) pointed out that Yager's proposal (Yager (1987a)) failed to capture the uncertainty nature of conclusions obtained via default rules. This is also the case in Reiter's default logic (Reiter (1980)) where facts inferred from default rules have the same status as facts inferred from standard rules with no exceptions (Dubois & Prade (1988)). They proposed an idea to assign a certainty grade to a default statement. Let λ be the certainty grade of the statement "*V* is *A*". When $\lambda = 1$, this statement is the same as the standard statement. On the other hand, this statement is less certain than the standard statement when $\lambda < 1$. The statement "*V* is *A*" with the certainty grade λ was handled as a fuzzy set A^{λ} with the membership function $\max\{\mu_A(v), 1-\lambda\}$ in Dubois & Prade (1988) where $\mu_A(v)$ is the membership function of *A*. This means that $(1-\lambda)$ was considered as the possibility that *V* is outside *A*. In Dubois & Prade (1987), the handling of multiple statements of the form "*V* is A_i with $Cr(A_i)$ ", $i = 1,2,...,n$ was discussed in the framework of evidence theory where $Cr(A_i)$ is the grade of credibility of the statement "*V* is A_i ". They proposed the minimum specificity principle for inducing a basic probability assignment $m(\cdot)$ from the given set of statements with credibility grades. Dubois et al. (1994a, 1994b) discussed the handling of default rules using possibilistic logic. They considered two types of statements: a certainty-qualified statement $(\varphi, N(\varphi))$ and a possibility-qualified statement $(\varphi, \Pi(\varphi))$ where φ is a statement, $N(\varphi)$ is the certainty grade, and $\Pi(\varphi)$ is the possibility grade. While Yager (1987a, 1887b, 1988a) discussed the determination of possible interpretations, the determination of a possibility distribution over the 2^n interpretations was discussed in Dubois et al. (1994a, 1994b) using possibilistic logic. Dubois & Prade (1996) explained the handling of certainty-qualified and possibility-qualified fuzzy if-then rules: $(\varphi, N(\varphi))$ and $(\varphi, \Pi(\varphi))$ where φ is a fuzzy if-then rule.

Our goal in this paper is to linguistically describe unknown nonlinear functions using a small number of fuzzy if-then rules. For this purpose, we propose an inclusion-based fuzzy reasoning method that can simultaneously handle fuzzy if-then rules with different specificity levels in an intuitively acceptable manner. We also propose a genetics-based machine learning (GBML) algorithm for automatically extracting fuzzy if-then rules with different specificity levels from numerical data. Our GBML algorithm simultaneously minimizes three objectives: the approximation error, the number of fuzzy if-then rules, and the number of antecedent conditions in each rule. Our fuzzy reasoning method is based on the preference for more specific rules, which is a commonly used idea in the field of default reasoning. More specifically, we use a preference order defined by an inclusion relation among fuzzy if-then rules. In the terminology of default reasoning, our fuzzy reasoning method can be viewed as a kind of priority-based approach (Brewka & Eiter (2000), Delgrande & Schaub (2000)). Specificity is frequently used for resolving the conflict among contradictory conclusions in priority-based approaches (Dung & Son (2001)). The characteristic features of our inclusion-based fuzzy reasoning method are summarized as follows:

1. Our fuzzy reasoning method is used for function approximation.
2. A preference order among fuzzy if-then rules in a fuzzy rule base is automatically specified with no intervention of human users during the iterative execution of our GBML algorithm.
3. The weight of each fuzzy if-then rule is also automatically specified with no intervention of human users.
4. The weight of each fuzzy if-then rule depends on the input vector as well as the other rules in the same rule base. This means that the same fuzzy if-then rule may have different weights for different input vectors in the same rule base. It may also have different weights for the same input vector in different rule bases.
5. In our fuzzy reasoning method, the weight of each fuzzy if-then rule is used for decreasing the membership functions of its antecedent linguistic values.

An idea of inhibiting the activation of general rules when more specific rules match the input has already been suggested and discussed in Yager (1993) and Hoffmann & Pfister (1996,1997). The proposed fuzzy reasoning method in this paper can be viewed as a practical implementation scheme of their idea. The main contribution of this paper is that the usefulness of such an idea is clearly demonstrated in various situations such as function approximation, fuzzy genetics-based machine learning, and approximation of fuzzy number-valued functions. Another contribution is that such an idea is extended for manually controlling the magnitude of the preference for more specific rules using a user-definable parameter.

The rest of this paper is organized as follows. In Section 2, we show that counterintuitive results can be obtained from a standard interpolation-based fuzzy reasoning method when our fuzzy rule base is a mixture of general and specific rules. Then we show that intuitively acceptable results are obtained from our non-standard inclusion-based fuzzy reasoning method. In Section 3, our inclusion-based fuzzy reasoning method is modified for handling partially overlapping rules. In Section 4, we propose a genetics-based machine learning (GBML) algorithm for extracting a small number of fuzzy if-then rules with different specificity levels from numerical data using our inclusion-based fuzzy reasoning method. In Section 5, our inclusion-based fuzzy reasoning method is applied to approximation problems of fuzzy number-valued nonlinear functions. Finally Section 6 concludes this paper.

2

Inclusion-based fuzzy reasoning

2.1

Standard interpolation-based fuzzy reasoning

The following fuzzy reasoning method has been frequently used in fuzzy rule-based systems since its first proposal in a neuro-fuzzy system (Ichihashi (1991)):

$$\hat{y}(\mathbf{x}) = \frac{\sum_{k=1}^{N} \mu_k(\mathbf{x}) \cdot b_k}{\sum_{k=1}^{N} \mu_k(\mathbf{x})}, \quad (8)$$

where $\mu_k(\mathbf{x})$ is the compatibility grade of the fuzzy if-then rule R_k with the input vector **x**, and b_k is a representative real number of the consequent linguistic value

B_k . The compatibility grade $\mu_k(\mathbf{x})$ is usually calculated by the product operation as

$$\mu_k(\mathbf{x}) = \mu_{k1}(x_1) \times \cdots \times \mu_{kn}(x_n), \tag{9}$$

where $\mu_{ki}(\cdot)$ is the membership function of the antecedent linguistic value A_{ki} . The representative real number b_k can be viewed as a result of the defuzzification of the consequent linguistic value B_k . In computational experiments of this paper, the center of the triangular membership function of each linguistic value is used as its representative real number. That is, 0, 0.25, 0.5, 0.75, and 1 are used for the five linguistic values *small*, *medium small*, *medium*, *medium large*, and *large* in Fig. 1, respectively. The three-dimensional graphic in Fig. 2 was drawn using the fuzzy reasoning method in (8) and the 25 fuzzy if-then rules in Fig. 1.

The fuzzy reasoning method in (8) can be viewed as a simplified version of the Takagi-Sugeno (TS) model (Takagi & Sugeno (1985)) where a linear function is used in the consequent part of each fuzzy if-then rule. The simplified fuzzy reasoning method in (8) has several advantages. For example, its reasoning mechanism is very simple, and it is suitable for gradient-based learning algorithms.

We applied the fuzzy reasoning method in (8) to the three fuzzy if-then rules in (2)-(4). The generated nonlinear function is shown in Fig. 3. We can see that the three-dimensional graphic in Fig. 3 is different from Fig. 2 depicted by the 25 fuzzy if-then rules in Fig. 1. As we have already described, we intuitively feel that the three fuzzy if-then rules in (2)-(4) describe almost the same input-output relation as the 25 fuzzy if-then rules in Fig. 1. The fuzzy reasoning result in Fig. 3 is, however, different from Fig. 2.

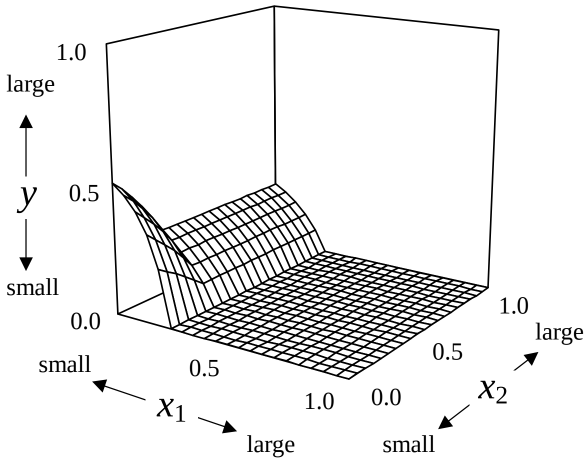


Fig. 3. Nonlinear function obtained from the three fuzzy if-then rules in (2)-(4) using the standard interpolation-based fuzzy reasoning method in (8).

In general, fuzzy reasoning for function approximation is based on interpolation. That is, the estimated output value $\hat{y}(\mathbf{x})$ is calculated by interpolating the consequent parts of compatible fuzzy if-then rules with the input vector $\mathbf{x}$. This interpolation nature is, however, contradictory to the preference for more specific rules. Let us consider the calculation of the estimated output $\hat{y}(\mathbf{x})$ for a *small* x_1 and a *small* x_2. For such an input vector, we intuitively infer the output value *y* as *large* from the three fuzzy if-then rules in (2)-(4) as we have already explained in Section 1. The estimated output by the fuzzy reasoning method in (8), however, is not *large* but *medium* in Fig. 3.

Such a counterintuitive result is obtained not only from the fuzzy reasoning method in (8) but also from many other reasoning methods. Let $\mathbf{x} = (0, 0)$ be the input vector to the fuzzy rule base with the three fuzzy if-then rules in (2)-(4). Since the membership function of the linguistic value *small* is 1 at the input value 0 (i.e., $\mu_{small}(0) = 1$), all the three rules are fully compatible with the input vector **x** independent of the choice of a *t*-norm (e.g., minimum and product):

$$\mu_{R_A}(\mathbf{x}) = \mu_{R_B}(\mathbf{x}) = \mu_{R_C}(\mathbf{x}) = 1.0. \qquad (10)$$

Thus the estimated output is calculated by the interpolation of the consequent linguistic values of the three fuzzy if-then rules. In Fig. 4, we show the membership function of each consequent linguistic value. We can see that the union of the three membership functions in Fig. 4 is symmetric with respect to the output value 0.5. Thus the estimated output $\hat{y}(\mathbf{x})$ is calculated as 0.5 independent of the choice of a defuzzification method (e.g., center of gravity). This discussion shows that the estimated output $\hat{y}(\mathbf{x})$ is calculated as 0.5 (i.e., *medium*) for the input vector $\mathbf{x} = (0, 0)$ from the three fuzzy if-then rules in (2)-(4) by almost all fuzzy reasoning methods for function approximation. On the contrary, we intuitively infer the output value as *large* (not *medium*) for a *small* x_1 and a *small* x_2 from the three fuzzy if-then rules.

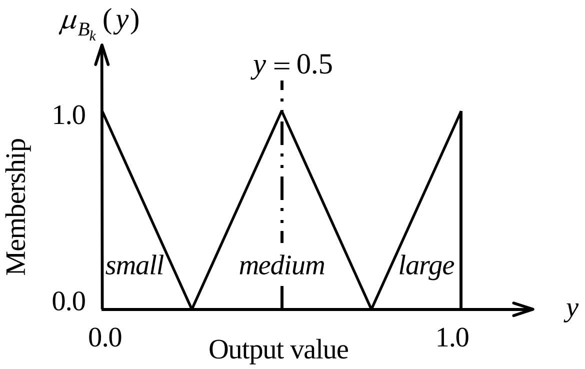


Fig. 4. Membership functions of the consequent linguistic values: *small*, *medium*, and *large*.

2.2

Non-standard inclusion-based fuzzy reasoning

For implementing the preference for more specific rules, we have suggested an idea of inclusion-based fuzzy reasoning (Ishibuchi (1999a, 1999b)). First, we define an inclusion relation between fuzzy if-then rules. Let us consider the following two fuzzy if-then rules R_k and R_q:

$$R_k: \text{If } x_1 \text{ is } A_{k1} \text{ and ... and } x_n \text{ is } A_{kn} \text{ then } y \text{ is } B_k, \qquad (11)$$

$$R_q: \text{If } x_1 \text{ is } A_{q1} \text{ and ... and } x_n \text{ is } A_{qn} \text{ then } y \text{ is } B_q. \qquad (12)$$

When $A_{qi} \subseteq A_{ki}$ for $\forall i$ and $A_{qi} \neq A_{ki}$ for $\exists i$, we say that R_q is strictly included in R_k (i.e., $R_q \subset R_k$).

When only the two rules R_k and R_q with the inclusion relation $R_q \subset R_k$ are compatible with the input vector $\mathbf{x}$, the specific rule R_q is mainly used in fuzzy reasoning. That is, the weight of the general rule R_k is discounted. Our idea is to determine the amount of the discount for R_k using the compatibility grade $\mu_q(\mathbf{x})$ of the specific rule R_q with the input vector **x**. More specifically, the weight of R_k is defined as $(1-\mu_q(\mathbf{x}))$. When the specific rule R_q is fully compatible with the input vector **x**, the weight of the general rule R_k is zero. This means that R_k has no effect on the calculation of the estimated output value $\hat{y}(\mathbf{x})$. On the other hand, when the compatibility grade of R_q with **x** is very small, the amount of the discount for R_k is also very small. In this case, R_k has almost the same weight as R_q. Since the general rule R_k may include multiple rules, its weight is defined as

$$w(R_k, \mathbf{x}) = \prod_{\substack{q=1 \\ R_q \subset R_k}}^{N} (1-\mu_q(\mathbf{x})). \tag{13}$$

When no fuzzy if-then rule is included in R_k, $w(R_k, \mathbf{x})$ is specified as $w(R_k, \mathbf{x}) = 1$ because the weight of R_k should not be discounted in this case. It should be noted that the weight of each rule depends on the compatibility grades of other rules with the input vector **x**. This means that the weight is context-dependent. Different weights are assigned to the same rule for different input vectors. Moreover, the same rule may have different weights for the same input vector in different rule bases because the weight of each rule depends on other rules.

Using the rule weight $w(R_k, \mathbf{x})$ of each fuzzy if-then rule R_k, our inclusion-based fuzzy reasoning method is written as

$$\hat{y}(\mathbf{x}) = \frac{\sum_{k=1}^{N} w(R_k, \mathbf{x}) \cdot \mu_k(\mathbf{x}) \cdot b_k}{\sum_{k=1}^{N} w(R_k, \mathbf{x}) \cdot \mu_k(\mathbf{x})}. \tag{14}$$

Let us illustrate this fuzzy reasoning method using the three fuzzy if-then rules R_A, R_B and R_C in (2)-(4). Since the inclusion relation $R_C \subset R_B \subset R_A$ holds, the weight $w(R_k, \mathbf{x})$ of each rule is calculated as

$$w(R_A, \mathbf{x}) = (1-\mu_B(\mathbf{x})) \times (1-\mu_C(\mathbf{x})), \tag{15}$$

$$w(R_B, \mathbf{x}) = 1-\mu_C(\mathbf{x}), \tag{16}$$

$$w(R_C,\ \mathbf{x}) = 1\,. \qquad (17)$$

For a *small* x_1 and a *small* x_2, all the three rules are applicable. Thus the weights of R_A and R_B become small due to (15) and (16). As a result, the most specific rule R_C is mainly used in our inclusion-based fuzzy reasoning method. On the other hand, R_B is mainly used when x_1 is *small* and x_2 is not *small*. When x_1 is not *small*, only the most general rule R_A is used because no other rules are applicable to this situation. In Fig. 5, we show the shape of the estimated nonlinear function using our inclusion-based fuzzy reasoning method. We can see from Fig. 5 that our method successfully implements the preference for more specific rules through the weighting scheme in (13)-(14). The estimated nonlinear function in Fig. 5 is more consistent with our intuition than Fig. 3.

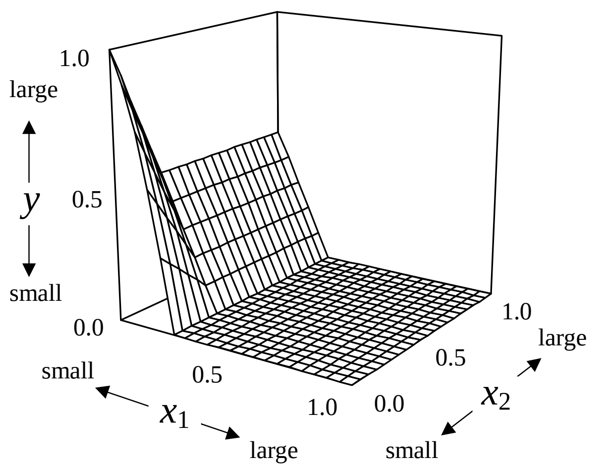


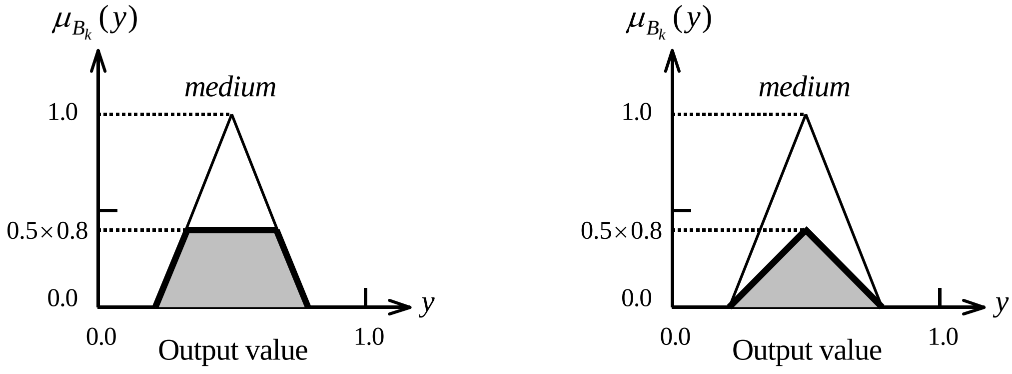
Fig. 5. Nonlinear function obtained from the three fuzzy if-then rules in (2)-(4) using our non-standard inclusion-based fuzzy reasoning method.

2.3

Discussions and extensions

Our idea can be applied to not only the fuzzy reasoning method in (8) but also other methods. Many fuzzy reasoning methods for function approximation can be classified into two categories (Cordon et al. (1997), Emami et al. (1999)): FITA (first-infer-then-aggregate) and FATI (first-aggregate-then-infer). The fuzzy reasoning method in (8) is one example of FITA models. In FITA models, first a crisp output value (e.g., b_k in (8)) from each fuzzy if-then rule is calculated by a defuzzification procedure. Then the final estimation $\hat{y}(\mathbf{x})$ is calculated by aggregating the crisp output values from compatible fuzzy if-then rules (e.g., using a weighted average in (8)). Our idea is directly applicable to FITA models. That is, the compatibility grade $\mu_k(\mathbf{x})$ is multiplied by the weight $w(R_k, \mathbf{x})$ as in (14). This is executed in the second phase (i.e., aggregation phase) of FITA models. The Takagi-Sugeno (TS) model (Takagi & Sugeno (1985)) can be viewed as an example of FITA models. In the TS model, a linear function $y_k(\mathbf{x})$ is used in the consequent part of each fuzzy if-then rule instead of the consequent linguistic value B_k in (1). Thus our inclusion-based fuzzy reasoning method in (14) can be directly applied to the TS model by replacing the representative real number b_k with the linear function $y_k(\mathbf{x})$.

In FATI models, first an aggregated fuzzy set is constructed from the consequent linguistic values B_k 's of compatible fuzzy if-then rules. Then the final estimation $\hat{y}(\mathbf{x})$ is calculated by a defuzzification procedure. In the first phase of FATI models, the consequent linguistic value B_k is modified to a fuzzy set B_k^* by the compatibility grade $\mu_k(\mathbf{x})$ for constructing the aggregated fuzzy set. Our idea is also applicable to FATI models while we cannot directly use (14). As in the application of our idea to FITA models, the compatibility grade $\mu_k(\mathbf{x})$ is multiplied by the weight $w(R_k, \mathbf{x})$ in the aggregation phase (i.e., the first phase of FATI models). More specifically, the modified fuzzy set B_k^* is generated from the consequent linguistic value B_k using $w(R_k, \mathbf{x}) \cdot \mu_k(\mathbf{x})$ instead of $\mu_k(\mathbf{x})$. Two examples of the modified fuzzy set B_k^* are shown in Fig. 6 for the case of $B_k = medium$, $\mu_k(\mathbf{x}) = 0.5$ and $w(R_k, \mathbf{x}) = 0.8$. The membership function of B_k^* is calculated by the minimum and product operations in Fig. 6 (a) and (b), respectively.



(a) Minimum operation. (b) Product operation.

Fig. 6. Examples of B_k^* obtained from *medium* when $\mu_k(\mathbf{x}) = 0.5$ and $w(R_k, \mathbf{x}) = 0.8$.

The multiplication of the compatibility grade $\mu_k(\mathbf{x})$ by the weight $w(R_k, \mathbf{x})$ can be also viewed as modifying the membership function $\mu_{ki}(x_i)$ of each antecedent fuzzy set A_{ki} so that the compatibility grade becomes $w(R_k, \mathbf{x}) \cdot \mu_k(\mathbf{x})$. For example, the membership function $\mu_{ki}(x_i)$ is modified as $\sqrt[n]{w(R_k, \mathbf{x})} \cdot \mu_{ki}(x_i)$ when the product operation is used for calculating the compatibility grade as in (9). On the other hand, when the minimum operation is used, $\mu_{ki}(x_i)$ is modified as $w(R_k, \mathbf{x}) \cdot \mu_{ki}(x_i)$. In this interpretation, our idea does not change fuzzy reasoning but modify fuzzy if-then rules. Thus our idea is applicable to any fuzzy reasoning methods. For example, let us consider an implication statement "If *V* is *A* then *U* is *B*" where *A* and *B* are fuzzy subsets of base sets *X* and *Y*, respectively. In the logical implication, this statement is handled as the relation $D = \overline{A} \cup B$ on $X \times Y$. When we have a piece of information in the form of "*V* is *C*", the inference result is the projection of $(\overline{A} \cap C) \cup (B \cap C)$ on the base set *Y*. The inference result is "*U* is *B*" when $(\overline{A} \cap C) = \phi$ holds. On the other hand, we cannot infer anything (i.e., the inference result is "*U* is unknown") when $(\overline{A} \cap C)$ is a normal fuzzy set (i.e., its maximum membership value is 1). Let us apply the logical implication to the three fuzzy if-then rules R_A, R_B and R_C in (2)-(4) for inferring the output value for the input vector $\mathbf{x} = (x_1, x_2) = (0, 0)$ using our idea. In this case, we can infer from the three fuzzy if-then rules that "*y* is *large*". This is explained as follows. Since R_C is the most specific rule, its weight is always 1. Thus "*y* is *large*" is inferred from R_C because the compatibility grade of $\mathbf{x} = (0,\ 0)$ with the antecedent part of R_C is 1. On the other hand, the inference result from R_B is *"y is unknown"* for $\mathbf{x} = (0,\ 0)$ because its antecedent part (*small*, *don't care*) is modified such that $\mu_{R_B}(\mathbf{x}) = 0$ for $\mathbf{x} = (0,\ 0)$ by the weight $w(R_B,\ \mathbf{x}) = 0$. From the same reason, the inference result for $\mathbf{x} = (0,\ 0)$ from R_A is also *"y is unknown"*. Thus we can infer that *"y is large"* for $\mathbf{x} = (0,\ 0)$ from the three fuzzy if-then rules in (2)-(4). The inconsistency among those rules for $\mathbf{x} = (0,\ 0)$ is resolved by the proposed weighting scheme in (13).

The validity of each fuzzy if-then rule as a local approximator of the nonlinear function can be evaluated by a measure called *confidence* in the field of data mining (Agrawal et al. (1996)). Let us assume that we have *m* input-output pairs $(\mathbf{x}_p,\ y_p)$, $p = 1, 2, ..., m$ from the nonlinear function. The confidence of the fuzzy if-then rule R_k is defined as follows (Hong et al. (2001), Ishibuchi et al. (2001b)):

$$Confidence(R_k) = \frac{\sum_{p=1}^{m} \mu_k(\mathbf{x}_p) \cdot \mu_k(y_p)}{\sum_{p=1}^{m} \mu_k(\mathbf{x}_p)}, \qquad (18)$$

where $\mu_k(\mathbf{x}_p)$ is the compatibility grade of the antecedent part of R_k with the input vector $\mathbf{x}_p$, and $\mu_k(y_p)$ is the compatibility grade of the consequent part of R_k with the output value y_p. The denominator in (18) corresponds to the number of input-output pairs that are compatible with the antecedent part. The numerator corresponds to the number of input-output pairs that are compatible with both the antecedent and consequent parts.

Let us examine the validity of our interpolation-based fuzzy reasoning method using the confidence measure. First we generate 441 input-output pairs $(x_{p1},\ x_{p2},\ y_p)$, $p = 1, 2, ..., 441$ where $x_{pi} =$ 0.00, 0.05, ..., 1.00 for $i = 1, 2$ from the nonlinear function in Fig. 3 depicted by the standard interpolation-based fuzzy reasoning method. That is, 441 input-output pairs are generated using the uniformly partitioned 21×21 grid of the two-dimensional input space $[0, 1] \times [0, 1]$. Then we calculate the confidence values of the three fuzzy if-then rules in (2)-(4):

R_A : y is *small* (Confidence: 0.82), (19)

R_B : If x_1 is *small* then y is *medium* (Confidence: 0.11), (20)

R_C : If x_1 is *small* and x_2 is *small* then y is *large* (Confidence: 0.00). (21)

While the most general rule R_A has a high confidence value, the confidence values of the other rules are small. The zero confidence of R_C means that no input-output pairs are compatible with this rule. That is, R_C does not correctly describe the nonlinear function in Fig. 3. From these results, we can see that the standard interpolation-based fuzzy reasoning method is not appropriate for estimating the output value from the three fuzzy if-then rules.

In the same manner, we calculate the confidence values for the nonlinear function in Fig. 5 depicted by our inclusion-based fuzzy reasoning method. The confidence value of each rule is calculated as

R_A : y is *small* (Confidence: 0.79), (22)

R_B : If x_1 is *small* then y is *medium* (Confidence: 0.49), (23)

R_C : If x_1 is *small* and x_2 is *small* then y is *large* (Confidence: 0.23). (24)

We can see that the confidence values for Fig. 5 are higher than those for Fig. 3 on the average. This observation suggests that our inclusion-based fuzzy reasoning method in Fig. 5 is more appropriate than the standard interpolation-based fuzzy reasoning method in Fig. 3 for estimating the output value from the three fuzzy if-then rules.

The magnitude of the preference for more specific rules in our inclusion-based fuzzy reasoning method can be adjusted by introducing an additional non-negative parameter β to (14) as

$$\hat{y}(\mathbf{x}) = \frac{\sum_{k=1}^{N} \left(w(R_k, \mathbf{x})\right)^{\beta} \cdot \mu_k(\mathbf{x}) \cdot b_k}{\sum_{k=1}^{N} \left(w(R_k, \mathbf{x})\right)^{\beta} \cdot \mu_k(\mathbf{x})} . \quad (25)$$

Our inclusion-based fuzzy reasoning method in (25) is reduced to the standard interpolation-based method when $\beta = 0$. The original formulation of our inclusion-based fuzzy reasoning method in (14) corresponds to the case of $\beta = 1$. The larger the value of β is, the larger the magnitude of the preference for specific rules is. In Fig. 7, we show the three-dimensional graphic depicted from the above three fuzzy if-then rules using (25) with $\beta = 10$. From the comparison between Fig. 5 with $\beta = 1$ and Fig. 7 with $\beta = 10$, we can see that the preference for more specific rules is strengthened in Fig. 7 by a large value of β. For the nonlinear function in Fig. 7, the confidence values are calculated as

R_A : y is *small* (Confidence: 0.76), (26)

R_B : If x_1 is *small* then y is *medium* (Confidence: 0.73), (27)

R_C : If x_1 is *small* and x_2 is *small* then y is *large* (Confidence: 0.87). (28)

These confidence values show that the validity of each fuzzy if-then rule as a local approximator is increased by the use of our inclusion-based fuzzy reasoning method with a large value of β. The negative effect of a large value of β is that the change of the estimated output value becomes abrupt as shown in Fig. 7.

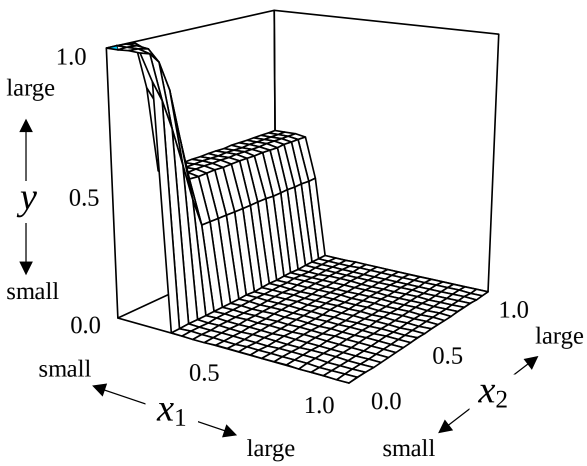


Fig. 7. Nonlinear function obtained from the three fuzzy if-then rules in (2)-(4) using the non-standard inclusion-based fuzzy reasoning method with $\beta = 10$.

Our inclusion-based fuzzy reasoning method is based on the strict inclusion relation $R_q \subset R_k$ defined by $A_{qi} \subseteq A_{ki}$ for $\forall i$ and $A_{qi} \neq A_{ki}$ for $\exists i$. When no inclusion relation holds, our method is reduced to standard interpolation-based fuzzy reasoning. There may be, however, many cases where general rules do not perfectly but partially include specific rules. For handling such a case, we generalize our inclusion-based method to a preference order-based method in the next section where we assume the existence of a preference order-based rule hierarchy among fuzzy if-then rules. Another possible extension is the use of a kind of inclusion grade between fuzzy if-then rules (i.e., a fuzzy relation over fuzzy if-then rules) instead of the crisp inclusion (or preference) relation. The weight of a general rule is discounted by a specific rule depending on the inclusion grade between them. Such an extension to our inclusion-based method is not discussed in this paper (i.e., it is left for future research). This is because the connection between the inclusion grade and the magnitude of the preference for more specific rules is not clear in the case of partially overlapping fuzzy if-then rules.

3

Extension to the case of partially overlapping rules

3.1

Motivation

In our inclusion-based fuzzy reasoning method, the preference for more specific rules is implemented by discounting the weights of general rules when they include specific rules. Our method, however, does not discount the weights of general rules when they partially overlap with specific rules. Let us consider the following three fuzzy if-then rules for a nonlinear function with three inputs:

Rule R_a : y is *small*, (29)

Rule R_b : If x_1 is *small* then y is *medium*, (30)

Rule R_c : If x_2 is *large* and x_3 is *large* then y is *large*. (31)

The weight of the most general rule R_a is discounted for any input vector $\mathbf{x}$ with a *small* x_1 because such an input vector is compatible with the second rule R_b. The weight of R_a is also discounted using the compatibility grade of the most specific rule R_c when x_2 is *large* and x_3 is *large*. The weight of the second rule R_b is, however, never discounted for any input vector while R_b is more general than R_c. This is because no inclusion relation holds between R_b and R_c.

In Fig. 8, we show the three-dimensional graphic of the nonlinear function estimated from the three fuzzy if-then rules in (29)-(31) using our inclusion-based fuzzy reasoning method with $\beta = 5$. This figure is the projection of the estimated nonlinear function with three input variables to the $(x_2 \text{ - } x_3 \text{ - } y)$ space by fixing the value of x_1 as $x_1 = 0$. Since the second rule R_b is fully compatible with any input vector with $x_1 = 0$, the weight of the most general rule R_a is always zero in Fig. 8. The output is *medium* in a large region of the input space from the same reason. Only when both x_2 and x_3 are *large*, the output is not *medium*. In this case, the output is about 0.75, which is the result of the interpolation between the consequent linguistic value *medium* of R_b and *large* of R_c. One may think that the output should be *large* for a *large* x_2 and a *large* x_3 because the most specific rule R_c is fully compatible with the input vector.

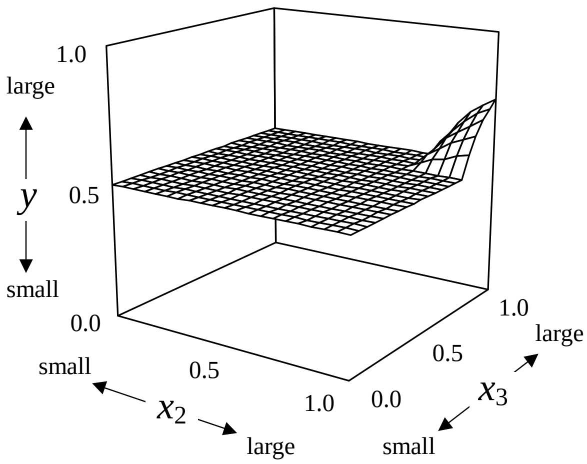


Fig. 8. Projection to the $(x_2 \text{ - } x_3 \text{ - } y)$ space of the nonlinear function obtained by the inclusion-based fuzzy reasoning method.

3.2

Preference order-based fuzzy reasoning

Since our inclusion-based fuzzy reasoning method uses the preference order defined by the strict inclusion relation, it cannot take into account the difference in specificity levels of fuzzy if-then rules when no inclusion relation holds among them. In this subsection, we extend our method to the case of an arbitrarily given preference order. For this purpose, we assume the existence of a rule hierarchy in our rule base. The inclusion relation is an example of the preference order that specifies the rule hierarchy.

Let us denote the preference order between fuzzy if-then rules by "$\prec$" where $R_k \prec R_q$ means that R_q is preferred to R_k (i.e., R_q dominates R_k) in fuzzy reasoning. Note that $R_k \prec R_q$ corresponds to $R_k \supset R_q$ in the inclusion-based fuzzy reasoning method in Section 2 where specific rules is preferred to general rules. The transitivity should hold for the preference order "$\prec$": If $R_1 \prec R_2$ and $R_2 \prec R_3$ then $R_1 \prec R_3$. Using the preference order, we define the rule weight of each fuzzy if-then rule R_k for the input vector $\mathbf{x}$ as

$$w(R_k, \mathbf{x}) = \prod_{\substack{q=1 \\ R_k \prec R_q}}^{N} (1 - \mu_q(\mathbf{x})) . \tag{32}$$

In this formulation, the weight of R_k is discounted when the input vector $\mathbf{x}$ is compatible with other rules R_q's that are preferred to R_k. When there is no rule that is preferred to R_k, the weight $w(R_k, \mathbf{x})$ is defined as $w(R_k, \mathbf{x}) = 1$ because the weight of R_k should not be discounted. The estimated output $\hat{y}(\mathbf{x})$ is calculated in the same manner as in Section 2 (i.e., by (25)).

For illustrating our preference order-based fuzzy reasoning method, let us consider the fuzzy if-then rules in (29)-(31) again. Among these rules, the inclusion relations $R_a \supset R_b$ and $R_a \supset R_c$ hold, but no inclusion relation holds between R_b and R_c. Thus $R_a \prec R_b$ and $R_a \prec R_c$ were used in our inclusion-based fuzzy reasoning method for depicting Fig. 8. If we use the rule hierarchy $R_a \prec R_b \prec R_c$, the estimated output $\hat{y}(\mathbf{x})$ is calculated from the three fuzzy if-then rules by our preference order-based fuzzy reasoning method with $\beta = 5$ as shown in Fig. 9. In this figure, the estimated output $\hat{y}(\mathbf{x})$ is *large* for a *small* x_1, a *large* x_2 and a *large* x_3. This is because the most specific rule R_c in (31) is most preferred in fuzzy reasoning. As shown by Fig. 9, our preference order-based fuzzy reasoning method can calculate the estimated output value from the fuzzy if-then rules based on the given rule hierarchy.

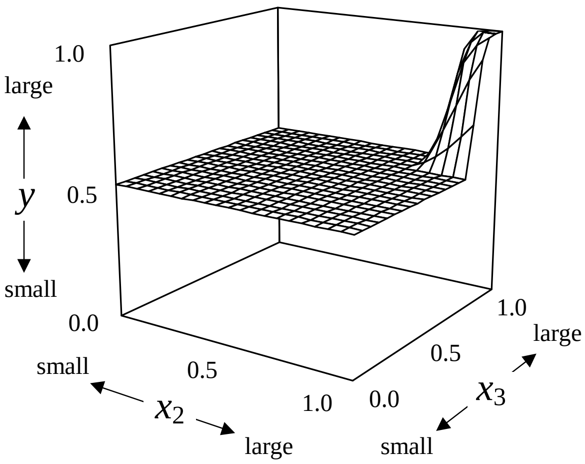


Fig. 9. Projection to the $(x_2 - x_3 - y)$ space of the nonlinear function obtained by the preference order-based fuzzy reasoning method.

4

Fuzzy rule generation from numerical data

Fuzzy if-then rules are usually obtained from human experts as linguistic knowledge or from numerical data through inductive learning algorithms. When a tabular form (e.g., Fig. 1) is used in rule generation for low-dimensional problems, no inclusion relation holds among fuzzy if-then rules. For high-dimensional problems, clustering-based methods are often used for fuzzy rule extraction from numerical data. No inclusion relation holds among extracted rules because each rule is located at a different position from other rules by clustering algorithms. When decision tree-based methods are used, the input space is divided into some fuzzy subspaces with a tree structure. In this case, no inclusion relation holds, either. As we can see from these discussions, usually there exists no inclusion relation among fuzzy if-then rules generated by existing inductive learning algorithms. This means that our inclusion-based fuzzy reasoning method has no effect on fuzzy reasoning (i.e., it is reduced to standard interpolation-based fuzzy reasoning).

In this section, we propose a genetics-based machine learning (GBML) algorithm with our inclusion-based fuzzy reasoning method for finding a small number of fuzzy if-then rules with different specificity levels from numerical data. In our GBML algorithm, some fuzzy if-then rules may have many *don't care* conditions and others may have only a few (or no) *don't care* conditions. The use of *don't care* conditions is essential for finding a small number of fuzzy if-then rules. This trick was used in some GBML algorithms (Castillo et al. (2001), Castro et al. (2001), Ishibuchi et al. (2001a)) for handling high-dimensional pattern classification problems. In those algorithms, the preference for more specific rules was not explicitly taken into account.

4.1

Problem formulation

Let us assume that we have *m* input-output pairs $(\mathbf{x}_p,\ y_p)$, $p = 1,2,...,m$ obtained from an unknown nonlinear function $y = y(\mathbf{x})$ with *n* input variables where $\mathbf{x}_p = (x_{p1},\ x_{p2},\ ...,\ x_{pn})$. For simplicity of explanation, the input and output spaces are assumed to be an *n*-dimensional unit cube $[0,1]^n$ and a unit interval $[0,1]$, respectively. We also assume that a set of linguistic values is given for describing each input (and output) variable. We explain our GBML algorithm using the five linguistic values in Fig. 1. Our task is to find a compact rule set *S* for linguistically describing the unknown nonlinear function $y = y(\mathbf{x})$.

The performance of the rule set *S* is measured by its approximation ability, its compactness, and the simplicity of fuzzy rules in *S*. We evaluate the approximation ability of the rule set *S* by the total squared error for the given input-output pairs:

$$f_1(S) = \sum_{p=1}^{m} \{\hat{y}(\mathbf{x}_p) - y_p\}^2 / 2\,, \qquad (33)$$

where $\hat{y}(\mathbf{x}_p)$ is the estimated output by the rule set *S* for the input vector $\mathbf{x}_p$. When the estimated output $\hat{y}(\mathbf{x}_p)$ cannot be calculated (i.e., when there is no compatible fuzzy if-then rule with the input vector $\mathbf{x}_p$), a pre-specified penalty value is used as the difference between the estimated output $\hat{y}(\mathbf{x}_p)$ and the target output y_p. In our computational experiments, the penalty value is specified as

$$|\hat{y}(\mathbf{x}_p) - y_p| = 1, \qquad (34)$$

when $\hat{y}(\mathbf{x}_p)$ cannot be calculated. This penalty value is equal to the width of the output space $[0,1]$.

It is a troublesome task for human users to manually examine a large number of fuzzy if-then rules. It is much easier to understand a compact rule set with a few rules than a large rule set with many rules. Let $f_2(S)$ be the number of fuzzy if-then rules in the rule set *S*. The compactness of *S* is evaluated by $f_2(S)$.

It is not easy for human users to intuitively understand long fuzzy if-then rules with many antecedent conditions. The number of antecedent conditions in each fuzzy if-then rule is referred to as the rule length. The simplicity of each fuzzy if-then rule is evaluated by the rule length. Let $f_3(S)$ be the total length of fuzzy if-then rules in the rule set *S*. The simplicity of fuzzy if-then rules in *S* is measured by $f_3(S)$.

Thus our rule extraction problem is formulated as follows:

$$\text{Minimize } f_1(S),\ f_2(S),\ f_3(S). \qquad (35)$$

4.2

GBML Algorithm

Our GBML algorithm simultaneously minimizes the three objectives in (35). Due to the existence of tradeoff, there is no absolutely optimal solution with respect to all the three objectives. We use the following scalar fitness function for handling our three-objective rule extraction problem in the framework of single-objective optimization:

$$fitness(S) = -w_1 \cdot f_1(S) - w_2 \cdot f_2(S) - w_3 \cdot f_3(S), \qquad (36)$$

where w_1, w_2 and w_3 are non-negative real numbers. The weights w_1, w_2 and w_3 should be specified according to the preference of human users with respect to the three objectives. We assume that the weight values are given by human users. When the weight values are not given, our problem will be handled in the framework of multi-objective optimization. In this case, a three-objective GBML algorithm will be used for finding a number of non-dominated rule sets with respect to the three objectives.

In our fuzzy GBML algorithm, the fuzzy if-then rule R_k in (1) is coded by its *n* antecedent and a single consequent linguistic values as $R_k = A_{k1}A_{k2} \cdots A_{kn}B_k$. A rule set *S* is denoted by a concatenated string where each substring of the length $(n+1)$ corresponds to a single fuzzy if-then rule in *S*. Initial rules are generated by randomly assigning an antecedent linguistic value (or *don't care*) to A_{ki} and a consequent linguistic value to B_k. Let us denote the five linguistic values in Fig. 1 by five integers as 1: *small*, 2: *medium small*, 3: *medium*, 4: *medium large*, and 5: *large*. We also use 0 for denoting *don't care* (i.e., 0: *don't care*). Thus the antecedent linguistic value A_{ki} is denoted by one of the six integers {0, 1, 2, 3, 4, 5} while the consequent linguistic value B_k is one of the five integers {1, 2, 3, 4, 5}.

From the current population, two parent strings are selected according to their fitness values. We use the roulette wheel selection with the linear scaling for specifying the selection probability $P(S)$ of each string *S*:

$$P(S) = \frac{fitness(S) - f_{\min}(\Psi)}{\sum_{S \in \Psi}(fitness(S) - f_{\min}(\Psi))}, \tag{37}$$

where Ψ is the current population and $f_{\min}(\Psi)$ is the smallest fitness value of rule sets in the current population Ψ.

Since the number of fuzzy if-then rules should be minimized in our fuzzy GBML algorithm, the string length is not fixed. The number of fuzzy if-then rules is modified by a crossover operation, which generates a new string whose length is different from its parent strings. We use a kind of one-point crossover with different cutoff points illustrated in Fig. 10 where R_k denotes a substring of the length $(n+1)$. One of the two children in Fig. 10 is randomly selected as an offspring. This crossover operation is applied to each pair of selected parents with a pre-specified crossover probability. When the crossover operation is not executed, one of the two parents is randomly chosen and handled as an offspring.

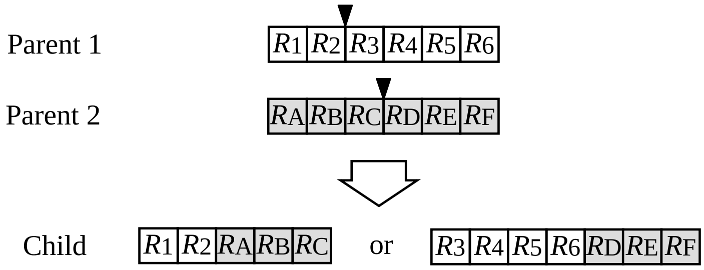


Fig. 10. A kind of one-point crossover with different cutoff points.

The crossover operation generates a new combination of existing fuzzy if-then rules as shown in Fig. 10. The modification of each fuzzy if-then rule is performed by a mutation operation. Our mutation operation randomly replaces each antecedent (and consequent) linguistic value with another one. After the crossover operation, this mutation operation is applied to each linguistic value with a pre-specified mutation probability. We also use a different kind of mutation, which randomly removes fuzzy if-then rules from each string. This mutation operation is applied to each rule with a pre-specified mutation probability.

The selection, crossover, and mutation operations are iterated for generating a new population. Let N_{pop} be the population size (i.e., the number of strings in each population). The genetic operations are iterated in each generation $(N_{\text{pop}} - 1)$ times for generating $(N_{\text{pop}} - 1)$ new strings as a new population. The best string with the largest fitness value in the previous population is added to the new population with no modifications as an elite individual. The genetic operations are applied to the new population in the same manner. The generation update is iterated until a pre-specified stopping condition is satisfied. Outline of our GBML algorithm is written as follows:

***Step 1*:** Randomly generate a number of rule sets as an initial population.

***Step 2*:** Repeat the following procedures for generating a new population.

(a) Select a pair of parent strings from the current population.

(b) Generate a new string from the selected pair of parents by a crossover operation.

(c) Apply two mutation operations to the generated strings by the crossover operation.

Step 3**:** Add the best string in the previous population to the newly generated population.

Step 4**:** If a pre-specified stopping condition is not satisfied, return to Step 2.

4.3

Results of computational experiments

As a test problem, we used the nonlinear function in Fig. 2. In the same manner as Section 2, we generated 441 input-output pairs (x_{p1}, x_{p2}, y_p), $p = 1, 2, ..., 441$ where $x_{pi} =$ 0.00, 0.05, ..., 1.00 for $i = 1, 2$. Our GBML algorithm with the inclusion-based fuzzy reasoning method was applied to this test problem using the following parameter specifications:

Weights in the fitness function: $(w_1, w_2, w_3) = (50, 1, 1)$,

Population size: 100,

The number of fuzzy if-then rules in each initial string: 10,

Crossover probability: 0.8,

Mutation probability for replacing each linguistic value with another one: 0.1,

Mutation probability for removing each fuzzy if-then rule: 0.1,

Termination condition: 10000 generations,

Parameter in the inclusion-based fuzzy reasoning method: $\beta = 1$.

This computational experiment was iterated 20 times using different initial populations. In all the 20 trials, our GBML algorithm found the rule set of the three fuzzy if-then rules R_A, R_B and R_C in (2)-(4). The nonlinear function in Fig. 2 is approximated by the three fuzzy if-then rules very well: $f_1(S) = 0.054$. As mentioned in Section 1 and Section 2, the three fuzzy if-then rules R_A, R_B and R_C in (2)-(4) are consistent with our intuitive understanding of the three-dimensional graphic in Fig. 2.

For comparison, we also applied our GBML algorithm with the standard interpolation-based fuzzy reasoning method to the same test problem 20 times. In 19 out of 20 trials, our GBML algorithm found the rule set of the following five fuzzy if-then rules:

y is *small*, (38)

If x_1 is *small* then y is *large*, (39)

If x_1 is *medium small* then y is *small*, (40)

If x_1 is *small* and x_2 is *small* then y is *large*, (41)

If x_1 is *small* and x_2 is *small* then y is *large*. (42)

While the nonlinear function in Fig. 2 is approximated well by these five fuzzy if-then rules using the standard interpolation-based fuzzy reasoning method (i.e., $f_1(S) = 0.078$), the second rule in (39) is not consistent with the actual three-dimensional shape of the nonlinear function in Fig. 2. Moreover the last two rules are exactly the same. This rule was selected twice by our GBML algorithm for realizing the preference for more specific rules in the framework of the standard interpolation-based fuzzy reasoning method.

In the same manner, we also applied our GBML algorithm to the nonlinear function in Fig. 3 using the standard interpolation-based fuzzy reasoning method 20 times. In all the 20 trials, our GBML algorithm found the rule set of the three fuzzy if-then rules R_A, R_B and R_C in (2)-(4). While the nonlinear function in Fig. 3 is perfectly approximated by the three fuzzy if-then rules using the standard interpolation-based fuzzy reasoning method, the extracted knowledge about the nonlinear function is misleading. For example, the fuzzy if-then rule R_C says that "If x_1 is *small* and x_2 is *small* then y is *large*". As we have already explained using the confidence measure in Section 2, this rule is not consistent with the actual three-dimensional shape of the nonlinear function in Fig. 3. The actual output is *medium* when x_1 is *small* and x_2 is *small* in Fig. 3. This result suggests that misleading knowledge can be obtained from numerical data when general and specific rules are simultaneously obtained from numerical data using standard interpolation-based fuzzy reasoning.

We also performed the same computational experiment using the inclusion-based fuzzy reasoning method 20 times. Our GBML algorithm found the following three fuzzy if-then rules for the nonlinear function in Fig. 3 in all the 20 trials.

y is *small*, (43)

If x_1 is *small* then y is *medium small*, (44) If x_1 is *small* and x_2 is *medium* then *y* is *medium*. (45)

We can see that these fuzzy if-then rules are consistent with our intuitive understanding of the three-dimensional graphic in Fig. 3. That is, intuitively acceptable knowledge was extracted from the nonlinear function in Fig. 3 using the inclusion-based fuzzy reasoning method while counterintuitive (or misleading) knowledge was obtained for the same task using standard interpolation-based fuzzy reasoning.

4.4

Some heuristics in our GBML algorithm

In our computational experiments in the previous subsection, the total number of possible fuzzy if-then rules was $(5+1)\times(5+1)\times 5 = 180$ because we used the five linguistic values and *don't care* for the two input variables and the five linguistic values for the output variable. Thus the task of our GBML algorithm can be viewed as choosing a small number of fuzzy if-then rules from the 180 candidate rules. The number of candidate fuzzy if-then rules exponentially increases with the dimensionality of the input space. For example, the total number of fuzzy if-then rules is $6^5 \times 5 = 38880$ in the case of five input variables. In this case, the total number of rule sets is $2^{38880} \cong 1.1\times 10^{11704}$. Since our GBML algorithm has such a huge search space in its application to high-dimensional problems, some heuristics are required for efficiently finding a small number of fuzzy if-then rules. In this subsection, we describe such heuristics.

While initial fuzzy if-then rules were randomly generated in the previous subsection, they can be also generated from input-output pairs. When we generate *N* initial rules, we randomly choose *N* input-output pairs. A single fuzzy if-then rule is generated from each input-output pair by choosing the most compatible linguistic value with each input (and output) value. For generating more general rules, each antecedent linguistic value is replaced with *don't care* using a pre-specified replacement probability (e.g., 0.5 in our computational experiments).

Our task is to find a small number of fuzzy if-then rules. So we give an upper bound to the number of fuzzy if-then rules included in each string (e.g., 20 in our computational experiments). When more fuzzy if-then rules are included in a string than the upper bound, excess rules are removed from the right side of the string until the number of rules becomes the same as the upper bound. This procedure is applied to all strings after the crossover operation.

Since genetic operations do not take into account the given input-output pairs, generated fuzzy if-then rules do not always have appropriate consequent linguistic values. So we replace the consequent linguistic value of each fuzzy if-then rule with more appropriate one using the information on the given input-output pairs. More specifically, we replace the consequent linguistic value B_k of the fuzzy if-then rule R_k with B^i using the following probability:

$$P(B^i) = \frac{Confidence(\mathbf{A}_k \Rightarrow B^i)}{\sum_{j=1}^{5} Confidence(\mathbf{A}_k \Rightarrow B^j)}, \quad (46)$$

where $\mathbf{A}_k$ is the antecedent part of the fuzzy if-then rule R_k, B^i is one of the five consequent linguistic values, and $Confidence(\mathbf{A}_k \Rightarrow B^i)$ is the confidence value of the fuzzy if-then rule with the antecedent part $\mathbf{A}_k$ and the consequent linguistic value B^i. This procedure is applied to each linguistic rule with a pre-specified probability (e.g., 0.5 in our computational experiments) after the two mutation operations.

The heuristic procedure for generating initial fuzzy if-then rules can be also used in the generation update phase. For each rule set, we first identify the input-output pair with the maximum error. Then we generate a fuzzy if-then rule from the identified input-output pair in the same manner as the heuristic procedure for generating initial rules. This procedure is applied to each rule set with a pre-specified probability (e.g., 0.5 in our computational experiments) after the above-mentioned replacement procedure.

For examining the effect of the four heuristics (i.e., heuristic initial population, upper bound on the number of rules, replacement of consequent linguistic values, and heuristic generation of new rules) on the search ability of our GBML algorithm, we applied it to a function approximation problem with five input variables. This test problem was generated by adding three dummy variables x_3, x_4 and x_5 to the 441 input-output pairs generated from Fig. 2. The value of each dummy variable was randomly specified in the unit interval $[0,1]$. Thus we have 441 input-output pairs in the form of $(x_{p1}, ..., x_{p5}, y_p)$ where $x_{pi} = 0.00, 0.05, ..., 1.00$ for $i = 1, 2$ and x_{pi} is a random real number in $[0,1]$ for $i = 3,4,5$. Our task is to extract a small number of fuzzy if-then rules from the given input-output pairs with the three dummy variables.

For this task, we used fuzzy if-then rules of the following form:

$$\text{If } x_1 \text{ is } A_{k1} \text{ and } x_2 \text{ is } A_{k2} \text{ ... and } x_5 \text{ is } A_{k5} \text{ then } y \text{ is } B_k . \quad (47)$$

In the same manner as the previous subsection, we applied our GBML algorithm with the four heuristics to the test problem with the five input variables 20 times. In all the 20 trials, our GBML algorithm found the rule set of the three fuzzy if-then rules R_A, R_B and R_C in (2)-(4). The three dummy input variables x_3, x_4 and x_5 were always identified correctly. We also performed the same computational experiments with no heuristics. The rule set of the three fuzzy if-then rules in (2)-(4) was correctly found in 13 out of 20 trials. For comparing the search ability of our GBML algorithm with/without the four heuristics, we examined when the optimal rule set of the three fuzzy if-then rules in (2)-(4) was found during the execution of our GBML algorithm in each trial. Table 1 shows the relation between the number of generations and the number of successful trials where the optimal rule set was found. For example, the optimal rule set was found in 12 out of 20 trials during the first 2000 generations when we used the four heuristics. From Table 1, we can see that the use of the four heuristics significantly improved the search ability of our GBML algorithm.

Table 1. The number of successful trials where the optimal rule set was found during the specified number of generations.

Number of generations	2000	4000	6000	8000	10000
With heuristics	12	19	20	20	20
Without heuristics	3	5	11	13	13

5

Fuzzy reasoning for fuzzy number-valued function approximation

We have already explained the inclusion-based fuzzy reasoning method for approximately realizing nonlinear functions by a small number of fuzzy if-then rules. In this section, we discuss the approximate realization of fuzzy number-valued nonlinear functions. A fuzzy number-valued nonlinear function $\tilde{y} = \tilde{y}(\mathbf{x})$ maps an *n*-dimensional non-fuzzy input vector **x** to a fuzzy number $\tilde{y}$. For describing the fuzzy number-valued nonlinear function $\tilde{y} = \tilde{y}(\mathbf{x})$, we use the following fuzzy if-then rules:

Rule R_k : If x_1 is A_{k1} and ... and x_n is A_{kn} then y is B_k , $k = 1,2,...,N$. (48)

These fuzzy if-then rules are the same as those used for describing the real number-valued nonlinear function $y = y(\mathbf{x})$ in the previous sections.

5.1

Interpolation-based fuzzy reasoning for fuzzy number-valued function approximation

The standard interpolation-based fuzzy reasoning method in (8) can be directly extended for approximately realizing the fuzzy number-valued nonlinear function $\tilde{y} = \tilde{y}(\mathbf{x})$ as

$$\hat{\tilde{y}}(\mathbf{x}) = \frac{\sum_{k=1}^{N} \mu_k(\mathbf{x}) \cdot B_k}{\sum_{k=1}^{N} \mu_k(\mathbf{x})} . \qquad (49)$$

This formulation was referred to as “weighted average of fuzzy sets” and examined in Uehara (1994). This formulation involves fuzzy arithmetic (Kaufmann & Gupta (1985)) on linguistic values. While the numerical calculation of fuzzy arithmetic is performed by interval arithmetic (Moore (1979), Alefeld &

Herzberger (1983)) on level sets of fuzzy numbers in general, it is easily performed when the consequent linguistic values have membership functions of a special form (e.g., triangular and trapezoidal).

For illustrating the interpolation-based fuzzy reasoning method in (49), we calculated a fuzzy number-valued nonlinear function realized by the 25 fuzzy if-then rules in Fig. 1. The estimated fuzzy output $\hat{\tilde{y}}(\mathbf{x})$ is shown in Fig. 11 for $\mathbf{x} = (0, 0), (0.05, 0.05), ..., (0.25, 0.25)$. When $\mathbf{x} = (0, 0)$, the estimated fuzzy output is *large* as shown in Fig. 11. This corresponds to the bottom-left fuzzy if-then rule in Fig. 1. As the input vector increases from (0, 0) to (0.25, 0.25), the estimated fuzzy output decreases from *large* to *small* as shown in Fig. 11.

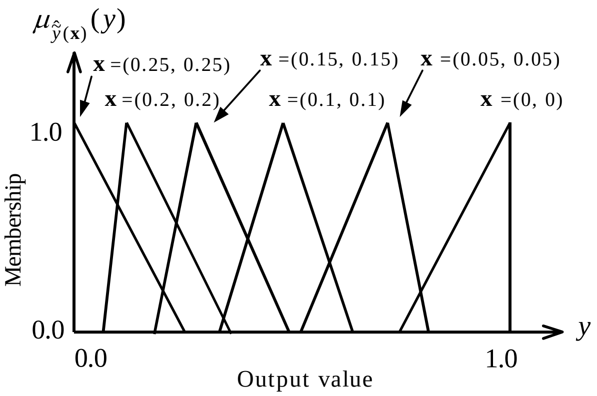


Fig. 11. Estimated fuzzy outputs from the 25 fuzzy if-then rules in Fig. 1 using the interpolation-based fuzzy reasoning method.

As in the case of the approximate realization of real number-valued nonlinear functions in Subsection 2.1, the interpolation-based fuzzy reasoning method in (49) leads to counterintuitive results. Let us consider the three fuzzy if-then rules R_A, R_B and R_C in (2)-(4) again. The estimated fuzzy output for the input vector $\mathbf{x} = (0, 0)$ is calculated by the interpolation of the consequent linguistic values *small*, *medium* and *large* of the three fuzzy if-then rules because they are fully compatible with the input vector **x**. As a result, the estimated fuzzy output is not *large* but similar to *medium*. In Fig. 12, we show the estimated fuzzy outputs for $\mathbf{x} = (0, 0), (0.05, 0.05), ..., (0.25, 0.25)$.

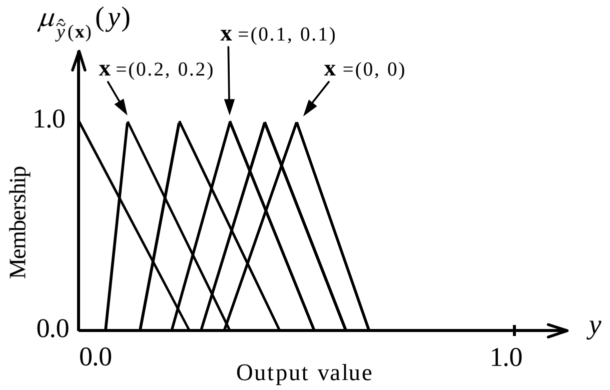


Fig. 12. Estimated fuzzy outputs from the three fuzzy if-then rules in (2)-(4) using the interpolation-based fuzzy reasoning method.

5.2

Inclusion-based fuzzy reasoning for fuzzy number-valued function approximation

Our inclusion-based fuzzy reasoning method in Section 2 (and the preference order-based fuzzy reasoning method in Section 3) can be used for approximately realizing fuzzy number-valued nonlinear functions. Our formulation is modified as

$$\hat{y}(\mathbf{x}) = \frac{\sum_{k=1}^{N} \left(w(R_k, \mathbf{x})\right)^{\beta} \cdot \mu_k(\mathbf{x}) \cdot B_k}{\sum_{k=1}^{N} \left(w(R_k, \mathbf{x})\right)^{\beta} \cdot \mu_k(\mathbf{x})}, \tag{50}$$

where the weight $w(R_k, \mathbf{x})$ is specified in the same manner as in Section 2 and Section 3.

Let us apply our inclusion-based fuzzy reasoning method to the three fuzzy if-then rules in (2)-(4). The estimated fuzzy output is calculated as *large* for the input vector $\mathbf{x} = (0, 0)$. In Fig. 13, we show the estimated fuzzy outputs for $\mathbf{x} = (0, 0), (0.05, 0.05), ..., (0.25, 0.25)$ using our inclusion-based fuzzy reasoning method with $\beta = 1$. From the comparison between Fig. 11 and Fig. 13, we can see that almost the same results were obtained in those figures. This observation shows that the three fuzzy if-then rules in (2)-(4) play almost the same role in our inclusion-based fuzzy reasoning method as the 25 fuzzy if-then rules in Fig. 1.

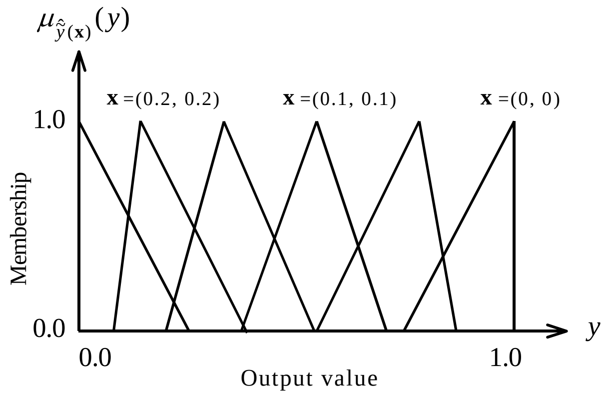


Fig. 13. Estimated fuzzy outputs from the three fuzzy if-then rules in (2)-(4) using the inclusion-based fuzzy reasoning method.

6

Concluding remarks

In this paper, we proposed an inclusion-based fuzzy reasoning method for describing nonlinear functions using a small number of fuzzy if-then rules. The motivation for our proposal is that counterintuitive results are often obtained from standard interpolation-based fuzzy reasoning when our fuzzy rule base is a mixture of general and specific rules. Our proposal is based on the preference for more specific rules, which is a widely accepted idea in the field of default reasoning. We demonstrated that intuitively acceptable results were obtained from our inclusion-based fuzzy reasoning method. Our method is applicable to various fuzzy reasoning models. We combined our method with a simplified version of the Takagi-Sugeno (TS) model because it has been frequently used in the literature. The magnitude of the preference for more specific rules can be controlled by a user-definable parameter β in our method. When $\beta = 0$, our method is reduced to standard interpolation-based fuzzy reasoning. The larger the value of β is, the larger the magnitude of the preference for specific rules is. That is, specific rules play a dominant role in our fuzzy reasoning method when β is very large.

For handling partially overlapping rules with different specificity levels, we generalized our inclusion-based method to a preference order-based method.

When a general rule does not perfectly include a specific rule, the preference for the specific rule over the general rule is not implemented in our inclusion-based method. For handling this situation, we assumed the existence of a rule hierarchy among fuzzy if-then rules in our preference order-based method. Such a rule hierarchy can be constructed by the specificity level of each rule (i.e., the length of the antecedent part of each rule). Partially overlapping rules are also handled in the framework of our inclusion-based method if we can appropriately define an inclusion grade between fuzzy if-then rules and appropriately combine it with our method. Such an extension of our method, which is left for future research, may be able to handle fuzzy if-then rules with different specificity levels more flexibly than the preference order-based fuzzy reasoning method. This is because the inclusion grade-based method uses a fuzzy relation over fuzzy if-then rules while the preference order-based method uses a pre-specified crisp relation.

We also described a genetics-based machine learning (GBML) algorithm for generating a small number of fuzzy if-then rules with different specificity levels from numerical data. This algorithm can simultaneously minimize the approximation error, the number of fuzzy if-then rules, and the number of antecedent conditions in each rule. Thus we can obtain a small number of simple fuzzy if-then rules with high comprehensibility as well as high approximation ability. Finally, we discussed the approximation of fuzzy number-valued nonlinear functions using fuzzy if-then rules. Our inclusion-based fuzzy reasoning method is applicable to the approximate realization of fuzzy number-valued nonlinear functions. Through computation experiments, we demonstrated that intuitively acceptable results were obtained from our inclusion-based method while standard fuzzy reasoning led to counterintuitive results.

The main advantage of our inclusion-based fuzzy reasoning method over traditional interpolation-based schemes is its consistency with our intuition: Intuitively acceptable reasoning results are always obtained even when our fuzzy rule-based system is a mixture of general and specific fuzzy rules overlapping with each other. This was discussed in this paper through illustrative examples where counterintuitive results were obtained from traditional interpolation-based schemes. Another advantage is the possibility to significantly decrease the number of fuzzy rules by using general rules. This was also discussed in this paper through illustrative examples. For example, a non-linear function in Fig. 2

generated by 25 fuzzy rules in Fig. 1 can be represented by only three fuzzy rules. One important future research topic is to examine the performance of our inclusion-based fuzzy reasoning method through computational experiments on real-world test problems in terms of the approximation ability and the generalization ability of fuzzy rule-based systems. For such a computational experiment, some test problems are available in the literature (for example, see Roubos & Babuska (2003) and Cordon & Herrera (2003)).

Acknowledgment

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

References

Agrawal R, Mannila H, Srikant R, Toivonen H, Verkamo AI (1996) Fast discovery of association rules. In: U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (eds) Advances in Knowledge Discovery and Data Mining. AAAI Press, Metro Park: pp 307-328

Alefeld G, Herzberger J (1983) Introduction to Interval Computations, Academic Press, New York

Bacchus F, Grove AJ, Halpern JY, Koller D (1996) From statistical knowledge bases to degrees of belief, Artificial Intelligence **87**(1-2): 75-143

Bien Z, Yu W (1995) Extracting core information from inconsistent fuzzy control rules, Fuzzy Sets and Systems **71**(1): 95-111

Brewka G, Eiter T (2000) Prioritizing default logic. In: S.Hölldobler (ed) Intellectics and Computational Logic. Kluwer Academic Publishers, Dordrecht, pp 27-45

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

Castro JL, Castro-Schez JJ, Zurita JM (2001) Use of a fuzzy machine learning technique in the knowledge acquisition process, Fuzzy Sets and Systems, **123**(3): 307-320

Cordon O, Herrera F (2003) Author's reply, IEEE Trans. on Fuzzy Systems, **11**(6): 866-869

Cordon O, Herrera F, Peregrin A (1997) Applicability of the fuzzy operators in the design of fuzzy logic controllers, Fuzzy Sets and Systems, **86**(1): 15-41

Delgrande JP, Schaub T (2000) Expressing preferences in default logic, Artificial Intelligence, **123**(1-2): 41-87

Dubois D, Lang J, Prade H (1994a) Possibilistic logic. In: Gabbay DM, Hogger CJ, and Robinson JA(eds) Handbook of Logic in Artificial Intelligence and Logic Programming - Volume 3: Nonmonotonic Reasoning and Uncertain Reasoning. Oxford University Press, Oxford, pp 439-513

Dubois D, Lang J, Prade H (1994b) Automated reasoning using possibilistic logic: Semantics, brief revision, and variable certainty weights, IEEE Trans. on Knowledge and Data Engineering, **6**(1): 64-71

Dubois D, Prade H (1987) The principle of minimum specificity as a basis for evidential reasoning. In: B. Bouchon and R. R. Yager (Eds) Uncertainty in Knowledge Based Systems (Lecture Notes in Computer Science 286). Springer-Verlag, Berlin, pp 75-84

Dubois D, Prade H (1988) Default reasoning and possibility theory, Artificial Intelligence, **35**(2): 243-257

Dubois D, Prade H (1996) What are fuzzy rules and how to use them, Fuzzy Sets and Systems, **84**(2): 169-185

Dung PM, Son TC (2001) An argument-based approach to reasoning with specificity, Artificial Intelligence, **133**(1-2): 35-85

Emami MR, Turksen IB, Goldenberg AA (1999) A unified parameterized formulation of reasoning in fuzzy modeling and control, Fuzzy Sets and Systems, **108**(1): 59-81

Goldszmidt M, Pearl J (1996) Qualitative probabilities for default reasoning, belief revision, and causal modeling, Artificial Intelligence, **84**(1-2): 57-112

Hoffmann F, Pfister G (1996) Learning of a fuzzy control rule base using messy genetic algorithm. In: Herrera F and Verdegay JL (eds) Genetic Algorithm and Soft Computing. Physica-Verlag, Heidelberg, pp 279-305

Hoffmann F, Pfister G (1997) Evolutionary design of a fuzzy knowledge base for a mobile robot, International Journal of Approximate Reasoning, **17**(4): 447-469

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

Ichihashi H (1991) Iterative fuzzy modeling and a hierarchical network, Proc. of 4th International Fuzzy System Association World Congress - Engineering, Brussels, Belgium, pp 49-52

Ishibuchi H (1999a) Fuzzy reasoning method for handling fuzzy rules with different specificity levels, Proc. of 18th International Conference of the North American Fuzzy Information Processing Society, New York, USA, pp 110-114

Ishibuchi H (1999b) Fuzzy reasoning method in fuzzy rule-based systems with general and specific rules for function approximation, Proc. of 8th IEEE International Conference on Fuzzy Systems, Seoul, Korea, pp 198-203

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

Ishibuchi H, Yamamoto T, Nakashima T (2001b) Fuzzy data mining: Effect of fuzzy discretization, Proc. of 1st IEEE International Conference on Data Mining, San Jose, USA, pp 241-248

Kaufmann A, Gupta MM (1985) Introduction to Fuzzy Arithmetic, Van Nostrand Reinhold, New York

Knorr EM, Ng RT (1999) Finding intentional knowledge of distance-based outliers, Proc. of 25th International Conference on Very Large Data Bases, Edinburgh, Scotland, pp 211-222

Knorr EM, Ng RT, Tucakov V (2000) Distance-based outliers: Algorithms and applications, International Journal on Very Large Data Bases, **8**(3): 237-253

Leondes CT (ed.) (1999) Fuzzy Theory Systems: Techniques and Applications (1-4). Academic Press, San Diego

Mees W (1999) Detection of defects in a fuzzy knowledge base, Proc. of 8th IEEE International Conference on Fuzzy Systems, Seoul, Korea, pp 204-209

Moore RE (1979) Methods and Applications of Interval Analysis, SIAM Studies in Applied Mathematics, Philadelphia

Poole D (1991) The effect of knowledge on belief: conditioning, specificity and the lottery paradox in default reasoning, Artificial Intelligence, **49**(1-3): 281-307

Reiter R (1980) A logic for default reasoning, Artificial Intelligence, **13**(1-2): 81-132

Roubos JA, Babuska R (2003) Comments on the benchmarks in "a proposal for improving the accuracy of linguistic modeling" and related articles, IEEE Trans. on Fuzzy Systems, 11(6) 861- 865

Russell SJ, Norvig P (1995) Artificial Intelligence: A Modern Approach, Prentice-Hall, Upper Saddle River

Suzuki E (2000) Hypothesis-driven discovery of exception rules based on thresholds scheduling, Journal of Japanese Society for Artificial Intelligence, **15**(4): 649-656 (in Japanese)

Suzuki E, Kodratoff Y (1998) Discovery of surprising exception rules based on intensity of implication. In: Zytkow JM and Quafafou M (eds) Principles of Data Mining and Knowledge Discovery (PKDD) (Lecture Notes in Computer Science 1510). Springer-Verlag, Heidelberg

Takagi T, Sugeno M (1985) Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. on Systems, Man, and Cybernetics, **15**(1): 116-132

Uehara K (1994) Fuzzy inference based on a weighted average of fuzzy sets and its learning algorithm for fuzzy exemplars, Proc. of International Joint Conference of 4th IEEE International Conference on Fuzzy Systems and 2nd International Fuzzy Engineering Symposium, Yokohama, Japan, pp 2253-2260

Viaene S, Wets G, Vanthienen J (2000) A synthesis of fuzzy rule-based system verification, Fuzzy Sets and Systems, **113**(2): 253-265

Yager RR (1987a) Using approximate reasoning to represent default knowledge, Artificial Intelligence, **31**(1): 99-112

Yager RR (1987b) Possibilistic qualification and default rules. In: Bouchon B and Yager RR (eds) Uncertainty in Knowledge Based Systems (Lecture Notes in Computer Science 286). Springer-Verlag, Berlin, pp 41-57

Yager RR (1988a) A generalized view of non-monotonic knowledge: A set of theoretic perspective, International Journal of General Systems, **14**: 251-265

Yager RR (1988b) A mathematical programming approach to inference with the capability to implement default rules, International Journal of Man-Machine Studies, **29**: 685-714

Yager RR (1993) On a hierarchical structure for fuzzy modeling and control, IEEE Trans. on Systems, Man, and Cybernetics, **23**(4): 1189-1197

Yager RR, Larsen HL (1991) On discovery potential inconsistencies in validating uncertain knowledge bases by reflecting on the input, IEEE Trans. on Systems, Man, and Cybernetics, **21**(4): 790-801