Generating Learning Sequences for Decision Makers Through Data Mining and Competence Set Expansion

Yi-Chung Hu, Ruey-Shun Chen, and Gwo-Hshiung Tzeng

***Abstract*—For each decision problem, there is a competence set, proposed by Yu, consisting of ideas, knowledge, information, and skills required for solving the problem. Thus, it is reasonable that we view a set of useful patterns discovered from a relational database by data mining techniques as a needed competence set for solving one problem. Significantly, when decision makers have not acquired the competence set, they may lack confidence in making decisions. In order to effectively acquire a needed competence set to cope with the corresponding problem, it is necessary to find appropriate learning sequences of acquiring those useful patterns, the so-called competence set expansion. This paper thus proposes an effective method consisting of two phases to generate learning sequences. The first phase finds a competence set consisting of useful patterns by using a proposed data mining technique. The other phase expands that competence set with minimum learning cost by the minimum spanning table method proposed by Feng and Yu. From a numerical example, we can see that it is possible to help decision makers to solve the decision problems by use of the data mining technique and the competence set expansion, enabling them to make better decisions.**

***Index Terms*—Competence set, data mining, decision making, fuzzy sets.**

NOMENCLATURE

K	Number of linguistic values of a linguistic variable.
k	Dimension of a fuzzy grid.
d	Number of attributes of a database relation, where $d \geq 1$.
n	Total number of tuples of a relational database.
$A^{x_m}_{K,i_m}$	i_m-th linguistic value of K linguistic values which are defined for the linguistic variable x_m, where $1 \leq i_m \leq K$.
$\mu^{x_m}_{K,i_m}$	Membership function of $A^{x_m}_{K,i_m}$.
t_p	p-th tuple of a specified relation, where $t_p = (t_{p_1}, t_{p_2}, \ldots, t_{p_d})$, where $1 \leq p \leq n$.

I. INTRODUCTION

Data mining is the exploration and analysis of large quantities of data in order to discover meaningful patterns [1]. It extracts implicit, previously unknown, and potentially useful patterns from data [22]. Significantly, it is a methodology for the extraction of knowledge from data, knowledge relating to a problem that we wish to solve [2]. Some patterns, such as "purchase amount of product A is large" or "age of customers is close to 30" may be discovered from a relational database set up in one supermarket by data mining techniques, and these patterns could be useful for decision makers.

Decision makers can try to "learn" those useful patterns, that is, they can investigate the corresponding factors or current strategies that can result in those mining results. They can thus acquire or learn the corresponding knowledge from those useful patterns. Finally, decision makers can be confident of solving some decision problems, e.g., proposing a more competitive marketing strategy. To effectively learn the corresponding patterns, it is necessary to generate appropriate learning sequences of those patterns. For example, since learning directly from one pattern to another pattern requires learning cost, which can be measured by time or money, it may be more effective for decision makers to learn "purchase amount of product B is small" before learning "purchase amount of product A is large." A similar example would be that, to obtain knowledge of mathematics, it would be appropriate to learn calculus before learning statistics.

Competence sets were initiated by Yu [24]. Its mathematical foundation was provided by Yu and Zhang [3]. For each decision problem (e.g., promoting products or improving services for a business), there exists a competence set consisting of ideas, knowledge, information, and skills for solving the problem [24]. From this viewpoint, it is reasonable that we view a set of useful patterns discovered from a relational database by data mining techniques as a needed competence set for solving one problem. When decision makers have already acquired the needed competence set and are proficient at it, they will be comfortable and confident in making decisions [4], [23]. Otherwise, they must try to acquire the needed competence set to solve the problem. In order to acquire a needed competence set to cope with the facing decision problem, finding appropriate learning sequences for acquiring those useful patterns, so-called competence set expansion, are very necessary.

This paper thus proposes an effective method consisting of two phases to generate learning sequences. The first phase finds a needed competence set consisting of useful patterns by an proposed algorithm for finding frequent and necessary fuzzy grids, which is a significant part of the fuzzy grids based rules mining algorithm (FGBRMA) proposed by Hu *et al.* [5]. The other phase expands that needed competence set with minimum learning cost by the minimum spanning table method proposed by Feng and Yu [6]. From a numerical example, we can see that it is possible to help decision makers to confidently solve the decision problems they face through the data mining technique and the competence set expansion.

In the following sections, since we incorporate the concepts of fuzzy partition into the proposed data mining technique, we thus introduce the cases for partitioning quantitative and categorical attributes by various linguistic values in Section II. In Section III, we introduce the determination of useful patterns, the data structure for implementing the proposed algorithm, and a heuristic method for determining necessary patterns. Then, we briefly describe the proposed algorithm. The concepts of the competence set expansion are demonstrated in Section IV, and we also briefly introduce the minimum spanning table method in this section. A detailed simulation of a numerical example is described in Section V. Discussions and conclusions are presented in Sections VI and VII, respectively.

II. PARTITION ATTRIBUTES

Fuzzy sets were originally proposed by Zadeh [7], who also proposed the concept of linguistic variables and its applications to approximate reasoning [8]. Formally, a linguistic variable is characterized by a quintuple [9], [21] denoted by $(x, T(x), U, G, M)$, where

x	name of the variable;
$T(x)$	term set of x, that is, the set of names of linguistic values or terms, which are linguistic words or sentences in a natural language [10], of x;
U	universe of discourse;
G	syntactic rule for generating values of x;
M	semantic rule for associating a linguistic value with a meaning.

For example, we can view "Age" as a linguistic variable, $T(\text{Age}) = \{\text{young, close to 30, close to 50, old}\}$, G is a rule which generates the linguistic values in $T(\text{Age})$, and $U = [0, 60]$. $M(\text{young})$ assigns a membership function to young.

A relational database is a collection of tables, each of which is assigned a unique name and consists of a set of attributes and stores a

Manuscript received September 3, 2001; revised January 8, 2002. This paper was recommended by Associate Editor W. Pedrycz.

Y.-C. Hu and R.-S. Chen are with the Institute of Information Management, National Chiao Tung University, Hsinchu 300, Taiwan, R.O.C.

G.-H. Tzeng is with the Institute of Management of Technology, National Chiao Tung University, Hsinchu 300, Taiwan, R.O.C.

Publisher Item Identifier S 1083-4419(02)05723-0.

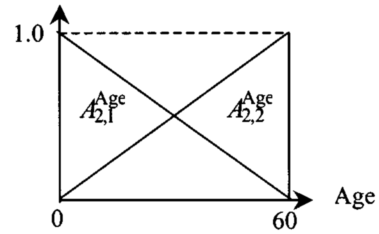

Fig. 1. $K = 2$ for "Age."

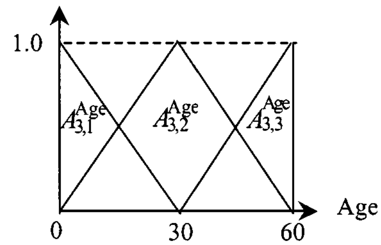

Fig. 2. $K = 3$ for "Age."

large set of records or tuples [11]. It seems to be reasonable that we view each attribute as a linguistic variable. It is noted that, if there exist d attributes in a database, then a d-dimensional feature space is constructed and each attribute is viewed as an axis of this space. The cases for partitioning quantitative and categorical attributes by various linguistic values (i.e., fuzzy partition) are introduced as follows.

A. Partition Quantitative Attributes

Quantitative attributes of a relational database are numeric and have an implicit ordering among values (e.g., age, salary) [11]. Moreover, a quantitative attribute can be partitioned by linguistic values and this technique has been widely used in pattern recognition and fuzzy inference. For example, there are the applications to pattern classification by Ishibuchi *et al.* [12], [13], and Hu *et al.* [14], and to fuzzy rule generation by Wang and Mendel [15]. In addition, some methods for partitioning feature space were discussed by Sun [16] and Bezdek [17].

As we have mentioned earlier, a quantitative attribute can be partitioned by K linguistic values $(K = 2, 3, 4 \ldots)$. It should be noted that the value of K is dependent on the actual requirement or preference of decision makers. For example, $K = 2, K = 3,$ and $K = 4$ for the linguistic variable "Age" are depicted in Figs. 1, 2, and 3, respectively. Triangular membership functions are used for each linguistic value defined in each quantitative attribute for simplicity. A linguistic value denoted by A_{K,i_m}^{Age} can be described in a linguistic sentence. For example

$$A_{K,1}^{\mathrm{Age}}\text{: young, and below } 60/(K-1) \tag{1}$$

$$A_{K,K}^{\mathrm{Age}}\text{: ;old, and above } [60 - 60/(K-1)] \tag{2}$$

$$\begin{aligned} A_{K,i_m}^{\mathrm{Age}}\text{: }&\text{close to } (i_m - 1) \times [60 - 60/(K-1)] \\ &\text{and between } (i_m - 2) \times [60 - 60/(K-1)] \\ &\text{and } i_m \times [60 - 60/(K-1)], \quad \text{for } 1 < i_m < K. \end{aligned} \tag{3}$$

In addition, the membership function of A_{K,i_m}^{Age}, which is denoted by $\mu_{K,i_m}^{\mathrm{Age}}$, is stated as follows:

$$\mu_{K,i_m}^{\mathrm{Age}}(x) = \max\left\{1 - \left|x - a_{i_m}^K\right| \Big/ b^K, 0\right\} \tag{4}$$

where

$$a_{i_m}^K = \mathrm{mi} + (\mathrm{ma} - \mathrm{mi}) \cdot (i_m - 1)/(K-1) \tag{5}$$

$$b^K = (\mathrm{ma} - \mathrm{mi})/(K-1) \tag{6}$$

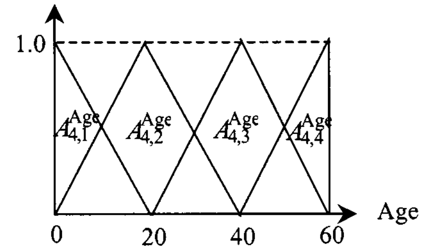

Fig. 3. $K = 4$ for "Age."

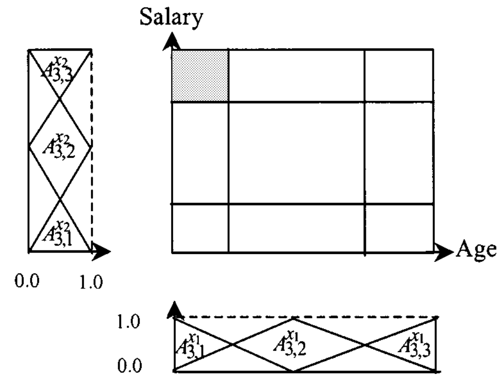

Fig. 4. Both "Age" and "Salary" are divided into three fuzzy partitions.

where ma and mi are the maximal value and the minimal value of the domain of Age (i.e., $U = [0, 60]$), respectively. In this example, we can see that ma and mi are equal to 60 and 0, respectively. Ishibuchi *et al.* [12] referred to such a partition method as the simple fuzzy grid method. Furthermore, a high-dimensional fuzzy grid can be generated. For example, if both attributes, e.g., "Age" (x_1) and "Salary" (x_2), are all partitioned by three linguistic values, then a feature space is partitioned into 3 × 3 two-dimensional (2-D) fuzzy grids (i.e., a grid-type fuzzy partition), as shown in Fig. 4. For the shaded 2-D fuzzy grid shown in Fig. 4, whose linguistic value is "young AND high," we use $A_{3,1}^{x_1} \wedge A_{3,3}^{x_2}$ to denote it.

B. Partition Categorical Attributes

Categorical attributes of a relational database have a finite number of possible values, with no ordering among values (e.g., sex, color) [11]. If the distinct attribute values are n' (n' is finite), then this attribute can only be partitioned by n' linguistic values. For example, since the attribute "Sex" is categorical, the linguistic sentence of each linguistic value can thus be stated as follows:

$$A_{2,1}^{\mathrm{Sex}}\text{: male} \tag{7}$$

$$A_{2,2}^{\mathrm{Sex}}\text{: female.} \tag{8}$$

A linguistic value $A_{n,i_m}^{x_m}$ is defined in $(i_m - \varepsilon, i_m + \varepsilon)(\varepsilon \to 0)$, where the membership function of $A_{n',i_m}^{x_m}$ is "1" for $(i_m - \varepsilon, i_m + \varepsilon)$.

III. Algorithm for Finding Frequent and Necessary Fuzzy Grids

In the proposed algorithm for finding frequent and necessary fuzzy grids, which is a significant part of the FGBRMA, $A_{K,i_m}^{x_m}$ is viewed as a candidate one–dimensional (1-D) fuzzy grid. After all candidate 1-D fuzzy grids have been generated, we need to determine how to generate frequent fuzzy grids by those candidate 1-D fuzzy grids. That is, each frequent fuzzy grid denoted by its linguistic value is a useful pattern for decision making. In fact, the terms "candidate" and "frequent" originated from the well-known *apriori* algorithm [18], which

TABLE I
INITIAL TABULAR FGTTFS

Fuzzy grids	FG						TT		FS
	$A^{x_1}_{3,1}$	$A^{x_1}_{3,2}$	$A^{x_1}_{3,3}$	$A^{x_2}_{3,1}$	$A^{x_2}_{3,2}$	$A^{x_2}_{3,3}$	t_1	t_2	
$A^{x_1}_{3,1}$	1	0	0	0	0	0	$\mu^{x_1}_{3,1}(t_{1_1})$	$\mu^{x_1}_{3,1}(t_{2_1})$	FS($A^{x_1}_{3,1}$)
$A^{x_1}_{3,2}$	0	1	0	0	0	0	$\mu^{x_1}_{3,2}(t_{1_1})$	$\mu^{x_1}_{3,2}(t_{2_1})$	FS($A^{x_1}_{3,2}$)
$A^{x_1}_{3,3}$	0	0	1	0	0	0	$\mu^{x_1}_{3,3}(t_{1_1})$	$\mu^{x_1}_{3,3}(t_{2_1})$	FS($A^{x_1}_{3,3}$)
$A^{x_2}_{3,1}$	0	0	0	1	0	0	$\mu^{x_2}_{3,1}(t_{1_2})$	$\mu^{x_2}_{3,1}(t_{2_2})$	FS($A^{x_2}_{3,1}$)
$A^{x_2}_{3,2}$	0	0	0	0	1	0	$\mu^{x_2}_{3,2}(t_{1_2})$	$\mu^{x_2}_{3,2}(t_{2_2})$	FS($A^{x_2}_{3,2}$)
$A^{x_2}_{3,3}$	0	0	0	0	0	1	$\mu^{x_2}_{3,3}(t_{1_2})$	$\mu^{x_2}_{3,3}(t_{2_2})$	FS($A^{x_2}_{3,3}$)

is an influential algorithm for mining association rules [11]. Generally, a frequent item set means that this set is useful for decision makers.

Example: Any set of fruits, e.g., {Apple, Orange}, sold in one supermarket is a candidate item set. If the purchase frequency (i.e., divide the number of transactions that buy apples and oranges by the total number of transactions) of {Apple, Orange} is larger or equal to a prespecified threshold named support, then {Apple, Orange} is a frequent item set.

In the following sections, we introduce the determination of frequent fuzzy grids, the data structure for implementing the proposed algorithm, and a heuristic method for determining necessary patterns. Subsequently, we briefly describe the proposed algorithm.

A. Determine Frequent Fuzzy Grids

Suppose that each quantitative attribute x_m is partitioned by K linguistic values, and let the universe of discourse $U = \{t_1, t_2, \ldots, t_n\}$. Then, the linguistic value that is assigned to a candidate k-dimensional $(1 \le k \le d)$ fuzzy grid, e.g., $A^{x_1}_{K,i_1} \wedge A^{x_2}_{K,i_2} \wedge \cdots \wedge A^{x_{k-1}}_{K,i_{k-1}} \wedge A^{x_k}_{K,i_k}$, can be alternatively represented as

$$A^{x_1}_{K,i_1} \wedge A^{x_2}_{K,i_2} \wedge \cdots \wedge A^{x_{k-1}}_{K,i_{k-1}} \wedge A^{x_k}_{K,i_k} = \sum_{p=1}^{n} \mu_{A^{x_1}_{K,i_1} \wedge A^{x_2}_{K,i_2} \wedge \cdots \wedge A^{x_{k-1}}_{K,i_{k-1}} \wedge A^{x_k}_{K,i_k}}(t_p)/t_p. \quad (9)$$

The degree to which t_p belongs to

$$A^{x_1}_{K,i_1} \wedge A^{x_2}_{K,i_2} \wedge \cdots \wedge A^{x_{k-1}}_{K,i_{k-1}} \wedge A^{x_k}_{K,i_k}$$

[i.e., $\mu_{A^{x_1}_{K,i_1} \wedge A^{x_2}_{K,i_2} \wedge \cdots \wedge A^{x_{k-1}}_{K,i_{k-1}} \wedge A^{x_k}_{K,i_k}}(t_p)$] can be computed as

$$\mu^{x_1}_{K,i_1}(t_{p_1}) \cdot \mu^{x_2}_{K,i_2}(t_{p_2}) \cdot \ldots \cdot \mu^{x_{k-1}}_{K,i_{k-1}}(t_{p_{k-1}}) \cdot \mu^{x_k}_{K,i_k}(t_{p_k})$$

by the algebraic product. It is noted that, in comparison with the other intersection operators such as the minimum operator or the drastic product, the algebraic product "gently" performs the fuzzy intersection. To check whether or not this fuzzy grid is frequent, we formally define the fuzzy support $\mathrm{FS}(A^{x_1}_{K,i_1} \wedge A^{x_2}_{K,i_2} \wedge \cdots \wedge A^{x_{k-1}}_{K,i_{k-1}} \wedge A^{x_k}_{K,i_k})$ of $A^{x_1}_{K,i_1} \wedge A^{x_2}_{K,i_2} \wedge \cdots \wedge A^{x_{k-1}}_{K,i_{k-1}} \wedge A^{x_k}_{K,i_k}$ as follows:

$$\begin{aligned}\mathrm{FS}&\left(A^{x_1}_{K,i_1} \wedge A^{x_2}_{K,i_2} \wedge \cdots \wedge A^{x_{k-1}}_{K,i_{k-1}} \wedge A^{x_k}_{K,i_k}\right)\\ &= \left[\sum_{p=1}^{n} \mu^{x_1}_{K,i_1}(t_{p_1}) \cdot \mu^{x_2}_{K,i_2}(t_{p_2}) \cdot \ldots \cdot \right.\\ &\quad \left. \mu^{x_{k-1}}_{K,i_{k-1}}(t_{p_{k-1}}) \cdot \mu^{x_k}_{K,i_k}(t_{p_k})\right] \Big/ n \end{aligned} \quad (10)$$

where n is the number of training samples. When $\mathrm{FS}(A^{x_1}_{K,i_1} \wedge A^{x_2}_{K,i_2} \wedge \cdots \wedge A^{x_{k-1}}_{K,i_{k-1}} \wedge A^{x_k}_{K,i_k})$ is larger than or equal to the user-specified minimum fuzzy support (min FS), we can say that $A^{x_1}_{K,i_1} \wedge A^{x_2}_{K,i_2} \wedge \cdots \wedge A^{x_{k-1}}_{K,i_{k-1}} \wedge A^{x_k}_{K,i_k}$ is a frequent k-dimensional fuzzy grid. A frequent fuzzy grid actually stands for a useful pattern discovered from a relational database by the proposed data mining technique. This is similar to defining a frequent k-item set, the support of which is larger than or equal to the user-specified minimum support, used in the *apriori* algorithm. It is noted that, if there exist d attributes of a database, then the dimensions of a fuzzy grid is at most d.

B. Data Structures and Tabular Operations

Significantly, we employ the tabular FGTTFS to generate frequent fuzzy grids, consisting of the following substructures:

1) Fuzzy grids table (FG): each row represents a fuzzy grid, and each column represents a 1-D fuzzy grid denoted by $A^{x_m}_{K,i_m}$.
2) Transaction table (TT): each column represents t_p, and each element records the membership degree to which t_p belongs to the corresponding fuzzy grid.
3) Column FS: stores the fuzzy support corresponding to the fuzzy grid in FG.

An example of an initial tabular FGTTFS is shown in Table I, from which we can see that there are two tuples, t_1 and t_2, and two attributes, x_1 and x_2, in a database relation. Both x_1 and x_2 are divided into three fuzzy partitions. Since each row of FG is a bit string consisting of 0 and 1, $\mathrm{FG}[u]$ and $\mathrm{FG}[v]$ (i.e., the uth row and the vth row of FG) can be paired to generate desired results by applying the Boolean operations.

Example: If we apply the OR operation on two rows, $\mathrm{FG}[1] = (1,0,0,0,0,0)$ and $\mathrm{FG}[4] = (0,0,0,1,0,0)$, then $(\mathrm{FG}[1]\ \mathrm{OR}\ \mathrm{FG}[4]) = (1,0,0,1,0,0)$ corresponding to a candidate 2-D fuzzy grid (i.e., $A^{x_1}_{3,1} \wedge A^{x_2}_{3,1}$) is generated. Then, $\mathrm{FS}(A^{x_1}_{3,1} \wedge A^{x_2}_{3,1}) = \mathrm{TT}[1] \cdot \mathrm{TT}[4] = [\mu^{x_1}_{3,1}(t_{1_1}) \cdot \mu^{x_2}_{3,1}(t_{1_2}) + \mu^{x_1}_{3,1}(t_{2_1}) \cdot \mu^{x_2}_{3,1}(t_{2_2})]/2$ is obtained for comparison with the min FS. If $A^{x_1}_{3,1} \wedge A^{x_2}_{3,1}$ is large, then corresponding data (i.e., FG[1] OR FG[4], $\mathrm{TT}_3[1] \cdot \mathrm{TT}_3[4]$, and $\mathrm{FS}(A^{x_1}_{3,1} \wedge A^{x_2}_{3,1})$ will be inserted into corresponding data structures (i.e., FG, TT, and FS).

In the well-known *apriori* algorithm, two frequent $(k-1)$ item sets are joined to be a candidate k- item set $(3 \le k \le d)$, and these two frequent item sets share $(k-2)$ items. Similarly, a candidate k-dimensional fuzzy grid is also derived by joining two frequent $(k-1)$-dimensional fuzzy grids, and these two frequent grids share $(k-2)$ linguistic values. We define that, if any two frequent $(k-1)$-dimensional fuzzy grids share $(k-2)$ linguistic values, then there exists the same $(k-2)$ linguistic values in those two fuzzy grids.

Example: If $A^{x_1}_{3,2} \wedge A^{x_2}_{3,1}$ and $A^{x_1}_{3,2} \wedge A^{x_3}_{3,3}$ are two frequent fuzzy grids, then we can use $A^{x_1}_{3,2} \wedge A^{x_2}_{3,1}$ and $A^{x_1}_{3,2} \wedge A^{x_3}_{3,3}$ to generate the candidate three–dimensional (3-D) fuzzy grid $A_{3,2}^{x_1} \wedge A_{3,1}^{x_2} \wedge A_{3,3}^{x_3}$ because both $A_{3,2}^{x_1} \wedge A_{3,1}^{x_2}$ and $A_{3,2}^{x_1} \wedge A_{3,3}^{x_3}$ share the linguistic value $A_{3,2}^{x_1}$. However, $A_{3,2}^{x_1} \wedge A_{3,1}^{x_2} \wedge A_{3,3}^{x_3}$ can also be constructed from $A_{3,2}^{x_1} \wedge A_{3,1}^{x_2}$ and $A_{3,1}^{x_2} \wedge A_{3,3}^{x_3}$. This means that we must ensure that no extra constructions of a candidate fuzzy grid are made.

To cope with this problem, the method we adopt here is that if there exist k integers $e_1, e_2, \ldots, e_{k-1}, e_k$ where $1 \le e_1 < e_2 < \cdots < e_{k-1} < e_k \le d$, such that $\mathrm{FG}[u, e_1] = \mathrm{FG}[u, e_2] = \cdots = \mathrm{FG}[u, e_{k-2}] = \mathrm{FG}[u, e_{k-1}] = 1$ and $\mathrm{FG}[v, e_1] = \mathrm{FG}[v, e_2] = \cdots = \mathrm{FG}[v, e_{k-2}] = \mathrm{FG}[v, e_k] = 1$, where $\mathrm{FG}[u]$ and $\mathrm{FG}[v]$ correspond to large $(k-1)$-dimensional fuzzy grids; then $\mathrm{FG}[u]$ and $\mathrm{FG}[v]$ can be paired to generate a candidate k-dimensional fuzzy grid. However, it should be noted that any two linguistic values defined in the same linguistic variable (i.e., attribute) cannot be constructed into a fuzzy grid. For example, since $\mathrm{FG}[1]$ OR $\mathrm{FG}[2] = (1,0,0,0,0,0)$ OR $(0,1,0,0,0,0)$, $(1,1,0,0,0,0)$ is thus invalid. Therefore, $(1,0,1,0,0,0)$, $(0,0,0,1,1,0)$ and $(0,0,0,1,0,1)$ are invalid.

C. Determine Necessary Patterns for Learning

We consider that it is not necessary for decision makers to learn all useful patterns. We previously proposed a heuristic method [20] to determine which frequent fuzzy grids are necessary, and this method is incorporated into the proposed data mining technique. For any two frequent fuzzy grids, e.g., $A_{K_1,i_1}^{x_1} \wedge A_{K_2,i_2}^{x_2} \wedge \cdots \wedge A_{K_{m-1},i_{m-1}}^{x_{m-1}} \wedge A_{K_m,i_m}^{x_m}$ and $A_{K_1,i_1}^{x_1} \wedge A_{K_2,i_2}^{x_2} \wedge \cdots \wedge A_{K_{m-1},i_{m-1}}^{x_{m-1}} \wedge A_{K_m,i_m}^{x_m} \wedge A_{K_{m+1},i_{m+1}}^{x_{m+1}}$, since $\mu_{A_{K,i_1}^{x_1} \wedge A_{K,i_2}^{x_2} \wedge \cdots \wedge A_{K,i_m}^{x_m} \wedge A_{K,i_{m+1}}^{x_{m+1}}}(c_r) \le \mu_{A_{K,i_1}^{x_1} \wedge A_{K,i_2}^{x_2} \wedge \cdots \wedge A_{K,i_{m-1}}^{x_{m-1}} \wedge A_{K,i_m}^{x_m}}(c_r)$ from (9), $A_{K_1,i_1}^{x_1} \wedge A_{K_2,i_2}^{x_2} \wedge \cdots \wedge A_{K_{m-1},i_{m-1}}^{x_{m-1}} \wedge A_{K_m,i_m}^{x_m} \wedge A_{K_{m+1},i_{m+1}}^{x_{m+1}} \subseteq A_{K_1,i_1}^{x_1} \wedge A_{K_2,i_2}^{x_2} \wedge \cdots \wedge A_{K_{m-1},i_{m-1}}^{x_{m-1}} \wedge A_{K_m,i_m}^{x_m}$ thus holds. For two frequent fuzzy grids, e.g., F_i and F_j $(i \ne j)$, if F_j is contained in F_i (i.e., $F_j \subseteq F_i$), then F_j is necessary and F_i is unnecessary. That is, if F_j is acquired by decision makers, then it is unnecessary for them to learn F_i. In addition, F_i is not selected for generating learning sequences. For example, in comparison with $A_{3,2}^{\mathrm{Age}} \wedge A_{2,2}^{\mathrm{Numcars}} \wedge A_{3,2}^{\mathrm{Income}}$, $A_{3,2}^{\mathrm{Age}} \wedge A_{2,2}^{\mathrm{Numcars}}$ is unnecessary for decision makers to learn since $A_{3,2}^{\mathrm{Age}} \wedge A_{2,2}^{\mathrm{Numcars}} \wedge A_{3,2}^{\mathrm{Income}}$ is contained in $A_{3,2}^{\mathrm{Age}} \wedge A_{2,2}^{\mathrm{Numcars}}$.

We now briefly describe the algorithm for finding frequent and necessary fuzzy grids.

Algorithm: Algorithm for Finding Frequent and Necessary Fuzzy Grids
Input: a: A relational database
b. User-specified minimum fuzzy support
Output: Frequent and necessary fuzzy grids
Method:
Step 1. Fuzzy partitioning in each attribute.
Step 2. Scan the database, and then construct the initial table FGTTFS.
Step 3. Generate frequent fuzzy grids.
3-1: Generate frequent 1-D fuzzy grids
Set $k = 1$ and eliminate the rows of initial FGTTFS corresponding to candidate 1-D fuzzy grids that are not frequent.
3-2: Generate frequent k-dimensional fuzzy grids
Set $k + 1$ to k.
For any two unpaired rows, $\mathrm{FGTTFS}[u]$ *and* $\mathrm{FGTTFS}[v]$ $(u \ne v)$, *corresponding to frequent* $(k-1)$*-dim fuzzy grids do*
3-2-1. From $(\mathrm{FG}[u]\ \mathrm{OR}\ \mathrm{FG}[v])$ corresponding to a candidate k-dimensional fuzzy grid c,
if any two linguistic values are defined in the same linguistic variable (i.e., attribute), then discard c and skip Steps 3-2-2, 3-2-3, and 3-2-4. That is, c is invalid.
3-2-2. If $\mathrm{FG}[u]$ and $\mathrm{FG}[v]$ do not share $(k-2)$ linguistic values, then discard c and skip Steps 3-2-3 and 3-2-4. That is, c is invalid.
3-2-3. If there exist integers $1 \le e_1 < e_2 < \cdots < e_k \le d$ such that $(\mathrm{FG}[u]\ \mathrm{OR}\ \mathrm{FG}[v])[e_1] = (\mathrm{FG}[u]\ \mathrm{OR}\ \mathrm{FG}[v])[e_2] = \cdots = (\mathrm{FG}[u]\ \mathrm{OR}\ \mathrm{FG}[v])[e_{k-1}] = (\mathrm{FG}_K[u]\ \mathrm{OR}\ \mathrm{FG}[v])[e_k] = 1$, then compute $(\mathrm{TT}[e_1] \cdot \mathrm{TT}[e_2] \cdot \ldots \cdot \mathrm{TT}[e_k])$ and the fuzzy support of c.
3-2-4. Add $(\mathrm{FG}[u]\ \mathrm{OR}\ \mathrm{FG}[v]))$ to FG, $(\mathrm{TT}[e_1] \cdot \mathrm{TT}[e_2] \cdot \ldots \cdot \mathrm{TT}[e_k])$ to TT and the fuzzy support to FS when fs is not smaller than the minimum fuzzy support; otherwise, discard c.
End
3-3. Check whether any frequent k-dimensional fuzzy grid is generated or not
If any frequent k-dimensional fuzzy grid is generated, then go to Step 3-2. It is noted that the final FGTTFS stores only frequent fuzzy grids.
Step 4. Find necessary fuzzy grids
For any two rows, $\mathrm{FCTTFS}[u]$ and $\mathrm{FCTTFS}[v]$ $(u \ne v)$, *corresponding to frequent fuzzy grids do*
If $\mathrm{FC}[u] = (\mathrm{FC}[u]\ \mathrm{AND}\ \mathrm{FC}[v])$, then frequent fuzzy grids corresponding to $\mathrm{FC}[u]$ is unnecessary, and $\mathrm{FCTTFS}[u]$ is eliminated; else, $\mathrm{FCTTFS}[v]$ is eliminated.
End

Since the set of frequent and necessary fuzzy grids is viewed as a competence set for solving one decision problem, we try to use a method proposed by Feng and Yu [6] to expand it. This effective method is introduced in Section IV.

IV. Competence Set Expansion

As we have mentioned earlier, for each decision problem E, there is a competence set, denoted by $\mathrm{CS}(E)$, consisting of ideas, knowledge, information, and skills required for successfully solving the problem. In addition, there exists a skill set, denoted by $\mathrm{Sk}(E)$, that has been acquired by decision makers. Roughly speaking, $\mathrm{CS}(E)$ is the union of the already acquired competence set (i.e., $\mathrm{Sk}(E)$) and the truly needed competence set denoted by $\mathrm{Tr}(E)$. However, it seems that it is not sufficient to solve E only by $\mathrm{Sk}(E)$. That is, in order to solve problem E, decision makers must acquire $\mathrm{Tr}(E) \backslash \mathrm{Sk}(E)$ through the competence set expansion. A competence set expansion means we are trying to find an effective way to generate learning sequences of acquiring the needed skills so that the needed competence set $\mathrm{Tr}(E) \backslash \mathrm{Sk}(E)$ is obtained [6]. We depict the concept of the competence set expansion in Fig. 5. The shaded area shown in Fig. 5 is just $\mathrm{Tr}(E) \backslash \mathrm{Sk}(E)$. Since the set of the useful patterns discovered from a relational database can be viewed as a set of needed skills for solving one decision problem, e.g., E, it is necessary to generate learning sequences of useful patterns. However, for simplicity $\mathrm{Sk}(E)$ is not discussed in this paper. Previously, some methods for expanding the competence set were proposed, such as deduction graphs by Li and Yu [19] and the minimum spanning table method by Feng and Yu [6].

A competence set expansion can be roughly regarded as a tree construction process. For example, Feng and Yu employed a directed graph with an expansion table to find a spanning tree. This procedure views each useful pattern in the competence set to be a node in a directed graph, and set the learning cost $c(y_i, y_j)$, which may be measured by time or money, in the directed path directly from node y_i to node y_j. It is noted that $c(y_i, y_j) \ne c(y_j, y_i)$ usually holds. Then. in this graph

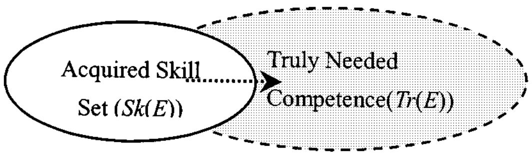

Fig. 5. Competence set "Expansion."

we can find a spanning tree with minimum learning cost (i.e., minimum spanning tree). An optimal expansion with minimum cost is thus acquired from the minimum spanning tree. The starting node in the directed path is the pattern that we suggest decision makers learn first. The minimum spanning table method is especially powerful for the expansion of a set of single skills. Therefore, we adopt this powerful algorithm to generate a minimum spanning tree. We briefly introduce the minimum spanning table method as follows.

Algorithm: ***Minimum spanning table method***

Input: A directed graph with an expansion table T_0. The element (y_i, y_j) of T_0 stores the learning cost $c(y_i, y_j)$ directly from node y_i to node y_j. Initially, no columns of T_0 are crossed out and an integer number, z, is set to zero.

Output: The minimum spanning table ST_0 and corresponding spanning tree.

Method:

Step 1. Selecting and marking procedure

Select the smallest element $c(y_i, y_j)$ $(i \neq j)$ among the remaining not-crossed-out columns of expansion table T_z, and mark it.

Step 2. Cycle detecting procedure

Determine whether a cycle has been formed among the marked elements; if so, go to Step 5.

Step 3. Crossing out procedure

Cross out the column to which the newly selected marked element belongs.

Step 4. Stopping rule

If only one not-crossed-out column is left, then the minimum spanning table ST_z can be constructed, and go to Step 6; otherwise, go to Step 1.

Step 5. Compressing procedure

Compress the nodes in the detected cycle C into a node $\underline{x}$. Define transformation equations as follows:

$$c(\underline{x}, y_i) = \min\{c(y, y_i) \mid y \in \underline{x}\} \tag{11}$$

$$c(y_i, \underline{x}) = \min\{c(y_i, y) + c(y_s, yt) - c(y_\alpha, y) \mid y \in \underline{x}\} \tag{12}$$

where $c(y_s, y_t)$ is the largest cost in C; and (y_α, y) is a marked element. Set $z+1$ to z, and a new expansion table T_z is constructed, and then go to Step 1.

Step 6. Unfolding procedure

From the minimum spanning table of T_z, the minimum spanning table of $T_z, T_{z-1}, \ldots, T_0$ can be generated. Note that ST_{z-1} produced by the unfolding procedure is the minimum spanning table of T_{z-1}.

The minimum spanning table method will be stopped in a finite number of steps, at which point an optimal expansion of the competence set can be acquired from a spanning tree with minimum learning cost. In following section, we present a numerical example to demonstrate its ability to help decision makers solve decision problems through the data mining technique and the competence set expansion.

V. Numerical Example

In this section, a specified database relation EMP stores the basic data of customers for one supermarket, with five attributes and ten tuples t_p $(1 \leq p \leq 10)$, as shown in Table II. For simplicity, some columns or rows of the following tables are omitted, as denoted by "..."

Phase I. Find frequent and necessary fuzzy grids

- **Fuzzy partitioning in each attribute**

Since both "Age" and "Income" are quantitative attributes, $K = 3$ is considered for these two attributes. Suppose the a universe of discourse of "Age" is [0, 60], and that of Income is [15 000, 60 000], then there are six linguistic values (i.e., three for Age and three for Income) generated in the following:

Age: $A_{3,1}^{\mathrm{Age}}$: young, and below 30;
$A_{3,2}^{\mathrm{Age}}$: close to 30, and between 0 to 60;
$A_{3,3}^{\mathrm{Age}}$: old, and above 30.
Income: $A_{3,1}^{\mathrm{Income}}$: low, and below $37\,500$;
$A_{3,2}^{\mathrm{Income}}$: close to 37500, and between $15\,000$ and $60\,000$;
$A_{3,3}^{\mathrm{Income}}$: high, and above $37\,500$.

On the other hand, "Married," "Numcars," and "Career" are categorical attributes. Various linguistic values defined in each attribute are described as follows:

Married: $A_{2,1}^{\mathrm{Married}}$: yes; $A_{2,2}^{\mathrm{Married}}$: no.
Numcars: $A_{2,1}^{\mathrm{Married}}$: zero Car; $A_{2,1}^{\mathrm{Numcars}}$: one car.
Career: $A_{4,1}^{\mathrm{Career}}$: student; $A_{4,2}^{\mathrm{Career}}$: teacher;
$A_{4,3}^{\mathrm{Career}}$: engineer; $A_{4,4}^{\mathrm{Career}}$: business person.

- **Find frequent 1-D fuzzy grids**

After scanning EMP, we can construct the initial FCTTFS. Assuming that the user-specified min FS is 0.3, then we can find frequent 1-D fuzzy grids by deleting the infrequent fuzzy grids from the initial FCTTFS. We reconstruct FCTTFS, as shown in Table III(a) and (b).

- **Find frequent 2-D fuzzy grids**

Now, rows 1, 2, 3, 4, 5, 6, and 7 correspond to frequent 1-D fuzzy grids $A_{3,2}^{\mathrm{Age}}, A_{2,1}^{\mathrm{Married}}, A_{2,2}^{\mathrm{Married}}, A_{2,2}^{\mathrm{Numcars}}, A_{3,2}^{\mathrm{Income}}, A_{4,3}^{\mathrm{Career}}$, and $A_{4,4}^{\mathrm{Career}}$, respectively. We can find that $(\mathrm{FC}[2] \text{ OR } \mathrm{FC}[3]) = (0,1,1,0,0,0,0)$, which corresponds to the candidate 2-D fuzzy grid $c = A_{2,1}^{\mathrm{Married}} \times A_{2,2}^{\mathrm{Married}}$ is generated. However, c will be discarded since both $A_{2,1}^{\mathrm{Married}}$ and $A_{2,2}^{\mathrm{Married}}$ are linguistic values of the linguistic variable "Married."

We select FCTTFS[1] and FCTTFS[2] as an example to show how a frequent 2-D fuzzy grid is generated. Clearly, $\mathrm{FC}[1]$ and $\mathrm{TT}[1]$ corresponding to $A_{3,2}^{\mathrm{Age}}$ are (1, 0, 0, 0, 0, 0, 0) and (0.6333, 0.8333, 0.7667, 0.90, 0.50, 0.1333, 0.60, 0.6667, 0.90, 0.8333), respectively. In addition, FC[2] and TT[2] corresponding to $A_{2,1}^{\mathrm{Married}}$ are (0, 1, 0, 0, 0, 0, 0) and (0.0, 1.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 1.0, 1.0), respectively. $(\mathrm{FC}[1] \text{ OR } \mathrm{FC}[2]) = (1,1,0,0,0,0,0)$ corresponding to a candidate 2-D fuzzy grid (i.e., $c = A_{3,2}^{\mathrm{Age}} \wedge A_{2,1}^{\mathrm{Married}}$) is generated. Then, we calculate $(\mathrm{TT}[1] \cdot \mathrm{TT}[2]) = (0.0, 0.8333, 0.0, 0.90, 0.50, 0.1333, 0.0, 0.0, 0.90, 0.8333)$ and the fuzzy support $\mathrm{FS}(c) = 0.410$ is computed. Since $\mathrm{FS}(c) \geq 0.3$, c can thus be inserted to FCTTFS. Finally,

TABLE II
RELATION EMP

t_p	Age	Married	Numcars	Income	Career
t_1	19	N	0	20000	Student
t_2	35	Y	1	50000	Teacher
t_3	23	N	1	33000	Engineer
t_4	33	Y	1	35000	Engineer
t_5	45	Y	1	50000	Trader
t_6	56	Y	1	45000	Trader
t_7	18	N	0	25000	Student
t_8	20	N	1	30000	Engineer
t_9	33	Y	1	35000	Engineer
t_{10}	35	Y	1	45000	Trader

TABLE III
(a) FREQUENT 1-DIM FUZZY GRIDS IN TABLE FC. (b) FREQUENT 1-DIM FUZZY GRIDS IN TABLE TTFC

Fuzzy grid	$A_{3,2}^{\text{Age}}$	$A_{2,1}^{\text{Married}}$	$A_{2,2}^{\text{Married}}$	$A_{2,2}^{\text{Numcars}}$	$A_{3,2}^{\text{Income}}$	$A_{4,3}^{\text{Career}}$	$A_{4,4}^{\text{Career}}$
$A_{3,2}^{\text{Age}}$	1	0	0	0	0	0	0
$A_{2,1}^{\text{Married}}$	0	1	0	0	0	0	0
$A_{2,2}^{\text{Married}}$	0	0	1	0	0	0	0
$A_{2,2}^{\text{Numcars}}$	0	0	0	1	0	0	0
$A_{3,2}^{\text{Income}}$	0	0	0	0	1	0	0
$A_{4,3}^{\text{Career}}$	0	0	0	0	0	1	0
$A_{4,4}^{\text{Career}}$	0	0	0	0	0	0	1

(a)

TT										FS
t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	
0.6333	0.8333	0.7667	0.90	0.50	0.1333	0.60	0.6667	0.90	0.8333	0.6767
0.0	1.0	0.0	1.0	1.0	1.0	0.0	0.0	1.0	1.0	0.6
1.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.4
0.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	0.8
0.2222	0.4444	0.80	0.8889	0.4444	0.6667	0.4444	0.6667	0.8889	0.6667	0.6133
0.0	0.0	1.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	0.4
0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	1.0	0.3

(b)

frequent 2-D fuzzy grids that can be inserted into the table FCTTFS are shown in Table IV(a) and (b).

- **Find frequent 3-D fuzzy grids**

  In this step, six frequent 3-D fuzzy grids can be further discovered from those frequent 2-D fuzzy grids. These frequent 3-D fuzzy grids are $A_{3,2}^{\text{Age}} \wedge A_{2,1}^{\text{Married}} \wedge A_{2,2}^{\text{Numcars}}$, $A_{3,2}^{\text{Age}} \wedge A_{2,2}^{\text{Numcars}} \wedge A_{3,2}^{\text{Income}}$, $A_{3,2}^{\text{Age}} \wedge A_{2,2}^{\text{Numcars}} \wedge A_{4,3}^{\text{Career}}$, $A_{2,1}^{\text{Married}} \wedge A_{2,2}^{\text{Numcars}} \wedge A_{3,2}^{\text{Income}}$, $A_{2,1}^{\text{Married}} \wedge A_{2,2}^{\text{Numcars}} \wedge A_{4,4}^{\text{Career}}$, $A_{2,2}^{\text{Numcars}} \wedge A_{3,2}^{\text{Income}} \wedge A_{4,3}^{\text{Career}}$. The corresponding FCTTFS are omitted here for simplicity.
- **Find necessary patterns for finding learning sequences**

  We can observe that there is no frequent four-dimensional (4-D) fuzzy grid that can be generated. Thus, we must find necessary patterns from frequent 1-D, 2-D, and 3-D fuzzy grids. It is easy to find six frequent and necessary fuzzy grids, consisting of $A_{3,2}^{\text{Age}} \wedge A_{2,1}^{\text{Married}} \wedge A_{2,2}^{\text{Numcars}}$, $A_{3,2}^{\text{Age}} \wedge A_{2,2}^{\text{Numcars}} \wedge A_{3,2}^{\text{Income}}$, $A_{3,2}^{\text{Age}} \wedge A_{2,2}^{\text{Numcars}} \wedge A_{4,3}^{\text{Career}}$, $A_{2,1}^{\text{Married}} \wedge A_{2,2}^{\text{Numcars}} \wedge A_{3,2}^{\text{Income}}$, $A_{2,1}^{\text{Married}} \wedge A_{2,2}^{\text{Numcars}} \wedge A_{4,4}^{\text{Career}}$, and $A_{2,2}^{\text{Numcars}} \wedge A_{3,2}^{\text{Income}} \wedge A_{4,3}^{\text{Career}}$.

Phase II. Competence set expansion

Next, we employ the minimum spanning table method to expand the competence set for necessary and frequent fuzzy grids discovered from Phase I. At first, each necessary fuzzy grid corresponding to a node in the digraph is shown in Table V. An expansion table T_0 is constructed as Table VI with hypothesis learning cost.

Finally, ST_0 of T_0 is shown in both the alternatives of Tables VII and VIII. The minimum spanning trees corresponding to Tables VI and VIII are shown in Figs. 6 and 7, respectively. Clearly, the minimum learning cost is 2.7245. Learning sequences with minimum learning cost can be directly acquired by the minimum spanning trees. For example, the learning sequence shown in Fig. 6 represents that $A_{2,1}^{\text{Married}} \wedge A_{2,2}^{\text{Numcars}} \wedge A_{4,4}^{\text{Career}}$ (i.e., α_5) is the useful pattern that we suggest decision makers to learn first. $A_{2,1}^{\text{Married}} \wedge A_{2,2}^{\text{Numcars}} \wedge A_{3,2}^{\text{Income}}$ (i.e., α_4) are learned after $A_{2,1}^{\text{Married}} \wedge A_{2,2}^{\text{Numcars}} \wedge A_{4,4}^{\text{Career}}$ have been learned. Subsequently, $A_{2,2}^{\text{Numcars}} \wedge A_{3,2}^{\text{Income}} \wedge A_{4,3}^{\text{Career}}$ (i.e., α_6) are suggested to be learned. The main difference between Figs. 6 and 7 is that $A_{3,2}^{\text{Age}} \wedge A_{2,2}^{\text{Numcars}} \wedge A_{3,2}^{\text{Income}}$ (i.e., α_2) and $A_{3,2}^{\text{Age}} \wedge A_{2,2}^{\text{Numcars}} \wedge A_{4,3}^{\text{Career}}$ (i.e., α_3) are suggested to be learned simultaneously in Fig. 7. Since learning sequences are not unique, decision makers will subjectively select one of the learning sequences to acquire the competence set.

TABLE IV
(a) FREQUENT 2-DIM FUZZY GRIDS IN TABLE FC. (b) FREQUENT 2-DIM FUZZY GRIDS IN TABLE TTFC

Fuzzy grid	$A_{3,2}^{Age}$	$A_{2,1}^{Married}$	$A_{2,2}^{Married}$	$A_{2,2}^{Numcars}$	$A_{3,2}^{Income}$	$A_{4,3}^{Career}$	$A_{4,4}^{Career}$
$A_{3,2}^{Age} \wedge A_{2,1}^{Married}$	1	1	0	0	0	0	0
$A_{3,2}^{Age} \wedge A_{2,2}^{Numcars}$	1	0	0	1	0	0	0
$A_{3,2}^{Age} \wedge A_{3,2}^{Income}$	1	0	0	0	0	0	0
$A_{3,2}^{Age} \wedge A_{4,3}^{Career}$	1	0	0	0	0	1	0
$A_{2,1}^{Married} \wedge A_{2,2}^{Numcars}$	0	1	0	1	0	0	0
$A_{2,1}^{Married} \wedge A_{3,2}^{Income}$	0	1	0	0	0	0	0
$A_{2,1}^{Married} \wedge A_{4,4}^{Career}$	0	1	0	0	0	0	1
$A_{2,2}^{Numcars} \wedge A_{3,2}^{Income}$	0	0	0	1	1	0	0
$A_{2,2}^{Numcars} \wedge A_{4,3}^{Career}$	0	0	0	1	0	1	0
$A_{2,2}^{Numcars} \wedge A_{4,4}^{Career}$	0	0	0	1	0	0	1
$A_{3,2}^{Income} \wedge A_{4,3}^{Career}$	0	0	0	0	1	1	0

(a)

TT										FS
t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	
0.0	0.8333	0.0	0.90	0.50	0.1333	0.0	0.0	0.90	0.8333	0.41
0.0	0.8333	0.7667	0.90	0.50	0.1333	0.0	0.6667	0.90	0.8333	0.5533
0.1407	0.3703	0.6134	0.80	0.2222	0.0889	0.2666	0.4445	0.80	0.5556	0.4302
0.0	0.0	0.7667	0.90	0.0	0.0	0.0	0.6667	0.90	0.0	0.3233
0.0	1.0	0.0	1.0	1.0	1.0	0.0	0.0	1.0	1.0	0.6
0.0	0.4444	0.0	0.8889	0.4444	0.6667	0.0	0.0	0.8889	0.6667	0.4
0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	1.0	0.3
0.0	0.4444	0.80	0.8889	0.4444	0.6667	0.0	0.6667	0.8889	0.6667	0.5467
0.0	0.0	1.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	0.4
0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	1.0	0.3
0.0	0.0	0.80	0.8889	0.0	0.0	0.0	0.6667	0.8889	0.0	0.3245

(b)

TABLE V
EACH FREQUENT AND NECESSARY FUZZY GRID CORRESPONDS TO A NODE

Node	Frequent and necessary fuzzy grids
α_1	$A_{3,2}^{Age} \wedge A_{2,1}^{Married} \wedge A_{2,2}^{Numcars}$
α_2	$A_{3,2}^{Age} \wedge A_{2,2}^{Numcars} \wedge A_{3,2}^{Income}$
α_3	$A_{3,2}^{Age} \wedge A_{2,2}^{Numcars} \wedge A_{4,3}^{Career}$
α_4	$A_{2,1}^{Married} \wedge A_{2,2}^{Numcars} \wedge A_{3,2}^{Income}$
α_5	$A_{2,1}^{Married} \wedge A_{2,2}^{Numcars} \wedge A_{4,4}^{Career}$
α_6	$A_{2,2}^{Numcars} \wedge A_{3,2}^{Income} \wedge A_{4,3}^{Career}$

TABLE VI
EXPANSION TABLE T_0

	α_1	α_2	α_3	α_4	α_5	α_6
α_1	*	0.6297	0.8333	0.5556	1.0	0.8333
α_2	0.6134	*	0.5556	0.6134	0.8	0.5556
α_3	0.7667	0.5555	*	0.7667	0.9	0.3333
α_4	0.6667	0.6667	0.6667	*	0.8889	0.6667
α_5	0.8667	0.9111	1.0	0.5556	*	1.0
α_6	0.8	0.5555	0.3333	0.8	0.8889	*

TABLE VII
SELECT $c(\alpha_3, \alpha_2)$ AND $c(\alpha_4, \alpha_6)$, AND THEN CONSTRUCT ST_0 OF T_0

	α_1	α_2	α_3	α_4	α_5	α_6
α_1	*	0.6297	0.8333	0.5556	1.0	0.8333
α_2	0.6134	*	0.5556	0.6134	0.8	0.5556
α_3	0.7667	0.5555	*	0.7667	0.9	0.3333
α_4	0.6667	0.6667	0.6667	*	0.8889	0.6667
α_5	0.8667	0.9111	1.0	0.5556	*	1.0
α_6	0.8	0.5555	0.3333	0.8	0.8889	*

TABLE VIII
SELECT $c(\alpha_6, \alpha_2)$ AND $c(\alpha_4, \alpha_6)$, AND THEN CONSTRUCT ST_0 OF T_0

	α_1	α_2	α_3	α_4	α_5	α_6
α_1	*	0.6297	0.8333	0.5556	1.0	0.8333
α_2	0.6134	*	0.5556	0.6134	0.8	0.5556
α_3	0.7667	0.5555	*	0.7667	0.9	0.3333
α_4	0.6667	0.6667	0.6667	*	0.8889	0.6667
α_5	0.8667	0.9111	1.0	0.5556	*	1.0
α_6	0.8	0.5555	0.3333	0.8	0.8889	*

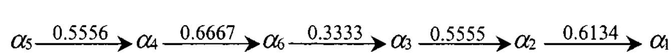

Fig. 6. Minimum spanning tree corresponds to Table VII.

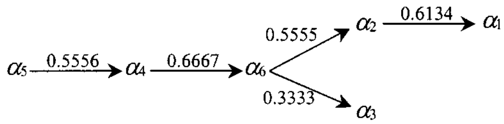

Fig. 7. Minimum spanning tree corresponds to Table VIII.

VI. Discussions and Conclusions

The primary contribution of this paper is to present a useful tool to support decision making through the data mining technique and the competence set expansion. From simulation results of the numerical example, we can see that it is possible to help decision makers to confidently solve decision problems using the data mining technique and the competence set expansion. Significantly, this is a starting point for integrating data mining techniques with the expansion of the competence set. Each data mining technique, such as clustering, can discover its particular type of useful patterns, which can also be viewed as a competence set for solving one decision problem. Then, the optimal expansion of the competence set with minimum learning cost becomes quite important. In this paper, we first use the proposed data mining technique to find the necessary patterns from a database. Then, we use the minimum spanning table method to optimally expand a needed competence set. For the combination of various techniques of data mining and the competence set expansion, we will study the feasibility and effectiveness.

In fact, the meaning of the linguistic values of quantitative attribute x_m can be changed by a linguistic hedge [9], [10], such as "very" or "more or less." For example

$$\text{very}A_{K,i_m}^{x_m} = \left(A_{K,i_m}^{x_m}\right)^2 \tag{13}$$

$$\text{more or less } A_{K,i_m}^{x_m} = \left(A_{K,i_m}^{x_m}\right)^{1/2}. \tag{14}$$

The membership functions of $(A_{K,i_m}^{x_m})^2$ and $(A_{K,i_m}^{x_m})^{1/2}$ are $[\mu_{K,i_m}^{x_m}(x)]^2$ and $[\mu_{K,i_m}^{x_m}(x)]^{1/2}$, respectively. The use of the linguistic hedge will make the frequent fuzzy grids discovered from a database more friendly and more flexible for decision makers. However, the number of linguistic values of each attribute may be available from domain experts.

Acknowledgment

The authors are very grateful to the anonymous referees for their valuable comments and constructive suggestions.

References

[1] M. Berry and G. Linoff, *Data Mining Techniques: For Marketing, Sales, and Customer Support*. New York: Wiley, 1997.

[2] S. Myra, "Web usage mining for web site evaluation," *Commun. ACM*, vol. 43, no. 8, pp. 127–134, 2000.

[3] P. L. Yu and D. Zhang, "A foundation for competence set analysis," *Math. Social Sci.*, vol. 20, pp. 251–299, 1990.

[4] M. J. Hwang, C. I. Chiang, I. C. Chiu, and G. H. Tzeng, "Multistages optimal expansion of competence sets in fuzzy environment," *Int. J. Fuzzy Syst.*, vol. 3, no. 3, pp. 486–492, 2001.

[5] Y. C. Hu, R. S. Chen, and G. H. Tzeng, Discovering fuzzy association rules using fuzzy partition methods, in Knowl.-Based Syst.. (accepted).

[6] J. W. Feng and P. L. Yu, "Minimum spanning table and optimal expansion of competence set," *J. Optim. Theory Appl.*, vol. 99, no. 3, pp. 655–679, 1998.

[7] L. A. Zadeh, "Fuzzy sets," *Inf. Control.*, vol. 8, no. 3, pp. 338–353, 1965.

[8] ——, "The concept of a linguistic variable and its application to approximate reasoning," *Inf. Sci.*, pt. 1, vol. 8, no. 3, pp. 199–249, 1975. (part 2) vol. 8, no. 4, pp. 301–357, 1975; (part 3) vol. 9, no. 1, pp. 43–80, 1976.

[9] H. J. Zimmermann, *Fuzzy Set Theory and Its Applications*. Norwell, MA: Kluwer, 1991.

[10] S. M. Chen and W. T. Jong, "Fuzzy query translation for relational database systems," *IEEE Trans. Syst., Man, Cybern. B*, vol. 27, pp. 714–721, Aug. 1997.

[11] J. W. Han and M. Kamber, *Data Mining: Concepts and Techniques*. San Mateo, CA: Morgan Kaufmann, 2001.

[12] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selecting fuzzy if-then rules for classification problems using genetic algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 260–270, Aug. 1995.

[13] H. Ishibuchi, T. Nakashima, and T. Murata, "Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems," *IEEE Trans. Syst., Man, Cybern. B*, vol. 29, pp. 601–618, Oct. 1999.

[14] Y. C. Hu, R. S. Chen, and G. H. Tzeng, "Mining fuzzy association rules for classification problems," *Comput. Ind. Eng.*, to be published.

[15] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 6, pp. 1414–1427, 1992.

[16] C. T. Sun, "Rule-base structure identification in an adaptive-network-based fuzzy inference system," *IEEE Trans. Fuzzy Syst.*, vol. 2, no. 1, pp. 64–73, 1994.

[17] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.

[18] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo, "Fast discovery of association rules," in *Advances in Knowledge Discovery and Data Mining*, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds: AAAI Press, 1995, pp. 307–328.

[19] H. L. Li and P. L. Yu, "Optimal competence set expansion using deduction graph," *J. Optim. Theory Appl.*, vol. 80, no. 1, pp. 75–91, 1994.

[20] Y. C. Hu, G. H. Tzeng, and R. S. Chen, "Discovering fuzzy concepts for expanding competence set," in *Proc. 2nd Int. Symp. Advanced Intelligent Systems*, Daejeon, Korea, 2001, pp. 396–401.

[21] W. Pedrycz and F. Gomide, *An Introduction to Fuzzy Sets: Analysis and Design*. Cambridge, MA: MIT Press, 1998.

[22] P. Adriaans and D. Zantinge, *Data Mining*. Reading, MA: Addison-Wesley, 1996.

[23] J. M. Li, C. I. Chiang, and P. L. Yu, "Optimal multiple stage expansion of competence set," *Eur. J. Oper. Res.*, vol. 120, no. 3, pp. 511–524, 2000.

[24] P. L. Yu, *Forming Winning Strategies: An Integrated Theory of Habitual Domains*. New York: Springer-Verlag, 1990.

Web Newspaper Layout Optimization Using Simulated Annealing

Jesús González, Ignacio Rojas, Héctor Pomares, Moisés Salmerón, and Juan Julián Merelo

***Abstract*—The web newspaper pagination problem consists of optimizing the layout of a set of articles extracted from several web newspapers and sending it to the user as the result of a previous query. This layout should be organized in columns, as in real newspapers, and should be adapted to the client web browser configuration in real time. This paper presents an approach to the problem based on simulated annealing (SA) that solves the problem on-line, adapts itself to the client's computer configuration, and supports articles with different widths.**

***Index Terms*—Greedy algorithm, pagination, real-time optimization.**

I. Introduction

Since the amount of information available on the Internet is growing day by day, when a user sends a query to a news site, a lot of information

Manuscript received March 12, 2000; revised January 8, 2002. This work was supported in part by the Spanish CICYT Project DPI2001-3219. This paper was recommended by Associate Editor C. Hsu.

The authors are with the Department of Computer Architecture and Computer Technology, E.T.S. Ingeniería Informática, University of Granada, E-18071 Granada, Spain.

Publisher Item Identifier S 1083-4419(02)03554-9.