GENERATING FUZZY RULES FROM RELATIONAL DATABASE SYSTEMS FOR ESTIMATING NULL VALUES

SHYI-MING CHEN
MING-SHIOW YEH

Department of Computer and Information Science National Chiao Tung University, Hsinchu, Taiwan, Republic of China

This paper presents a new algorithm for constructing fuzzy decision trees from relational database systems and generating fuzzy rules from the constructed fuzzy decision trees. We also present a method for dealing with the completeness of the constructed fuzzy decision trees. Based on the generated fuzzy rules, we also present a method for estimating null values in relational database systems. The proposed methods provide a useful way to estimate null values in relational database systems.

Kandel (1986) pointed out that during the past decade the research fields of applied computer science (e.g., information processing, artificial intelligence, knowledge processing in expert systems) have established the need for formulation of models of imprecise information systems that would simulate human approximate reasoning. Since Zadeh (1965) proposed the fuzzy set theory, the theory has been widely used for representing and reasoning with imprecise and uncertain informa-

This work was supported in part by the National Science Council, Republic of China, under grant NSC 87-2213-E-009-011.

Address correspondence to Professor Shyi-Ming Chen, Ph.D., Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

Cybernetics and Systems: An International Journal, 28:695-723, 1997

0196-9722/97 $12.00 + .00

tion. Roughly speaking, a fuzzy set is a set with fuzzy boundaries. Let **A** be a fuzzy set of the universe of discourse V, $V = \{v_1, v_2, \ldots, v_n\}$, and let $\mu_{\mathbf{A}}$ be the membership function of **A**, $\mu_{\mathbf{A}}: V \rightarrow [0, 1]$. Then $\mu_{\mathbf{A}}(v_i)$ indicates the degree of membership of v_i in **A**, where $1 \leq i \leq n$.

Traditionally, knowledge bases of rule-based systems are constructed by the process of knowledge acquisition. Some researchers have concentrated on generating rules by learning from examples (Jeng & Liang, 1993; Sudkamp & Hammell, 1994; Wang & Mendel, 1992; Yeh & Chen, 1995). Wang and Mendel (1992) use a fuzzy associative memory to construct a fuzzy rule base from numerical and linguistic information and apply it to the truck backer-upper control. Sudkamp and Hammell (1994) presented a method for learning fuzzy rules and used the generated fuzzy rules to determine the mapping from input space to output space. Hart (1985) applied the induction of decision trees to knowledge acquisition for expert systems, where experts supply various examples to construct decision trees and the resulting decision trees are used to generate rules.

Safavian and Landgrebe (1991) have made a survey of some methods for constructing decision trees from collections of examples. Yuan and Shaw (1995) pointed out that although the decision trees generated by these methods are useful in building knowledge-based expert systems, they often cannot properly express and handle the vagueness and ambiguity associated with human thinking and perception. To overcome these drawbacks, Quinlan (1987) suggested a probabilistic method for constructing decision trees as probabilistic classifiers, where inaccuracies of attribute values are treated as noise. Yuan and Shaw (1995) pointed out that the limitation of Quinlan's work (1987) is that the types of uncertainties arising in classification problems are not necessarily probabilistic, appearing as randomness or noise. Thus, Yuan and Shaw (1995) presented a fuzzy decision tree induction method, and they pointed out that fuzzy decision trees represent classification knowledge more natually and are more robust in tolerating imprecise, conflict, and missing information.

In this paper, we present a fuzzy concept learning system (FCLS) algorithm to construct fuzzy decision trees from relational database systems and to generate fuzzy rules from the constructed fuzzy decision trees. Furthermore, we present a method for dealing with the completeness of the constructed fuzzy decision trees. Based on the generated fuzzy rules, we also present a method for estimating null values in relational database systems. The proposed methods provide a useful way to estimate null values in relational database systems.

BASIC CONCEPTS OF GENERATING FUZZY RULES FROM RELATIONAL DATABASE SYSTEMS

The relational data model is the most popular data model of database systems used in commercial applications because it can be very easily understood and implemented. In the following, we will introduce the concepts of generating fuzzy rules from relational database systems. An example of a relation in a relational database system is shown in Figure 1, where **A**, **B**, . . . , and **C** are attributes and their values are in the interval $[0, 10]$. The relationship between the attributes **A**, **B**, and **C** can be defined as $\mathbf{C} = (\mathbf{A}^*\mathbf{B})/10$. That means that the values of attribute **C** are determined by the values of attributes **A** and **B**.

In order to generate fuzzy rules from a relational database system, we must use the concepts of fuzzy sets (Zadeh, 1965) and fuzzification (Wang & Mendel, 1992). The main purpose of the fuzzification process is to transfer the input crisp data into fuzzy data and incorporate the imprecision. If the attributes in a relation of a relational database system are considered as linguistic variables (Zadeh, 1975), then we must partition the input domains (i.e., the domains of attributes) into several fuzzy regions (linguistic terms) in advance, where a linguistic variable is a variable whose values are linguistic terms rather than numerical values. This means to define the corresponding membership functions of the linguistic terms (fuzzy regions) for each linguistic variable (attribute). Let **X** be a linguistic variable in a domain **V** and let $X_1, X_2, \ldots, X_n$ be their corresponding linguistic terms, where the membership function curves of the linguistic terms $X_1, X_2, \ldots, X_n$ are shown in Figure 2. From Figure 2, we can see that for every $v \in \mathbf{V}$, we can get one or two membership grades corresponding to different linguistic terms. For the example in Figure 2, we can get $\mu_{X_i}(v)$ and $\mu_{X_{i+1}}(v)$, where $\mu_{X_i}(v) + \mu_{X_{i+1}}(v) = 1$, $\mu_{X_i}(v) \in [0, 1]$, $\mu_{X_{i+1}}(v) \in [0, 1]$, and $1 \leq i \leq n - 1$. We also ensure that this partition must satisfy the ε-completion (Kandel 1986). In this paper, we let ε equal 0.5. This means that there exists at least one X_i such that $\mu_{X_i}(v) \geq 0.5$ for every $v \in \mathbf{V}$. In the process of constructing fuzzy decision trees, we transform the crisp value v into the singleton fuzzy set $\{X_i / \mu_{X_i}(v)\}$, where X_i is a linguistic term and $\mu_{X_i}(v) \geq 0.5$. If $\mu_{X_i}(v) = \mu_{X_{i+1}}(v) = 0.5$, we trans-

A	B	...	C
7	10	...	7
7	2	...	1.4
6	6	...	3.6
5	3	...	1.5

Figure 1. A relation in a relational database system.

form v into $\{X_i/\mu_{X_i}(v)\}$. For example, let us consider the relation shown in Figure 1. Assume that we would like to decompose the values of the attribute **A** into three fuzzy regions (linguistic terms): Low, Medium, and High, where the membership function curves of Low, Medium, High are shown in Figure 3, and **V** is the domain of the attribute **A**. In this case, {Low, Medium, High} are called the fuzzy domain of the attribute **A**. Through the fuzzification process, the relation shown in Figure 1 became the fuzzy relation shown in Figure 4.

If attribute **X** is an unfuzzifiable attribute, then we let each domain value of **X** be a singleton fuzzy set. For example, if the education degree of an employee is Bachelor, then we let the membership value of Bachelor be equal to 1.0; that is, the domain value Bachelor of the attribute Degree is fuzzified into {Bachelor/ 1.0}.

In the following, we introduce the concepts of deriving fuzzy decision trees from fuzzy relations. In a fuzzy decision tree, a nonterminal node is also called a decision node. There are two kinds of terminal nodes in a fuzzy decision tree, i.e., certainty factor (CF) nodes, denoted by ○, and hypothetical certainty factor (HCF) nodes, denoted by ⬚.

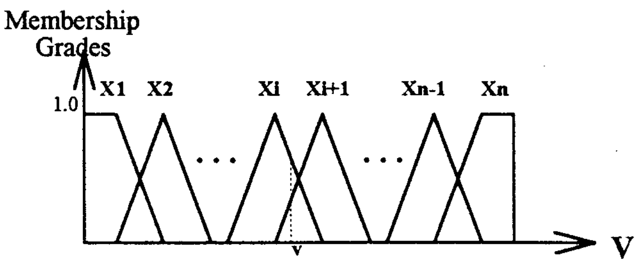


Figure 2. Fuzzy decomposition of domain V.

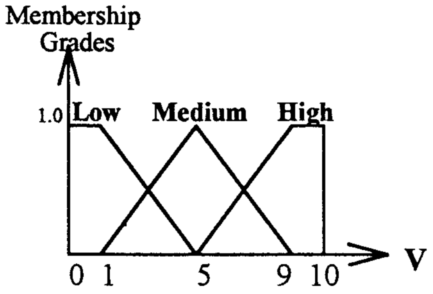


Figure 3. Membership function curves of linguistic terms "Low," "Medium," and "High."

The certainty factor nodes and the hypothetical certainty factor nodes are associated with values between zero and one. The path from the root node to each terminal node (i.e., certainty factor node or hypothetical certainty factor node) forms a fuzzy rule. Figure 5 shows an example of the fuzzy decision tree, where **X**, **Y**, **Z** are attributes in the relational database and X_i, Y_j, Z_k ($1 \leq i \leq n$, $1 \leq j \leq m$, $1 \leq k \leq p$) are linguistic terms, respectively. From Figure 5, we can see that the value of **Z** is determined by the values of **X** and **Y**. Note that if **X** or **Y** or **Z** is a linguistic variable, then X_i or Y_j or Z_k would be linguistic terms, respectively, where $1 \leq i \leq n$, $1 \leq j \leq m$, $1 \leq k \leq p$. In Figure 5, the certainty factor node CF_i indicates that there are some tuples in the relation shown in Figure 6 that satisfy the classification in a degree denoted by CF_i, where $CF_i \in [0, 1]$. A hypothetical certainty factor node HCF_x exists because there are no tuples that satisfy the classification and that are generated by tree growning. Consider the path $\mathbf{X} \xrightarrow{X_1} \mathbf{Y} \xrightarrow{Y_1} \mathbf{Z} \xrightarrow{Z_2} CF_j$ in the fuzzy decision tree shown in Figure 5. This path indicates that there is a fuzzy rule

$$\text{IF } \mathbf{X} \text{ is } X_1 \text{ and } \mathbf{Y} \text{ is } Y_1 \text{ THEN } \mathbf{Z} \text{ is } Z_2 \qquad (CF = CF_j) \tag{1}$$

A	B	...	C
{Medium/0.5}	{High/1.0}	...	{High/0.5}
{Medium/0.5}	{Low/0.75}	...	{Low/0.9}
{Medium/0.75}	{Medium/0.75}	...	{Medium/0.65}
{Medium/1.0}	{Medium/0.5}	...	{Low/0.88}

Figure 4. Fuzzification of the relation shown in Figure 1.

in the fuzzy rule base. A null path is a path whose terminal node is a hypothetical certainty factor node. For example, in Figure 5, the path $\mathbf{X} \xrightarrow{X_1} \mathbf{Y} \xrightarrow{Y_2} \mathbf{Z} \xrightarrow{Z_2} \mathrm{HCF}_x$ is a null path. A nonnull path is a path whose terminal node is a certainty factor node. For example, in Figure 5, the path $\mathbf{X} \xrightarrow{X_1} \mathbf{Y} \xrightarrow{Y_1} \mathbf{Z} \xrightarrow{Z_1} \mathrm{CF}_i$ is a nonnull path. Figure 7 shows an example of a subtree of a fuzzy decision tree constructed by the fuzzy relation shown in Figure 4.

In the following, we introduce the concepts of generating fuzzy rules from the generated fuzzy decision tree. Generally speaking, the form of a fuzzy rule is as follows:

$$\text{IF } \mathbf{X} \text{ is } X_i \text{ AND } \mathbf{Y} \text{ is } Y_j \text{ THEN } \mathbf{Z} \text{ is } Z_k \qquad (\mathrm{CF} = \mathrm{c}) \tag{2}$$

where

1. $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ are linguistic variables and X_i, Y_j, Z_k are linguistic terms represented by fuzzy sets.
2. c is the certainty factor (CF) value of the fuzzy rule indicating the degree of belief of the rule, where $c \in [0, 1]$.

Because in a fuzzy decision tree, the path from the root node to each terminal node (certainty factor node or hypothetical certainty factor node) forms a fuzzy rule, after constructing a fuzzy decision tree, we can generate fuzzy rules from the constructed fuzzy decision tree. Thus, if we have a fuzzy decision tree as shown in Figure 7, we can generate the following fuzzy rules:

$$\begin{array}{l} \vdots \\ \text{IF } \mathbf{A} \text{ is Medium and } \mathbf{B} \text{ is High THEN } \mathbf{C} \text{ is High} \quad (\mathrm{CF} = 0.50) \\ \text{IF } \mathbf{A} \text{ is Medium and } \mathbf{B} \text{ is Medium THEN } \mathbf{C} \text{ is Medium} \quad (\mathrm{CF} = 0.65) \\ \text{IF } \mathbf{A} \text{ is Medium and } \mathbf{B} \text{ is Medium THEN } \mathbf{C} \text{ is Low} \quad (\mathrm{CF} = 0.63) \\ \text{IF } \mathbf{A} \text{ is Medium and } \mathbf{B} \text{ is Low THEN } \mathbf{C} \text{ is Low} \quad (\mathrm{CF} = 0.69), \\ \vdots \end{array} \tag{3}$$

where **A** and **B** are called antecedent attributes, and **C** is called a consequent attribute.

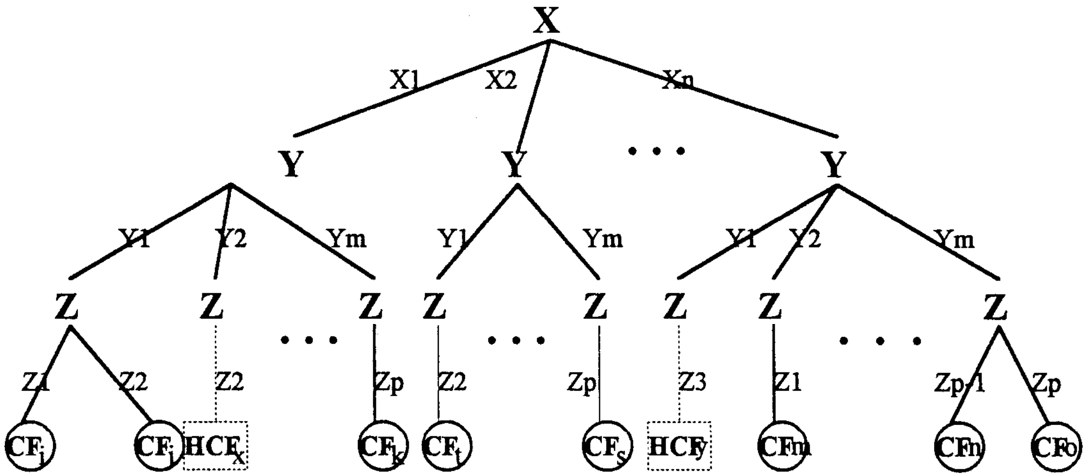


Figure 5. A fuzzy decision tree.

X	Y	. . .	Z
$\{X_1/c_{11}\}$	$\{Y_1/c_{12}\}$	⋮	$\{Z_1/c_{1n}\}$
⋮	⋮	. . .	⋮
$\{X_1/c_{21}\}$	$\{Y_1/c_{22}\}$	. . .	$\{Z_1/c_{2n}\}$
⋮	⋮	. . .	⋮
⋮	⋮	⋮	⋮

Figure 6. A fuzzy relation.

A FUZZY CONCEPT LEARNING SYSTEM ALGORITM

In the following, we present a fuzzy concept learning system (FCLS) algorithm based on Hunt et al. (1966) for constructing a fuzzy decision tree from a relational datasbase system and generate fuzzy rules from the constructed fuzzy decision tree.

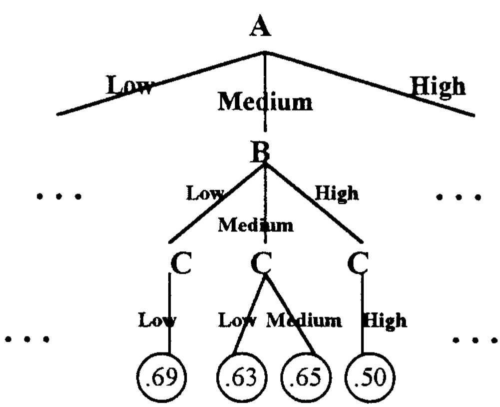


Figure 7. A subtree of a fuzzy decision tree.

Let $\mathbf{S}$ be a set of atrributes (i.e., $\mathbf{S} = \{\mathbf{X}, \mathbf{Y}, \ldots, \mathbf{W}\}$) that determine attribute $\mathbf{Z}$ (i.e., the set $\mathbf{S}$ contains a set of antecedent attributes, and $\mathbf{Z}$ is a consequent attribute). Selecting different elements in the set $\mathbf{S}$ as a root node may result in constructing different kinds of fuzzy decision trees. In the ID3 learning algorithm, Quinlan (1979) makes use of the entropy function of information theory to choose the feature that leads to the greatest reduction in the estimated entropy of information of the training instances. In this paper, we use the concept of fuzziness of attribute (FA) to select an attribute in $\mathbf{S}$ as a decision node that has the smallest FA, such that the number of nodes in the generated fuzzy decision tree can be reduced.

Definition 1: Let $\mathbf{S}$ be a set of attributes that determine attribute $\mathbf{Z}, \mathbf{S} = \{\mathbf{X}, \mathbf{Y}, \ldots, \mathbf{W}\}$, and let $t_j(\mathbf{X})$ denote the value of the attribute $\mathbf{X}$ of the jth training instance (i.e., the jth tuple of a relation) in a relational database; then the fuzziness of the attribute $\mathbf{X}$, denoted by $\mathrm{FA}(\mathbf{X})$, is defined by

$$\mathrm{FA}(\mathbf{X}) = \frac{\sum_{j=1}^{c} \left(1 - \mu_{X_i}\left(t_j(\mathbf{X})\right)\right)}{c} \tag{4}$$

where c is the number of training instances, X_i is any linguistic term of the attribute $\mathbf{X}$, and $\mu_{X_i}(t_j(\mathbf{X}))$ indicates the degree of membership that the value of the attribute $\mathbf{X}$ of the jth training instance belongs to the linguistic term X_i.

Example 1: Assume that we have a relation that has only four tuples as shown in Figure 1, and assume that Figure 4 is the result of the fuzzification of Figure 1, then the fuzziness of the attribute $\mathbf{A}, \mathrm{FA}(\mathbf{A})$, can be evaluated as follows:

$$\begin{aligned} \mathrm{FA}(\mathbf{A}) &= [(1 - 0.5) + (1 - 0.5) + (1 - 0.75) + (1 - 1.0)]/4 \\ &= 0.31 \end{aligned} \tag{5}$$

Definition 2: Every certainty factor node in the path of a fuzzy decision tree is associated with a certainty factor (CF) value. The certainty factor value CF is defined by

$$CF = \min\{Avg(F_1), Avg(F_2), Avg(F_3)\} \tag{6}$$

where $Avg(F_1)$, $Avg(F_2)$, $Avg(F_3)$ are the average values of the linguistic terms F_1, F_2, and F_3, respectively, and F_1, F_2, and F_3 are on a path $\mathbf{D}_1 \xrightarrow{F_1} \mathbf{D}_2 \xrightarrow{F_2} \mathbf{D}_3 \xrightarrow{F_3} \boxed{CF}$ in the tree, and

$$Avg(F_i) = \frac{\sum_{j=1}^{s} \mu_{F_i}\left(t_j(\mathbf{D}_i)\right)}{s} \tag{7}$$

where $t_j(\mathbf{D}_i)$ represents the value of the attribute $\mathbf{D}_i$ of the jth tuple of a relation, s is the the number of training instances (i.e., the number of tuples in the relation) in which the value of the attribute $\mathbf{D}_i$ is the linguistic term F_i, $\mu_{F_i}(t_j(\mathbf{D}_i))$ indicates the degree of membership that the value of the attribute $\mathbf{D}_i$ of the jth tuple of a relation belongs to the linguistic term D_i, and $1 \leq i \leq 3$.

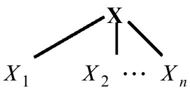
Before we present the FCLS algorithm, we must fuzzify a relation of a relational database system into a fuzzy relation as described previously. Let **S** be a set of antecedent attributes and **Z** be a consequent attribute of a relation in a relational database system. In a relational database system, a tuple in a relation forms a training instance. Let **T** be a set of training instances. The FCLS algorithm is now presented as follows:

FCLS Algorithm

Step 1: Fuzzify the relation into the fuzzy relation.

Step 2: Select an attribute among the set **S** of antecedent attributes that has the smallest FA. Assume attribute **X** with the smallest FA; then partition the set **T** of the traininhg instances into subsets $T_1, T_2, \ldots,$ and T_n according to the fuzzy domain $\{X_1, X_2, \ldots, X_n\}$ of the attribute **X**, respectively. Compute the average value $Avg(X_i)$ of X_i based on formula (7), where $1 \leq i \leq n$.

Step 3: Let the attribute **X** be the decision node, and sprout the tree according to the fuzzy domain of the attribute **X** shown as follows:



where $X_1, X_2, \ldots, X_n$ are linguistic terms represented by fuzzy sets and the set $\{X_1, X_2, \ldots, X_n\}$ is the fuzzy domain of the attribute **X**.

Step 4: Let $\mathbf{S} = \mathbf{S} - \{\mathbf{X}\}$, where $-$ is the difference operator between sets.

Step 5: For $i \leftarrow 1$ to n do

{

Let $\mathbf{T} \leftarrow T_i$;

If $\mathbf{S} = \varnothing$ then

{

create a decision node for consequent attribute **Z**.

partition the training instances **T** into $T_1, T_2, \ldots, T_k$ according to the fuzzy domain $\{Z_1, Z_2, \ldots, Z_k\}$ of the attribute **Z**.

compute the average value $\mathrm{Avg}(Z_i)$ of Z_i, where $1 \le i \le k$;

create a terminal node for every T_i with $\mathrm{Avg}(Z_i) \neq 0$ and compute the value CF_i associated with the created certainty factor node for every nonnull path in the tree.

}

else

go to step 2.

In the following, we use an example to illustrate the fuzzy rule generation process.

Example 2: Assume that in a relational database system we have a relation shown in Table 1.

From Table 1, we can see that the attribute Salary is determined by the attributes Degree and Experience. In this case, $\mathbf{S} = \{\text{Degree, Experience}\}$ and $\mathbf{Z} = \text{Salary}$, where the attributes Degree and Ex-

Table 1. A relation in a relational database system

Emp-ID	Degree	Experience	Salary
S1	Ph.D.	7.2	63,000
S2	Master	2.0	37,000
S3	Bachelor	7.0	40,000
S4	Ph.D.	1.2	47,000
S5	Master	7.5	53,000
S6	Bachelor	1.5	26,000
S7	Bachelor	2.3	29,000
S8	Ph.D.	2.0	50,000
S9	Ph.D.	3.8	54,000
S10	Bachelor	3.5	35,000
S11	Master	3.5	40,000
S12	Master	3.6	41,000
S13	Master	10	68,000
S14	Ph.D.	5.0	57,000
S15	Bachelor	5.0	36,000
S16	Master	6.2	50,000
S17	Bachelor	0.5	23,000
S18	Master	7.2	55,000
S19	Master	6.5	51,000
S20	Ph.D.	7.8	65,000
S21	Master	8.1	64,000
S22	Ph.D.	8.5	70,000

perience are called antecedent attributes and the attribute Salary is called a consequent attribute. From Table 1, we can see that the values of the attribute Degree are Ph.D. (P), Master (M), and Bachelor (B), and the domains of the attributes Experience and Salary are from 0 to 10 and from 20,000 to 70,000, respectively. In this example, we also let the fuzzy domain of the attribute Degree be {Ph.D (P), Master (M), Bachelor (B)} and the fuzzy domain of the attributes Experience and Salary be {high (H), somewhat-high (SH), medium (M), somewhat-low (SL), low (L)}, respectively. The membership function curves of these linguistic terms are shown in Figure 8.

First, we fuzzify the relation shown in Table 1. The result of fuzzification of Table 1 is a fuzzy relation shown in Table 2.

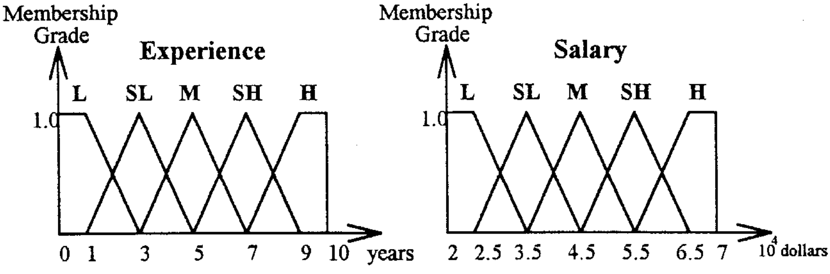


Figure 8. Membership function curves.

Table 2. A fuzzy relation

Emp-ID	Degree	Experience	Salary
S1	{Ph.D./ 1.0}	{SH / 0.9}	{H / 0.8}
S2	{Master/ 1.0}	{SL / 0.5}	{SL / 0.8}
S3	{Bachelor/ 1.0}	{SH / 1.0}	{M / 0.5}
S4	{Ph.D./ 1.0}	{L / 0.9}	{M / 0.8}
S5	{Master/ 1.0}	{SH / 0.75}	{SH / 0.8}
S6	{Bachelor/ 1.0}	{L / 0.75}	{L / 0.9}
S7	{Bachelor/ 1.0}	{SL / 0.65}	{L / 0.6}
S8	{Ph.D./ 1.0}	{L / 0.5}	{SH / 0.5}
S9	{Ph.D./ 1.0}	{SL / 0.6}	{SH / 0.9}
S10	{Bachelor/ 1.0}	{SL / 0.75}	{SL / 1.0}
S11	{Master/ 1.0}	{SL / 0.75}	{SL / 0.5}
S12	{Master/ 1.0}	{SL / 0.7}	{M / 0.6}
S13	{Master/ 1.0}	{H / 1.0}	{H / 1.0}
S14	{Ph.D./ 1.0}	{M / 1.0}	{SH / 0.8}
S15	{Bachelor/ 1.0}	{M / 1.0}	{SL / 0.9}
S16	{Master/ 1.0}	{SH / 0.6}	{M / 0.5}
S17	{Bachelor/ 1.0}	{L / 1.0}	{L / 1.0}
S18	{Master/ 1.0}	{SH / 0.9}	{SH / 1.0}
S19	{Master/ 1.0}	{SH / 0.75}	{SH / 0.6}
S20	{Ph.D./ 1.0}	{SH / 0.6}	{H / 1.0}
S21	{Master/ 1.0}	{H / 0.55}	{H / 0.9}
S22	{Ph.D./ 1.0}	{H / 0.75}	{H / 1.0}

Then, based on formula (4), we can compute the fuzziness of each attribute in the set $\mathbf{S}$, $\mathbf{S} = \{\text{Degree, Experience}\}$, shown as follows:

$$\mathrm{FA}(\text{Degree}) = 0$$

$$\begin{aligned}\mathrm{FA}(\text{Experience}) = \big[&(1-0.9)+(1-0.5)+(1-1.0)\\&+(1-0.9)+(1-0.75)+(1-0.75)+(1-0.65)\\&+(1-0.5)+(1-0.6)+(1-0.75)+(1-0.75)\\&+(1-0.7)+(1-1.0)+(1-1.0)\\&+(1-1.0)+(1-0.6)+(1-1.0)+(1-0.9)\\&+(1-0.75)+(1-0.6)+(1-0.55)\\&+(1-0.75)\big]/22\\=\ &0.23 \qquad (8)\end{aligned}$$

After applying the FCLS algorithm, the fuzzy decision tree is constructed as shown in Figure 9.

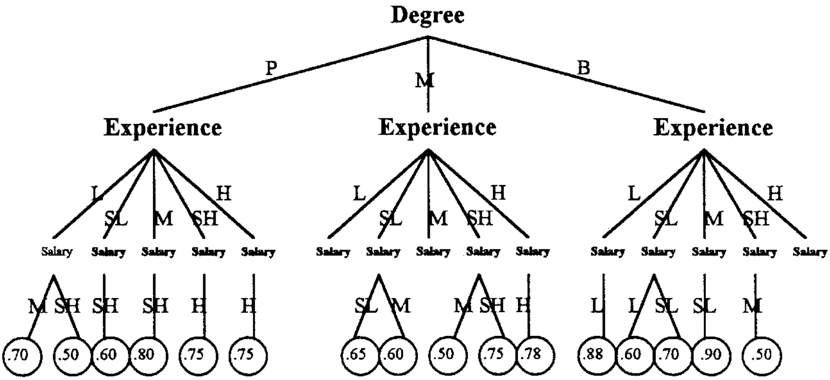


Figure 9. Fuzzy decision tree of Example 2.

Consequently, we can get 16 fuzzy rules from Figure 9 shown as follows:

Rule 1: IF Degree is P AND Experience is L THEN Salary is M (CF = 0.70)

Rule 2: IF Degree is P AND Experience is L THEN Salary is SH (CF = 0.50)

Rule 3: IF Degree is P AND Experience is SL THEN Salary is SH (CF = 0.60)

Rule 4: IF Degree is P AND Experience is M THEN Salary is SH (CF = 0.80)

Rule 5: IF Degree is P AND Experience is SH THEN Salary is H (CF = 0.75)

Rule 6: IF Degree is P AND Experience is H THEN Salary is H (CF = 0.75)

Rule 7: IF Degree is M AND Experience is SL THEN Salary is SL (CF = 0.65)

Rule 8: IF Degree is M AND Experience is SL THEN Salary is M (CF = 0.60)

(9)

Rule 9: IF Degree is M AND Experience is SH THEN Salary is M (CF = 0.50)

Rule 10: IF Degree is M AND Experience is SH THEN Salary is SH (CF = 0.75)

Rule 11: IF Degree is M AND Experience is H THEN Salary is H (CF = 0.78)

Rule 12: IF Degree is B AND Experience is L THEN Salary is L (CF = 0.88)

Rule 13: IF Degree is B AND Experience is SL THEN Salary is L (CF = 0.60)

Rule 14: IF Degree is B AND Experience is SL THEN Salary is SL (CF = 0.70)

Rule 15: IF Degree is B AND Experience is M THEN Salary is SL (CF = 0.90)

Rule 16: IF Degree is B AND Experience is SH THEN Salary is M (CF = 0.50).

A very important property of the proposed FCLS algorithm for constructing a fuzzy decision tree is that it terminates. The proof of this property requires the following lemma (Peterson, 1981):

Lemma 1: In any infinite directed tree in which each node has only a finite number of direct successors, there is an infinite path leading from the root.

Proof: See Peterson (1981, p. 97).

Theorem 1: The fuzzy decision tree constructed by the FCLS algorithm is finite.

Proof: The proof is by contradiction. Assume that there exists an infinite fuzzy decision tree. Because each node in the tree is associated with an attribute of a relation, and because the number of successors for each node X in the tree is limited by the number of linguistic terms (fuzzy regions) in the fuzzy domain of the attribute, by Lemma 1 there is an infinite path from the root node to either the certainty factor node or the hypothetical certainty factor node. But in a fuzzy decision tree, the path length from the root node to either the certainty factor node or the hypothetical certainty factor node is limited by $m - 1$, where m is the number of attributes in a relation of a relational database system. This is a contradiction. Proving that an infinite fuzzy decision tree existed was incorrect.

COMPLETENESS OF FUZZY DECISION TREES

In the following, we present a method for dealing with the completeness of the constructed fuzzy decision tree created by the proposed FCLS algorithm. The main purpose of a fuzzy learning algorithm is to use a set of training instances (i.e., tuples of relations) to learn or construct some fuzzy rules. If the training instances to be learned do not contain all kinds of conditions, null paths will be produced in the generated fuzzy decision tree. Sudkamp and Hammell (1994) proposed the region growing method and the weighted average method to complete the entries of fuzzy associative memory. In the following, we will present a method for completing the null paths in a fuzzy decision tree based on Sudkamp and Hammell (1994). After the fuzzy decision tree has become a completed fuzzy decision tree, the complete fuzzy rule base will be generated from the tree.

Let α be a mapping function from linguistic terms to ordinary numbers and let β be a mapping function from ordinary numbers to linguistic terms. For example, in Figure 8, we let

$$\begin{aligned} \alpha(\mathrm{L}) &= 1, & \beta(1) &= \mathrm{L} \\ \alpha(\mathrm{SL}) &= 2, & \beta(2) &= \mathrm{SL} \\ \alpha(\mathrm{M}) &= 3, & \beta(3) &= \mathrm{M} \\ \alpha(\mathrm{SH}) &= 4, & \beta(4) &= \mathrm{SH} \\ \alpha(\mathrm{H}) &= 5, & \beta(5) &= \mathrm{H} \end{aligned} \tag{10}$$

For every path in the fuzzy decision tree created by the proposed FCLS algorithm, if there are some null paths, then a hypothetical certainty factor node HCF is created for each null path. In this case, the path from the root node to a hypothetical certainty factor node forms a virtual fuzzy rule. Furthermore, in order to minimize the error of the degree of belief of the generated virtual fuzzy rules, we let the value associated with each hypothetical certainty factor node be equal to 0.5. Assume that Figure 10 is a subtree of a fuzzy decision tree, where Y_{i-1}, Y_i, and Y_{i+1} are fuzzy values in the fuzzy domain of **Y**; Y_{i-1} and Y_{i+1} are on the nonnull path with the rightmost value Z_{L} and the leftmost value Z_{R} of **Z**, where Z_{L} and Z_{R} are linguistic terms and where Y_i is on the null path of the tree. Furthermore, assume that we want to sprout the branch Z_{U} denoted by the dotted line shown in Figure 10, where Z_{U} is a linguistic term. Then,

Case 1: If we want to sprout the branch Z_{U} denoted by the dotted line shown in Figure 11, then Z_{U} can be evaluated as follows:

$$Z_{\mathrm{U}} = \begin{cases} \beta\left(\dfrac{\alpha(Z_{\mathrm{L}}) + \alpha(Z_{\mathrm{R}}) - 1}{2}\right), & \text{if } |\alpha(Z_{\mathrm{L}}) - \alpha(Z_{\mathrm{R}})| \\ & \quad \text{is an odd number and } \mathrm{CF_L} \geq \mathrm{CF_R} \\ \beta\left(\dfrac{\alpha(Z_{\mathrm{L}}) + \alpha(Z_{\mathrm{R}}) + 1}{2}\right), & \text{if } |\alpha(Z_{\mathrm{L}}) - \alpha(Z_{\mathrm{R}})| \\ & \quad \text{is an odd number and } \mathrm{CF_L} < \mathrm{CF_R} \\ \beta\left(\dfrac{\alpha(Z_{\mathrm{L}}) + \alpha(Z_{\mathrm{R}})}{2}\right), & \text{otherwise} \end{cases} \tag{11}$$

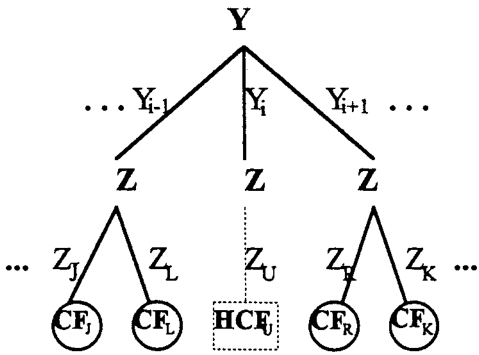


Figure 10. A subtree of a fuzzy decision tree.

and we let the associated value of the hypothetical certainty node HCF_U be equal to 0.5.

Case 2: If we want to sprout the branch Z_U denoted by the dotted line shown in Figure 12, where the node **Z** of Y_1 to Y_{i-1} cannot sprout out any branches, and assume that we have the following metaknowledge:

The smaller the values of the attribute **Y**, the smaller the value of the attribute **Z**,

then Z_U can be evaluated as follows:

$$Z_U = \beta\left(\left\lfloor \frac{1 + \alpha(Z_R)}{2} \right\rfloor\right) \tag{12}$$

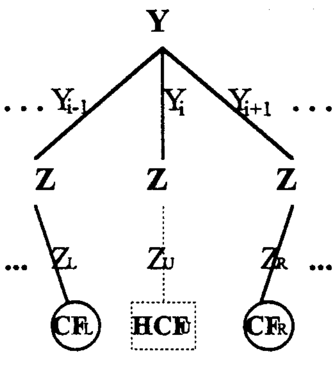


Figure 11. A subtree of a fuzzy decision tree (case 1).

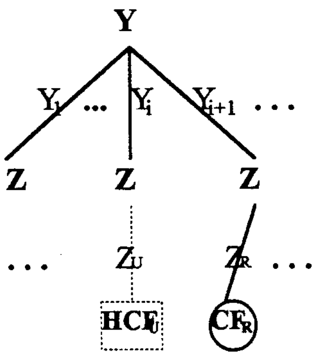


Figure 12. A subtree of a fuzzy decision tree (case 2).

Assume that we have the following metaknowledge:

The smaller the values of the attribute **Y**, the greater the values of the attribute **Z**,

then Z_U can be evaluated as follows:

$$Z_U = \beta\left(\left\lfloor \frac{\alpha(Z_K) + \alpha(Z_R)}{2} \right\rfloor\right) \tag{13}$$

where $\alpha(Z_K)$ has the largest ordinary number in the fuzzy domain of **Z**, and we let the value of hypothetical certainty node HCF_U be equal to 0.5.

Case 3: If we want to sprout the branch Z_U denoted by the dotted line shown in Figure 13, where the decision node **Z** of Y_{i+1} to Y_m cannot sprout out any branches, and assume that we have the following metaknowledge:

The greater the values of the attribute **Y**, the smaller the values of the attribute **Z**,

then Z_U can be evaluated as follows:

$$Z_U = \beta\left(\left\lfloor \frac{1 + \alpha(Z_L)}{2} \right\rfloor\right) \tag{14}$$

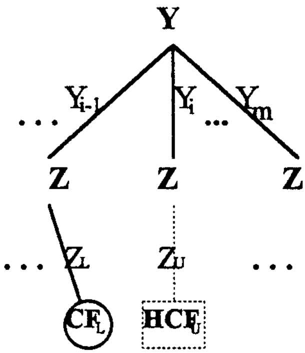


Figure 13. A subtree of a fuzzy decision tree (case 3).

Assume that we have the following metaknowledge:

The greater the values of the attribute **Y**, the greater the values of the attribute **Z**,

then Z_U can be evaluated as follows:

$$Z_U = \beta\left(\left\lfloor \frac{\alpha(Z_L) + \alpha(Z_K)}{2} \right\rfloor\right) \tag{15}$$

where $\alpha(Z_K)$ has the largest ordinary number in the fuzzy domain of **z**, and we let the value of hypothetical certainty node HCF_U be equal to 0.5.

This procedure will go on continuously until there is no null path in the fuzzy decision tree.

Example 3: The assumptions are the same as in Example 2, where Figure 9 is the constructed fuzzy decision tree of Example 2. By performing the proposed method, the complete fuzzy decision tree is constructed as shown in Figure 14. Figure 14 shows a complete fuzzy decision tree derived from Figure 9 after performing the proposed method.

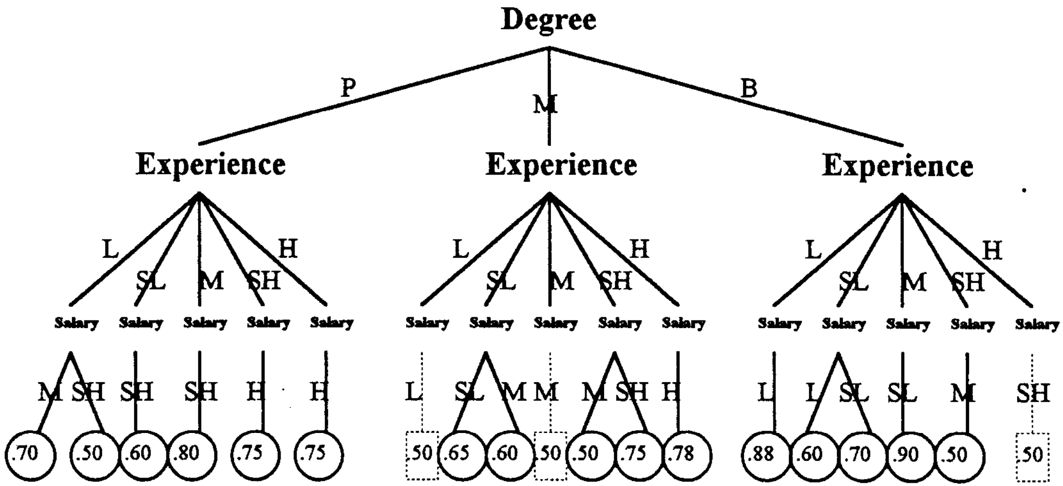


Figure 14. Complete fuzzy decision tree of Example 3.

For the path Degree $\xrightarrow{M}$ Experience $\xrightarrow{M}$ Salary shown in Figure 14, based on case 1 of the proposed method, we can see that

$$Y_i = \mathrm{M}, \quad Y_{i-1} = \mathrm{SL}, \quad Y_{i+1} = \mathrm{SH}, \quad Z_\mathrm{L} = M, \quad Z_\mathrm{R} = \mathrm{M}.$$

Because $|\alpha(Z_\mathrm{L}) - \alpha(Z_\mathrm{R})| = 0$, we can see that

$$\begin{aligned} Z_\mathrm{U} &= \beta\left(\frac{\alpha(Z_\mathrm{L}) + \alpha(Z_\mathrm{R})}{2}\right) \\ &= \beta\left(\frac{\alpha(\mathrm{M}) + \alpha(\mathrm{M})}{2}\right) \\ &= \beta\left(\frac{3+3}{2}\right) \\ &= \beta(3) \\ &= \mathrm{M} \end{aligned}$$

That is, we can get a virtual fuzzy rule shown as follows:

$$\text{Degree} \xrightarrow{M} \text{Experience} \xrightarrow{M} \text{Salary} \xrightarrow{M} \boxed{0.50} \qquad (16)$$

From Figure 14, we can see that there are three null paths in the tree, that is,

$$\begin{aligned} &\text{Degree} \xrightarrow{M} \text{Experience} \xrightarrow{L} \text{Salary} \xrightarrow{L} \boxed{0.50} \\ &\text{Degree} \xrightarrow{M} \text{Experience} \xrightarrow{M} \text{Salary} \xrightarrow{M} \boxed{0.50} \qquad (17) \\ &\text{Degree} \xrightarrow{B} \text{Experience} \xrightarrow{H} \text{Salary} \xrightarrow{SH} \boxed{0.50} \end{aligned}$$

Thus, we can get three virtual fuzzy rules shown as follows:

Rule 17: IF Degree is M AND Experience is L
THEN Salary is L (CF = 0.50)

Rule 18: IF Degree is M AND Experience is M
THEN Salary is M (CF = 0.50) (18)

Rule 19: IF Degree is B AND Experience is H
THEN Salary is SH (CF = 0.50)

ESTIMATION OF NULL VALUES IN RELATIONAL DATABASE SYSTEMS

In the following, we introduce the defuzzification technique of fuzzy numbers. In Chen (1994), we have presented a defuzzification technique of trapezoidal fuzzy numbers based on Kandel (1986) as shown in Figure 15, where the defuzzification value $\mathrm{DEF}(Z_k)$ of the fuzzy number Z_k is e and

$$e = (a + b + c + d)/4 \tag{19}$$

A triangular fuzzy number can be thought as a special case of a trapezoidal fuzzy number. Thus, the defuzzification value $\mathrm{DEF}(Z_k)$ of

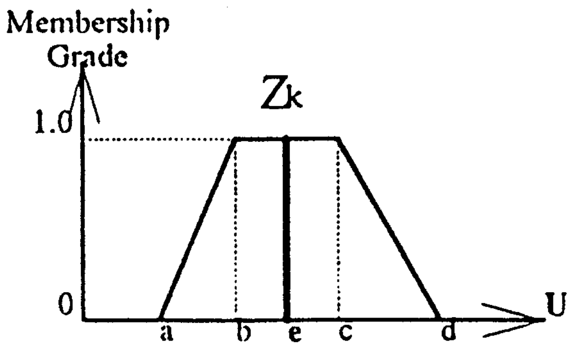


Figure 15. Defuzzification of a trapezoidal fuzzy number.

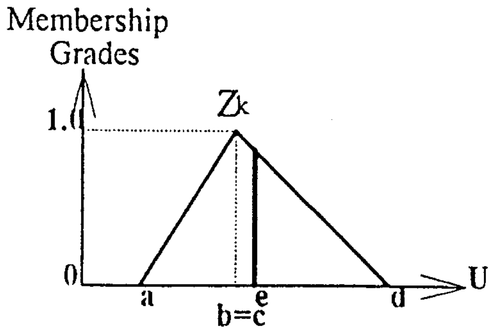


Figure 16. Defuzzification of a triangular fuzzy number.

the triangular fuzzy number Z_k shown in Figure 16 is e, where

$$e = (a + b + b + d)/4 \tag{20}$$

If v is a crisp value of a fuzzifable attribute **V** in some tuples of a relational database, then we let $\{V_s/\mu_{V_S}(v)), (V_t/\mu_{V_t}(v))\}$ be fuzzified values of v, where V_S and V_t are linguistic terms represented by fuzzy sets, $\mu_{V_S}(v) \geq \mu_{V_t}(v)$, and $\mu_{V_S}(v) + \mu_{V_t}(v) = 1.0$ If v is a crisp value of a unfuzzifiable attribute **V** in some tuples of a relational database, then we let $\{(v/1.0)\}$ be the fuzzified value of v. In order to estimate null values in a relational database system, we must first modify the fuzzified value of v into the form $F_v = [(V_m/\mu_{V_m}(v)), (V_n/\mu_{V_n}(v))]$, where V_m and V_n are linguistic terms represented by fuzzy sets.

Case 1: If **V** is a fuzzifiable attribute (linguistic variable) and $\mu_{V_S}(v) \neq 1.0$, then we let

$$V_m = V_S, \quad V_n = V_t, \quad \mu_{V_m}(v) = \mu_{V_S}(v), \quad \mu_{V_n}(v) = \mu_{V_t}(v)$$

Case 2: If **V** is a fuzzifiable attribute (linguistic variable) and $\mu_{V_S}(v) = 1.0$, then we let

$$V_m = V_n = V_S \quad \text{and} \quad \mu_{V_m}(v) = \mu_{V_n}(v) = \mu_{V_S}(v) = 1.0$$

Case 3: If **V** is an unfuzzifiable attribute, then we let

$$V_m = V_n = v \qquad \text{and} \qquad \mu_{V_m}(v) = \mu_{V_n}(v) = 1.0$$

Example 4: Let us consider the membership function curves shown in Figure 8.

1. If **V** = Experience and $v = 1.0$, then, based on Figure 8, the fuzzified value of v is {(L/1.0), (SL/0)} and $F_v = [(\text{L}/1.0), (\text{L}/1.0)]$.
2. If **V** = Experience and $v = 5.5$, then based on Figure 8, the fuzzified value of v is {(M/0.75), (SH/0.25)} and $F_v = [(\text{M}/0.75), (\text{SH}/0.25)]$.
3. If **V** = Experience and $v = 3$, then based on Figure 8, the fuzzified value of v is {(SL/1.0), (M/0)} and $F_v = [(\text{SL}/1.0), (\text{SL}/1.0)]$.
4. If **V** = Degree and v = Master, then the fuzzified value of v is {(Master/1.0)} and $F_v = [(\text{Master}/1.0), (\text{Master}/1.0)]$.

In the following, we present a method for estimating null values in relational database systems.

Assume that x and y are crisp domain values of attributes **X** and **Y** in some tuples of a relational database, respectively, and assume that z is a null value of attribute **Z**. Let $F_x = [(X_a/\mu_{X_a}(x)), (X_b/\mu_{X_b}(x))]$ and $F_y = [Y_c/\mu_{Y_c}(y)), (Y_d/\mu_{Y_d}(x))]$ be the modified forms of the fuzzified values of x and y, respectively. Assume that the fuzzy rule base contains the following fuzzy rules generated by the proposed FCLS algorithm:

$$\begin{aligned}
&\text{IF } \mathbf{X} \text{ is } X_a \text{ and } \mathbf{Y} \text{ is } Y_c \text{ THEN } \mathbf{Z} \text{ is } Z_{M_1} (\text{CF} = C_1)\\
&\text{IF } \mathbf{X} \text{ is } X_a \text{ and } \mathbf{Y} \text{ is } Y_c \text{ THEN } \mathbf{Z} \text{ is } Z_{M_2} (\text{CF} = C_2)\\
&\text{IF } \mathbf{X} \text{ is } X_a \text{ and } \mathbf{Y} \text{ is } Y_c \text{ THEN } \mathbf{Z} \text{ is } Z_{M_3} (\text{CF} = C_3) \qquad (21)\\
&\text{IF } \mathbf{X} \text{ is } X_b \text{ and } \mathbf{Y} \text{ is } Y_d \text{ THEN } \mathbf{Z} \text{ is } Z_{N_1} (\text{CF} = D_1)\\
&\text{IF } \mathbf{X} \text{ is } X_b \text{ and } \mathbf{Y} \text{ is } Y_d \text{ THEN } \mathbf{Z} \text{ is } Z_{N_2} (\text{CF} = D_2)
\end{aligned}$$

where **X** and **Y** are antecedent attributes; **Z** is a consequent attribute; X_a, X_b, Y_c, Y_d, Z_{M_1}, Z_{M_2}, Z_{M_3}, Z_{N_1}, and Z_{N_2} are linguistic terms represented by fuzzy sets. Then, the null value z can be evaluated as follows:

$$z = \frac{\mu_{X_s}(x) \times \mu_{Y_c}(y) \times \frac{\Sigma_{i-1}^{3} C_i \times \mathrm{DEF}(Z_{M_i})}{\Sigma_{i=1}^{3} C_i} + \mu_{X_b}(x) \times \mu_{Y_d}(y) \times \frac{\Sigma_{i=1}^{2} D_i \times \mathrm{DEF}(Z_{N_i})}{\Sigma_{i=1}^{2} D_i}}{\mu_{X_a}(x) \times \mu_{Y_c}(y) + \mu_{X_b}(x) \times \mu_{Y_d}(y)} \tag{22}$$

where $\mathrm{DEF}(Z_{M_i})$ and $\mathrm{DEF}(Z_{N_i})$ are defuzzified values of the fuzzy sets Z_{M_i} and Z_{N_i}, respectively.

Example 5: Assume that a relational database system contains a relation shown in Table 3, and assume that we want to estimate the null value of the attribute Salary shown in Table 3.

From Table 3, we can see that the tuple with Emp-ID = S23 has a null value in the attribute Salary. Based on the membership functions shown in Figure 8 and after performing the fuzzification process, Table 3 becomes Table 4.

Hence, we can see that $F_x = [(\text{Master}/1.0), (\text{Master}/1.0)]$ and $F_y = [(\text{M}/0.75), (\text{SL}/0.25)]$. Then, after executing the proposed FCLS algorithm and according to the generated fuzzy rules 7, 8, and 18 shown in Examples 2 and 3, the null value of the attribute Salary can be estimated, where rules 7, 8, and 18 are shown as follows:

Rule 7: IF Degree is Master AND Experience is SL

THEN Salary is SL (CF = 0.65)

Rule 8: IF Degree is Master AND Experience is SL (23)

THEN Salary is M (CF = 0.60)

Rule 18: IF Degree is Master AND Experience is M

THEN Salary is M (CF = 0.50)

Table 3. A relation contains null values

Emp-ID	Degree	Experience	Salary
S1	Ph.D.	7.2	63,000
S2	Master	2.0	37,000
S3	Bachelor	7.0	40,000
S4	Ph.D.	1.2	47,000
S5	Master	7.5	53,000
S6	Bachelor	1.5	26,000
S7	Bachelor	2.3	29,000
S8	Ph.D.	2.0	50,000
S9	Ph.D.	3.8	54,000
S10	Bachelor	3.5	35,000
S11	Master	3.5	40,000
S12	Master	3.6	41,000
S13	Master	10	68,000
S14	Ph.D.	5.0	57,000
S15	Bachelor	5.0	36,000
S16	Master	6.2	50,000
S17	Bachelor	0.5	23,000
S18	Master	7.2	55,000
S19	Master	6.5	51,000
S20	Ph.D.	7.8	65,000
S21	Master	8.1	64,000
S22	Ph.D.	8.5	70,000
S23	Master	4.5	Null

Based on formula (22), the null value of the attribute Salary of the employee whose EMP-ID = S23 shown in Table 3 can be evaluated as follows:

$$\frac{1 \times 0.75 \times \dfrac{0.5 \times \mathrm{DEF}(M)}{0.5} + 1 \times 0.25 \times \dfrac{0.6 \times \mathrm{DEF}(M) + 0.65 \times \mathrm{DEF}(SL)}{0.6 + 0.65}}{1 \times 0.75 + 1 \times 0.25}$$

$$= \frac{1 \times 0.75 \times 45{,}000 + 1 \times 0.25 \times \dfrac{0.6 \times 45{,}000 + 0.65 \times 35{,}000}{1.25}}{1 \times 0.75 + 1 \times 0.25}$$

$$= 43{,}700 \tag{24}$$

That is, the salary of the employee whose Emp-ID = S23 is about 43,700.

Table 4. A fuzzy relation contains null values

Emp-ID	Degree	Experience	Salary
S1	{Ph.D./1.0}	{SH/0.9}	{H/0.8}
S2	{Master/1.0}	{SL/0.5}	{SL/0.8}
S3	{Bachelor/1.0}	{SH/1.0}	{M/0.5}
S4	{Ph.D./1.0}	{L/0.9}	{M/0.8}
S5	{Master/1.0}	{SH/0.75}	{SH/0.8}
S6	{Bachelor/1.0}	{L/0.75}	{L/0.9}
S7	{Bachelor/1.0}	{SL/0.65}	{L/0.6}
S8	{Ph.D./1.0}	{L/0.5}	{SH/0.5}
S9	{Ph.D./1.0}	{SL/0.6}	{SH/0.9}
S10	{Bachelor/1.0}	{SL/0.75}	{SL/1.0}
S11	{Master/1.0}	{SL/0.75}	{SL/0.5}
S12	{Master/1.0}	{SL/0.7}	{M/0.6}
S13	{Master/1.0}	{H/1.0}	{H/1.0}
S14	{Ph.D./1.0}	{M/1.0}	{SH/0.8}
S15	{Bachelor/1.0}	{M/1.0}	{SL/0.9}
S16	{Master/1.0}	{SH/0.6}	{M/0.5}
S17	{Bachelor/1.0}	{L/1.0}	{L/1.0}
S18	{Master/1.0}	{SH/0.9}	{SH/1.0}
S19	{Master/1.0}	{SH/0.75}	{SH/0.6}
S20	{Ph.D./1.0}	{SH/0.6}	{H/1.0}
S21	{Master/1.0}	{H/0.55}	{H/0.9}
S22	{Ph.D./1.0}	{H/0.75}	{H/1.0}
S23	{Master/1.0}	{M/0.75, SL/0.25}	Null

CONCLUSIONS

In this paper, we have presented a new algorithm for constructing fuzzy decision trees from relational database systems and generating fuzzy rules from the constructed fuzzy decision trees. We also have presented a method for dealing with the completeness of the constructed fuzzy decision trees. Based on the generated fuzzy rules, we also present a method for estimating null values in relational database systems. The proposed method provides a useful way to estimate null values in relational database systems.

REFERENCES

Chen, S. M. 1994. Using fuzzy reasoning techniques for fault diagnosis of the J-85 jet engines. *Proceeding Third National Conference on Science and*

Technology of National Defense, Taoyuan, Taiwan, Republic of China, pp. 29–39.

Hart, A. 1985. The role of induction in knowledge elicitation. *Expert Syst.* 2:24–28.

Hunt, E. B., J. Martin, and P. J. Stone. 1966. *Experience in induction*. New York: Academic Press.

Jeng, B., and T. P. Liang. 1993. Fuzzy indexing and retrieval in case-based systems. *Proceedings 1993 Pan Pacific Conference on Information Systems*, Taiwan, Republic of China, pp. 258–266.

Kandel, A. 1986. *Fuzzy mathematical techniques with applications*. Reading, MA: Addison-Wesley.

Peterson, J. L. 1981. *Petri nets, theory and the modeling of systems*. Englewood Cliffs, NJ: Prentice-Hall.

Quinlan, J. R. 1979. Discovering rules by induction from large collection of examples. In *Expert systems in the micro electronic age*, ed. D. Michie. Edinburgh: Edinburgh University Press, pp. 168–201.

Quinlan, J. R. 1987. Decision trees as probabilistic classifers. *Proceedings 4th International Workshop on Machine Learning.* Los Altos, CA: Morgan Kauffman, pp. 33–37.

Safavian, S. R., and D. Landgrebe. 1991. A survey of decision tree classifier methodology. *IEEE Trans. Syst. Man Cybernet.* 21:660–674.

Sudkamp, T., and R. J. Hammell II. 1994. Interpolation, completion, and learning fuzzy rules. *IEEE Trans. Syst. Man Cybernet.* 24:332–342.

Wang, L. X., and J. M. Mendel. 1992. Generating fuzzy rules by learning from examples. *IEEE Trans. Syst. Man Cybernet.* 22:1414–1427.

Yeh, M. S., and S. M. Chen. 1995. Generating fuzzy rules from relational database systems. *Proceedings 6th International Conference on Information Management*, Taipei, Taiwan, Republic of China, 219–226.

Yuan, Y., and M. J. Shaw. 1995. Induction of fuzzy decision trees. *Fuzzy Sets Syst.* 69:125–139.

Zadeh, L. A. 1965. Fuzzy sets. *Inform. Control* 8:338–353.

Zadeh, L. A. 1975. The concepts of a linguistic variable and its application to approximate reasoning (I). *Inform. Sci.* 8:199–249.