Knowl Inf Syst (2010) 23:129–152
DOI 10.1007/s10115-009-0223-1

REGULAR PAPER

Mining fuzzy association rules from uncertain data

Cheng-Hsiung Weng · Yen-Liang Chen

Received: 6 May 2008 / Revised: 29 April 2009 / Accepted: 17 May 2009 /
Published online: 25 June 2009

Abstract Association rule mining is an important data analysis method that can discover associations within data. There are numerous previous studies that focus on finding fuzzy association rules from precise and certain data. Unfortunately, real-world data tends to be uncertain due to human errors, instrument errors, recording errors, and so on. Therefore, a question arising immediately is how we can mine fuzzy association rules from uncertain data. To this end, this paper proposes a representation scheme to represent uncertain data. This representation is based on possibility distributions because the possibility theory establishes a close connection between the concepts of similarity and uncertainty, providing an excellent framework for handling uncertain data. Then, we develop an algorithm to mine fuzzy association rules from uncertain data represented by possibility distributions. Experimental results from the survey data show that the proposed approach can discover interesting and valuable patterns with high certainty.

Keywords Learning · Fuzzy statistics and data analysis · Uncertain data · Data mining · Fuzzy association rules

1 Introduction

Association rule mining is an important data mining method that can discover consumer purchasing patterns from transaction databases (see [5,27] for an overview). The association rules mining techniques have been used in many applications, such as market basket analysis [7,10,11,14,26,34], image mining [17], recommender systems [40], medical domain [42] and classifying multiple databases [50]. Furthermore, Berti-Equille [6] integrated data

C.-H. Weng (✉)
Department of Management Information Systems, Central Taiwan University
of Science and Technology, Taichung, 406 Taiwan, ROC
e-mail: 107090@ctust.edu.tw; chweng@mgt.ncu.edu.tw

Y.-L. Chen
Department of Information Management, National Central University, Chung-Li, 320 Taiwan, ROC

quality indicators to propose a cost based probabilistic model for selecting legitimately interesting rules. Lee et al. [39] revise a graph-based algorithm to further speed up the process of itemsets generation.

Agrawal et al. [1] first introduced the problem, and defined it as finding all rules from transaction data satisfying the minimum support and the minimum confidence constraints. Briefly, an association rule mining algorithm works in two steps: (1) generate all frequent itemsets that satisfy the minimum support and (2) generate all association rules that satisfy the minimum confidence from the already discovered frequent itemsets.

Due to its great success and widespread usage, many variants of association rule mining algorithms have been proposed [13,16,38]. Based on different assumptions about the underlying data, different methods have been developed to discover association rules from databases. In general, a common assumption adopted by most previous approaches is that the data are precise and certain. Unfortunately, real-world data tends to be uncertain due to human errors, instrument errors, recording errors, and so on. Therefore, a question arising immediately is how we can mine association rules from uncertain data.

A possibility distribution is a representation of uncertain knowledge and information, where the center reflects the most possible case and the spread reflects the other cases with relatively low possibilities. Possibility theory is an extension of the fuzzy set theory, which describes uncertain and incomplete information [53]. In the past, possibility theory has been popularly used to handle uncertain information. For example, Hsu and Wang [33] proposed a possibility linear programming model to alleviate the influences of demand uncertainty in assemble-to-order (ATO) environments, and further manage production planning problems.

In this paper, we use the possibility theory to handle uncertain data. An item in a transaction is represented by a possibility distribution. When the data item is certain, the distribution can be as simple as a single point. But when the data item is uncertain, the distribution can be as complex as necessary. For example, if an uncertain data has four different plausible values, we can represent it using a discrete possibility distribution with four values.

After transforming the data into possibility distributions, our next problem is finding fuzzy association rules from the transformed data. Our algorithm must be able to discover patterns from possibility distributions rather than crisp data. The difficulty is that a possibility distribution may match a pattern with multiple support values, where a support value is obtained for every plausible value in the distribution. Therefore, this study computes two measures from these values, support and deviation, where the support and the deviation are the mean and the standard deviation of those multiple supports, respectively. A pattern is considered as a good pattern if its support is high but its deviation is low, because low deviation means the pattern is more certain and believable. Accordingly, we use these two measures as the basis for a new algorithm to mine fuzzy association rules from uncertain data represented by possibility distributions.

The rest of this paper is organized as follows. Section 2 reviews related work and Sect. 3 explains how to use possibility theory to represent uncertain data. The problem definitions are given in Sect. 4. The proposed algorithm and an example are illustrated in Sect. 5. Section 6 uses survey data as a case study to demonstrate the usefulness of the proposed algorithm. Conclusions and future works are discussed in Sect. 7.

2 Related work

It is widely recognized that many real world relationships are intrinsically fuzzy. Therefore, the theory of fuzzy sets can help data miners discover novel and meaningful patterns from data [44,54]. Fuzzy set theory [52] is primarily concerned with quantifying and reasoning using natural language, where words can have ambiguous meanings. Delgado et al. [19] considered fuzzy sets as an optimal tool for modeling the imprecise terms and relations commonly employed by humans in communication and understanding.

Determining the membership functions is very important in Fuzzy set applications. Actually, there are two approaches for the acquisition of membership functions: (1) querying experts; (2) applying machine learning techniques. In the second approach, membership functions are derived from training examples by machine learning approaches. Hong and Lee [31] proposed a general learning method that automatically derives membership functions from a set of given training examples using a decision table. Hong and Chen [29] improved Hong and Lee's method by first selecting relevant attributes and then building appropriate initial membership functions. Genetic algorithms (GAs), frequently used in various scientific applications, can also be used to determine the membership functions [3,4].

Recently, the theory of fuzzy sets has been widely used in mining association rules. This technique has been used in various applications, such as finding fuzzy association rules from quantitative transactions by Hong et al. [30,32] and finding fuzzy association rules from both quantitative and categorical attributes by dividing them into various linguistic values by Hu et al. [35,36]. Besides these works, a number of studies in the past have exploited fuzzy techniques to mine fuzzy association rules or fuzzy sequential patterns from databases [8,9]. However, these above works focus on discovering patterns from certain data, rather than uncertain data.

Some researches have extended association rules mining techniques to imprecise or uncertain data [12,15,22,48,49]. Chen and Weng [12] propose a new approach to discover association rules from imprecise ordinal data. Although this work is interesting, the expressive power of this model is limited because it can not handle other types of uncertain data, such as categorical and numerical data. Djouadi et al. [22] propose a generalized possibilistic relational model and discover association rules under imprecision and vagueness. However, in this study only numerical data are represented by possibility distributions, categorical and ordinal data are not involved. Chui et al. [15] discover frequent itemsets from uncertain data under a probabilistic framework which associate items of transactions with existential probabilities. Unfortunately, this study only processes categorical data represented by existential probabilities without considering other data types, such as ordinal data.

Shyu et al. [49] apply the Dempster–Shafer evidential reasoning theory to generate the association rules from uncertain data under probabilistic framework. Dempster–Shafer Theory is a mathematical theory of evidence [47] which is an expansion of [20]. Dempster–Shafer theory offers an alternative to traditional probabilistic theory for the mathematical representation of uncertainty. In a finite discrete space, Dempster–Shafer theory can be interpreted as a generalization of probability theory, where probabilities are assigned to sets as opposed to mutually exclusive singletons. An important property of possibility theory is the ability to merge different data sources in order to increase the quality of the information. Besides, Dubois and Prade [24] argued that possibility theory is especially suitable in cases where human opinions, judgments and decisions are involved.

As pointed out in the above discussion, previous research has applied fuzzy sets to discover fuzzy association rules from certain data rather than uncertain data. Since the possibility theory is more suitable for human to express their opinions, judgments and decisions with uncertainty, this study attempts to combine these both theories and then discover fuzzy association rule from uncertain data. The possibility theory is closely related to fuzzy set theory [52]. In the next section, an overview of using possibility theory to represent uncertain data is presented.

3 Using possibility theory to represent uncertain data

The concept of a possibility measure was introduced by Zade [53]. A possibility measure Π on a universe U is a function from $\rho(U)$ to [0, 1], where $\rho(U)$ denotes the set of subsets of U. Now, let us consider the representation of incomplete or uncertain knowledge. Assume that John thinks that the best teaching method for the computer course would be BBI (blackboard instruction) and CAI (computer-aided instruction).

Consider the proposition $q : q \triangleq X$ is *the best teaching method*, where "*the best teaching method*" is a fuzzy subset defined in the universe of teaching methods as *the best teaching method* = 0.8/BBI+1/CAI, where 0.8/BBI signifies that the grade of membership of *the teaching method* BBI in the fuzzy set "*the best teaching method*" or equivalently, the compatibility of the statement that BBI is *the best teaching method* is 0.8. The interpretation of the proposition q can be as follows:

It is possible for any teaching method to be *the best teaching method*, with the possibility of X taking a value of u being equal to the grade of membership of u in the fuzzy set *the best teaching method*. In other words, q induces a possibility distribution Π_x which associates with each teaching method the possibility that u could be a value of X, equal to the grade of membership of u in the fuzzy set *the best teaching method*. Thus, possibility$\{X =$ BBI$\}$=0.8, possibility$\{X =$CAI $\}$= 1.

Generally, a possibility measureΠ can be built from a so-called possibility distribution π, which is a function from $\rho(U)$ to [0, 1], in the following way:

$$\forall X \in \rho(U), \quad \Pi(A) = \sup_{u \in A} \pi(u).$$

Note that $\Pi(\{u\}) = \pi(u)$. The distribution π can be viewed as a fuzzy restriction on the possible values of a variable X which takes its value on U. If X is a variable that takes values in U, and F is a fuzzy set subset of U characterized by a membership function m_F, then the proposition $q \triangleq X$ is F, induces a possibility distribution Π_x which is equal to F, i.e., $\Pi_x = F$, implying that possibility$\{X = u\} = m_F(u)$ for all $u \in U$.

From the above description, we know that possibility theory is an extension of the fuzzy set theory, which establishes a close connection between the concepts of similarity and uncertainty, and it provides an excellent framework for handling uncertain data. Besides, Dubois [23] noted that possibility theory is one of the current uncertain theories devoted to the handling of incomplete information. As a result, possibility theory has been used successfully in a wide range of engineering and scientific fields, including robotics application [43], decision making [21], instance-based learning [37], and querying systems in databases [41,45,48]. In order to discover rules from uncertain data, we need a brand new approach that can deal with raw data represented by possibility distributions. In the next section, we discuss the problem of mining fuzzy association rules from the three types of uncertain data represented by possibility distributions.

4 Problem definition

Djouadi et al. [22] define the semantics of *certainty association rules*, $A \Rightarrow B$ as "the more *certainly* X is A, the more *certainly* Y is B". In this study, we extend this concept and attempt to discover certain patterns from uncertain data represented by possibility distributions. Different from the work of Djouadi et al. [22] which determines certainty value (support)

Table 1 A data set with possibility distributions

TID	Itemsets
1	(TM, (0.2/BBI)+(1.0/CAI)))
	(ES, ((0.7/80)+(1.0/85)+(0.3/90)))
	(TS, ((0.1/average)+(1.0/good)))
2	(TM, (1.0/BBI))
	(ES, ((1.0/[71, 75])+(0.5/[76, 80])+(0.3/[81, 85])))
	(TS, ((0.2/poor)+(1.0/average)+(0.3/good)))
3	(TM, (0.3/BBI)+(1.0/CAI)))
	(ES, ((0.1/[76, 80])+(1.0/[81, 85])+(0.5/[86, 90])))
	(TS, ((0.2/average)+(1.0/good)))
4	(TM, (1.0/BBI) +(0.5/AVI)))
	(ES, ((0.4/85)+(1.0/90)))
	(TS, ((0.3/poor)+(1.0/average)+(0.2/good)))
5	(TM, (1.0/BBI) +(0.6/AVI)))
	(ES, ((0.1/[71, 75])+(1.0/[76, 80])+(0.5/[81, 85])))
	(TS, ((0.2/poor)+(1.0/average)+(0.1/good)))

TM teaching method, *ES* estimated score, *TS* teaching skill

by choosing either "membership degree" or "possibility degree", we use the mean and the standard deviation of those multiple values to represent its support and certainty. With these two measures, support and deviation, we can mine frequent itemsets with high supports and small deviations.

In this section, we define the problem of mining fuzzy association rules from uncertain data. First, several items used in the proposed algorithm are introduced. Second, we define the membership degree for each different item and use them to compute the itemsets' support from uncertain data.

Definition 1 Let $IT = \{it_1, it_2, \ldots, it_m\}$ be a set of all items. A d-item is denoted as $\left(it_i, \frac{p_{i1}}{x_{i1}} + \frac{p_{i2}}{x_{i2}} + \cdots + \frac{p_{ik}}{x_{ik}}\right)$, where $it_i \in IT$ is the item name and $\frac{p_{i1}}{x_{i1}} + \frac{p_{i2}}{x_{i2}} + \cdots + \frac{p_{ik}}{x_{ik}}$ is the possibility distribution of it_i. Here, each $x_{ik} \in X$ is a possible value of it_i and $p_{ik} \in [0, 1]$ stands for the possibility of x_{ik}.

Example 1 In Table 1, we show a data set containing five transactions. The values "blackboard instruction", "computer-aided instruction", and "audio-visual instruction" are abbreviated as BBI, CAI, and AVI, respectively. There are three items i_1, i_2, and i_3. The d-item could be category d-item, point number d-item, or interval number d-item. Take TID#1, for example. The d-item, TM, is a category d-item with value ((0.2/BBI)+(1.0/CAI)). The d-item, ES, is a point number d-item with value ((0.7/80)+(1.0/85)+(0.3/90)). The d-item, ES, in TID#2 is an interval number d-item with value ((1.0/[71, 75])+(0.5/[76, 80])+(0.3/[81, 85])). In this study, we are interested in examining the associations between the uncertain data represented by the possibility distributions and discover the fuzzy association rules, such as (TM, BBI)⇒ (ES, high).

Definition 2 An r-item can be a category r-item or a linguistic r-item. For simplicity, we use $b_i = (ic_i, f_i)$ to denote an r-item. An r-itemset B is a set of r-items, where all r-items must have distinct items' names. We use $B = \{(ic_1, f_1), (ic_2, f_2), \ldots, (ic_n, f_n)\}$ to denote an r-itemset.

Example 2 For example, {(TM, CAI), (ES, high)} is an r-itemset.

Definition 3 For category c_i and category c_j, let $\text{sim}^{CC}(c_i, c_j)$ denote the similarity between category c_i and category c_j. In this study, the *CC* similarity matrix Sim_{CC} stores the similarities between all categories.

Example 3 Table 2 is an example of a *CC* similarity matrix $\text{Sim}_{CC\text{-}T}$. From this matrix, we know that $\text{sim}^{CC\text{-}T}$(BBI, CAI)=0.50, $\text{sim}^{CC\text{-}T}$(BBI, AVI)=0.20, and $\text{sim}^{CC\text{-}T}$(CAI, AVI)=0.50. Table 3 is another example of a *CC* similarity matrix $Sim_{CC\text{-}L}$. From this matrix, we know that $\text{sim}^{CC\text{-}L}$(average, good)=0.50, $\text{sim}^{CC\text{-}L}$(good, good)=1, and $\text{sim}^{CC\text{-}L}$(good, very-good)=0.50.

Assume that we have a category d-item $a_i = \left(it_i, \frac{p_{i1}}{x_{i1}} + \frac{p_{i2}}{x_{i2}} + \cdots + \frac{p_{ik}}{x_{ik}}\right)$ and a category r-item $b_j = (ic_j, f_j)$. Let $\text{sim}_r(a_i, b_j)$ denote the similarity between x_{ir}, the rth-value of a_i, and b_j, where $1 \le r \le k$. Then, $sim_r(a_i, b_j)$ is given as follows:

$$\text{sim}_r(a_i, b_j) = \begin{cases} \text{sim}^{CC}(x_{ir}, f_j), & \text{if } it_i = ic_j \text{ and } x_{ir} = f_j \\ 0, & \text{otherwise} \end{cases}$$

There are some different ways to determine support (certainty) from uncertain data [22, 49]. Because the approach proposed by Djouadi et al. [22] is closely related to our study, we describe the difference between our work and this approach in determining support (certainty). Djouadi et al. [22] uses the maximum value between $(m_F(x))$ and $(1 - \pi_A(x))$, that is, $\text{Max}(m_F(x), 1 - \pi_A(x))$, to represent the item's certainty value (support). The drawback of this approach is that it determines certainty value (support) by choosing either "membership degree" or "possibility degree". Since both degrees are influential factors of determining the certainty value, a better definition should consider these two simultaneously. Therefore, this study uses the product of the item's possibility and the similarity to calculate the item's support.

Let $\text{sup}_r(a_i, b_j)$ denote the degree to which x_{ir}, the rth-value of a_i, matches b_j, where $1 \le r \le k$. There are two factors affecting $\text{sup}_r(a_i, b_j)$; one is how certain x_{ir} is and the other is how similar x_{ir} is to b_j. Therefore, $\text{sup}_r(a_i, b_j)$ can be defined as the product of x_{ir}'s

Table 2 A *CC* similarity matrix $\text{Sim}_{CC\text{-}T}$

	BBI	CAI	AVI
BBI	1	0.50	0.20
CAI	0.50	1	0.50
AVI	0.20	0.50	1

Table 3 A *CC* similarity matrix $\text{Sim}_{CC\text{-}L}$

	Very poor	Poor	Average	Good	Very good
Very poor	1	0.5	0	0	0
Poor	0.5	1	0.5	0	0
Average	0	0.5	1	0.5	0
Good	0	0	0.5	1	0.5
Very good	0	0	0	0.5	1

possibility and the similarity between x_{ir} and b_j. Furthermore, let $\sup(a_i, b_j)$ be the degree to which a_i matches b_j. Since there are totally k $\sup_r(a_i, b_j)$ values, we define $\sup(a_i, b_j)$ and $\text{dev}(a_i, b_j)$ as the average and standard deviation of these k values, respectively.

Definition 4 The following are the formal expressions for $\sup_r(a_i, b_j)$, $\sup(a_i, b_j)$, and $\text{dev}(a_i, b_j)$.

$$\sup_r(a_i, b_j) = p_{ir} \times \text{sim}_r(a_i, b_j), \quad \text{if } it_i = ic_j \text{ and } 1 \le r \le k$$

$$\sup(a_i, b_j) = \frac{\sum_{r=1}^{k} \sup_r(a_i, b_j)}{k}, \quad \text{if } it_i = ic_j$$

$$\text{dev}(a_i, b_j) = \sqrt{\frac{k\sum_{r=1}^{k} \sup_r(a_i, b_j)^2 - \left(\sum_{r=1}^{k} \sup_r(a_i, b_j)\right)^2}{k^2}}$$

Example 4 Suppose we have the *CC* similarity matrix $\text{Sim}_{CC\text{-}T}$ shown in Table 2, a category d-item $a_1 = (TM$, ((0.2/BBI)+(1.0/CAI))), and a category r-item $b_1 =$ (TM, CAI). Then, the degree $\sup(a_1, b_1) = (0.2 \times 0.5 + 1.0 \times 1.0)/2 = (0.1 + 1.0)/2 = 0.55$ and the degree $\text{dev}(a_1, b_1) = 0.45$, which are computed from values 0.1 and 1.0.

Definition 5 (*Fuzzification*) Suppose we have a universe of discourse Xin a *quantitative domain*, where each element x belongs to X. Then, a fuzzy set F is characterized by membership function $m_F(x)$, which maps x to a membership degree in interval [0, 1].

Example 5 Assume that we have five membership functions for the score: $S_{\text{very\_low}}$, S_{low}, S_{middle}, S_{high}, and $S_{\text{very\_high}}$. From these five membership functions, we know that $S_{\text{middle}}(73) = 0.7$, $S_{\text{high}}(81) = 0.9$, and $S_{\text{very\_high}}(92) = 1.0$.

$$S_{\text{very\_low}}(q) = \begin{cases} 1, & \text{if } q \le 50 \\ \frac{60-q}{60-50}, & \text{if } 50 \le q \le 60 \end{cases} \tag{1}$$

$$S_{\text{low}}(q) = \begin{cases} \frac{q-50}{60-50}, & \text{if } 50 \le q \le 60 \\ 1, & \text{if } q = 60 \\ \frac{70-q}{70-60}, & \text{if } 60 \le q \le 70 \end{cases} \tag{2}$$

$$S_{\text{middle}}(q) = \begin{cases} \frac{q-60}{70-60}, & \text{if } 60 \le q \le 70 \\ 1, & \text{if } q = 70 \\ \frac{80-q}{80-70}, & \text{if } 70 \le q \le 80 \end{cases} \tag{3}$$

$$S_{\text{high}}(q) = \begin{cases} \frac{q-70}{80-70}, & \text{if } 70 \le q \le 80 \\ 1, & \text{if } q = 80 \\ \frac{90-q}{90-80}, & \text{if } 80 \le q \le 90 \end{cases} \tag{4}$$

$$S_{\text{very\_high}}(q) = \begin{cases} \frac{q-80}{90-80}, & \text{if } 80 \le q \le 90 \\ 1, & \text{if } 90 \le q \end{cases} \tag{5}$$

Definition 6 Assume that we have a point number d-item $a_i = \left(it_i, \frac{p_{i1}}{x_{i1}} + \frac{p_{i2}}{x_{i2}} + \cdots + \frac{p_{ik}}{x_{ik}}\right)$, a linguistic r-item $b_j = (ic_j, f_j)$, and a membership function (FS_{f_j}), where $\text{FS}_{f_j}(x)$ denotes the membership degree to which x belongs to f_j. Let $\text{sim}_r(a_i, b_j)$ denote the similarity between x_{ir}, the rth-value of a_i, and b_j for $1 \le r \le k$. Then, $\text{sim}_r(a_i, b_j)$ is given as follows:

$$\text{sim}_r(a_i, b_j) = FS_{f_j}(x_{ir}), \quad \text{if } it_i = ic_j$$

Thus, $\sup(a_i, b_j)$ and $\text{dev}(a_i, b_j)$ can be given as follows:

$$\sup_r(a_i, b_j) = p_{ir} \times \text{sim}_r(a_i, b_j), \quad \text{if } it_i = ic_j \text{ and } 1 \le r \le k$$

$$\sup(a_i, b_j) = \frac{\sum_{r=1}^{k} \sup_r(a_i, b_j)}{k}, \quad \text{if } it_i = ic_j$$

$$\text{dev}(a_i, b_j) = \sqrt{\frac{k \sum_{r=1}^{k} \sup_r(a_i, b_j)^2 - \left(\sum_{r=1}^{k} \sup_r(a_i, b_j)\right)^2}{k^2}}$$

Example 6 Suppose we have a point number d-item a_2 =(ES, ((0.7/80) + (1.0/85) + (0.3/90))), a linguistic r-item b_2 =(ES, high), and a fuzzy membership function (S_{high}), as shown in Example 5. Since $S_{\text{high}}(90) = 0$, the support of (0.3/90) is 0 and thus eliminated in computing support and deviation. Then, degree $\sup(a_2, b_2) = (0.7 \times S_{\text{high}}(80) + 1.0 \times S_{\text{high}}(85))/2 = (0.7\times1.0+1.0\times0.5)/2 = (0.7+0.5)/2 = 0.6$ and degree $\text{dev}(a_2, b_2) = 0.1$.

Definition 7 Assume that we have an interval number d-item $a_i = (it_i, \frac{p_{i1}}{[x_{i1_a}, x_{i1_b}]} + \frac{p_{i1}}{[x_{i2_a}, x_{i2_b}]} + \cdots + \frac{p_{i1}}{[x_{ik_a}, x_{ik_b}]})$, a linguistic r-item $b_j = (ic_j, f_j)$, and a membership function (FS_{f_j}), where $\text{FS}_{f_j}(x)$ denotes the membership degree to which x belongs to f_j. Let $\text{sim}_r(a_i, b_j)$ denote the similarity between x_{ir}, the rth-value of a_i, and b_j for $1 \le r \le k$. Then, $\text{sim}_r(a_i, b_j)$ is given as follows:

$$\text{sim}_r(a_i, b_j) = \max\{\text{FS}_{f_j}(x_{ir})\}, \quad \text{if } x_{ir_a} \le x_{ir} \le x_{ir_b} \text{ and } it_i = ic_j$$

Furthermore, $\sup(a_i, b_j)$ and $\text{dev}(a_i, b_j)$ can be defined as shown in Definition 6.

Example 7 Suppose we have an interval number d-item a_2 =(ES, ((1.0/[71, 75]) + (0.5/[76, 80]) + (0.3/[81, 85]))), a linguistic r-item b_2 =(ES, high), and a fuzzy membership function (S_{high}), as shown in Example 5. Then, the degree $\sup(a_2, b_2) = (1.0 \times S_{high}(75) + 0.5 \times S_{high}(80) + 0.3 \times S_{high}(81))/3 = (1.0 \times 0.5 + 0.5 \times 1.0 + 0.3 \times 0.9)/3 = 0.42$ and the deviation $\text{dev}(a_2, b_2) = 0.11$.

There are several different inference and composition techniques. The most common and simplest method is the "min–max". Without losing generality, this paper uses the "min–max" method in this study. The Min and Max operators infer the supports and deviations of itemsets, respectively.

Definition 8 Assume that we have a d-itemset $A = \{(it_1, v_1), (it_2, v_2), \ldots, (it_m, v_m)\}$ and an r-itemset $B = \{(ic_1, f_1), (ic_2, f_2), \ldots, (ic_n, f_n)\} (n \le m)$. Assume we can find $i_1, i_2, \ldots, i_n$ such that a_{i_j} matches b_j, for $1 \le j \le n$. Let $\sup(A, B)$ denote the degree to which A contains B. Then, $\sup(A, B)$ can be defined as follows:

$$\sup(A, B) = \text{Min}_{j=1}^{n} \sup(a_{i_j}, b_j)$$

Additionally, $\text{dev}(A, B)$ can be defined as follows:

$$\text{dev}(A, B) = \text{Max}_{j=1}^{n} \text{dev}(a_{i_j}, b_j)$$

Example 8 Suppose we have a d-itemset A={(TM, ((0.2/BBI)+(1.0/CAI))), (ES, ((0.7/80) + (1.0/85) + (0.3/90)))}, an r-itemset B ={(*TM*, CAI), (ES, high)}, and a fuzzy membership function (S_{high}), as shown in Example 5. Then, the degree $\sup(A, B) = \min(0.55, 0.6) = 0.55$ and the degree $\text{dev}(A, B) = \max(0.45, 0.10) = 0.45$.

Definition 9 Assume that we have a database DB formed from a set of transactions. Let d-itemset A_i be the i-th transaction in DB, denoted as $A_i = \{(it_1, v_1), (it_2, v_2), \ldots, (it_m, v_m)\}$, where $a_i = (it_i, v_i)$ could be a category d-item, point number d-item, or interval number d-item. Suppose we have an r-itemset $B = \{(ic_1, f_1), (ic_2, f_2), \ldots, (ic_n, f_n)\}(n \leq m)$, where $b_j = (ic_j, f_j)$ is an r-item. Then, the support of Bin database DB, denoted as $\text{sup}_{\text{DB}}(B)$, can be defined as follows:

$$\text{sup}_{\text{DB}}(B) = \left(\sum\nolimits_{\text{sid} \subseteq \text{DB}} \text{sup}(A_{\text{sid}}, B)\right) / |\text{DB}|,$$

where |DB| denotes the number of transactions in database DB.

The deviation of B in database DB, denoted as $\text{dev}_{\text{DB}}(B)$, can be defined as follows:

$$\text{dev}_{\text{DB}}(B) = \left(\sum\nolimits_{\text{sid} \subseteq DB_B} \text{dev}(A_{\text{sid}}, B)\right) / (|\text{DB}_B|),$$

where $|\text{DB}_B|$ is the subset of transactions in the DB containing itemset B.

Note that we use $|\text{DB}_B|$ as the denominator when computing $\text{dev}_{\text{DB}}(B)$ in Definition 9. This is because $\text{dev}(A_{\text{sid}}, B)$ is meaningless if transaction A_{sid} does not contain r-itemset B. Since $\text{dev}(A_{\text{sid}}, B)$ can only be applied to these $|\text{DB}_B|$ transactions, we use $|\text{DB}_B|$ as the denominator.

Example 9 Assume that we are given a database DB, as shown in Table 1, and an r-itemset B = ((TM, CAI), (ES, high)). Then, the degree $\text{sup}_{\text{DB}}(B) = (\min(0.55, 0.60) + \min(0.50, 0.42) + \min(0.57, 0.40) + \min(0.38, 0.20) + \min(0.40, 0.50))/5 = (0.55 + 0.42 + 0.40 + 0.20 + 0.40)/5 = 0.39$ and deviation $\text{dev}_{\text{DB}}(B) = (0.45 + 0.11 + 0.43 + 0.12 + 0.39)/5 = 0.30$. The details of this computation are shown in Table 4.

Definition 10 Given two user-specified thresholds σ_{sup} and σ_{dev}, an r-itemset B is frequent if $\text{sup}_{\text{DB}}(B)$ is no less than σ_{sup}; it is frequent-and-certain (or FC) if it is frequent and $\text{dev}_{\text{DB}}(B)$ is no larger than σ_{dev}. Let B be a FC r-itemset, where $B = X \cup Y$ and $X \cap Y = \phi$. Then, the confidence of the rule $X \Rightarrow Y$, denoted as $\text{conf}(X \Rightarrow Y)$, is defined as $\text{sup}_{\text{DB}}(B)/\text{sup}_{\text{DB}}(X)$. Given a confidence threshold σ_{conf}, if $\text{conf}(X \Rightarrow Y) \geq \sigma_{\text{conf}}$, then $X \Rightarrow Y$ holds in the database D.

Example 10 Suppose we have a database DB, as shown in Table 1, and three user-specified thresholds $\sigma_{\text{sup}} = 0.35$, $\sigma_{\text{dev}} = 0.3$, and $\sigma_{\text{conf}} = 60\%$. Let r-itemset X= (TM, CAI), Y= (ES, high), and $B = X \cup Y$. Then, from Table 4, we have $\text{sup}_{\text{DB}}(B) = 0.39$, $\text{sup}_{\text{DB}}(X) = 0.48$,

Table 4 The supports and deviations of the itemsets

TID	sup_{DB}			dev_{DB}		
	TM.CAI	ES.high	TM.CAI ∪ ES.high	TM.CAI	ES.high	TM.CAI ∪ ES.high
1	0.55	0.60	0.55	0.45	0.10	0.45
2	0.50	0.42	0.42	0.00	0.11	0.11
3	0.57	0.40	0.40	0.43	0.36	0.43
4	0.38	0.20	0.20	0.12	0.00	0.12
5	0.40	0.50	0.40	0.10	0.39	0.39
AVG	0.48	0.42	0.39	0.22	0.19	0.30

Input: A database, D_B; membership functions (FS_{f_j}); a predefined minimum support σ_{sup}; a predefined maximum deviation σ_{dev}.; a predefined minimum confidence λ.
Output: A set of fuzzy association rules
Method:
// Phase 1 Call the *Sup_Transform* Subroutine
(1). For each transcation
 Transform each d-item data into r-items;
 Store these results as a new transaction in new database D^T.
// Phase 2 Call the *FreqCertItemsets_gen* Subroutine
(1). For each r-item $ic_{j,r}$, calculate its support.
(2). Check whether the support of each r-item $ic_{j,r}$ is no less than the minimum support σ_{sup}. If it is, put it into the set of frequent one-itemsets (L_1).
(3). Check whether the deviation of each r-itemset in L_1 is no larger than the maximum deviation σ_{dev}. If it is, put it into the set of FC one-itemsets (L_1^c).
(4). Generate candidate set C_{k+1} from L_k.
(5). Compute the supports of all r-itemsets in C_{k+1} and determine L_{k+1}.
(6). Compute the deviations of all r-itemsets in L_{k+1} and determine L_{k+1}^C.
(7). If L_{k+1} is null, go to phase 3; otherwise, set $k = k + 1$ and repeat steps (4)–(6).
// Phase 3 Call the *FAR_gen* Subroutine
(1). Generate fuzzy association rules from all FC r-itemsets.

Fig. 1 The UDM algorithm

$\text{dev}_{\text{DB}}(B) = 0.30$, and $\text{dev}_{\text{DB}}(X) = 0.22$. Therefore, r-itemsets B and X are both FC. Furthermore, the confidence $\text{conf}(X \Rightarrow Y) = 0.39/0.48 = 81.25\%$ is greater than σ_{conf}. So, the rule (TM.CAI ⇒ ES.high) holds. The meaning of this rule is that if teaching method is CAI then the expected score is high.

5 Algorithm for mining fuzzy association rules from uncertain data

In this section, we introduce an Apriori-like algorithm, named the UDM (uncertain data mining) algorithm, to discover fuzzy association rules from uncertain data. The UDM algorithm was developed by modifying the well-known Apriori algorithm [2] to mine fuzzy rules from uncertain data. In Sect. 5.1, we introduce the UDM algorithm; an example is shown in Sect. 5.2.

5.1 The proposed algorithm

We now introduce a new algorithm for mining fuzzy association rules from uncertain data. The algorithm is outlined in Fig. 1. Although the basic structure of UDM is similar to the Apriori algorithm, they differ in the following respects:

(1) Data types: The Apriori algorithm was designed only for handling categorical data. The UDM algorithm, however, was developed for handling the three data types that may appear in uncertain data.
(2) Membership functions: In the Apriori algorithm, an item can only 100 or 0% match with another item. Therefore, the Apriori algorithm does not need a membership function to measure the membership degree between items. Since partial membership relations exist in raw data, the UDM algorithm uses the membership functions described in Sect. 4 to calculate the membership degree between items.

(3) Possibility: In the Apriori algorithm, an itemset is completely certain. In the UDM algorithm, an itemset can be partially certain. Therefore, this study uses the product of the value's possibility and similarity as the item's support.
(4) Deviation: In the Apriori algorithm, an itemset is completely certain. In the UDM algorithm, an itemset can hold multiple supports. Therefore, this study uses an itemset's deviation, which is calculated by these multiple supports to measure the itemset's uncertainty.
(5) Counting candidates: In the Apriori algorithm, an itemset is either completely contained in a transaction or not at all. In the UDM algorithm, an itemset can be partially contained in a transaction. As a result, the degree to which a transaction contains an itemset is a value between 0 and 1, instead of either 0 or 1. The UDM algorithm also uses deviation to determine whether an itemset is certain or not. Accordingly, the algorithm seeks patterns that are frequent and certain.

Besides the differences mentioned above, there is one more distinction that makes the structure of our algorithm different from the Apriori algorithm.

Definition 11 (*Anti-monotone constraints*) A constraint ζ is anti-monotone if it holds for an itemset S, than it holds for any subset of S.

Lemma 1 *Let the constraint that* $\sup_{DB}(B)$ *must be not less than* $\sigma_{\sup}$ *and that* $\mathrm{dev}_{DB}(B)$ *must be no larger than* σ_{dev} *be called the support constraint and the deviation constraint, respectively. Then the support constraint satisfies the anti-monotone property, while the deviation constraint does not.*

Proof It is obvious that the support constraint satisfies the anti-monotone property. Here, we use a counter example to show why the deviation constraint does not satisfy the anti-monotone property. Assume that we have a data set composed of two transactions, as shown in Table 5. The deviation of item X is $(0.3 + 0.4)/2 = 0.35$, but the deviation of itemset XY is 0.3. If $\sigma_{\mathrm{dev}} = 0.33$, then XY satisfies the deviation constraint but its subset X does not. □

Lemma 1 indicates that we can use the support constraint to develop the algorithm, similar to the traditional Apriori algorithm. Since the deviation constraint does not satisfy the anti-monotone property, after finding the frequent r-itemsets in each phase, we must further apply the deviation constraint to prune the uncertain frequent r-itemsets.

The proposed algorithm is composed of three phases, as shown in Fig. 1. In the first phase, we apply the membership functions as given in Sect. 4 to transform the original database into a new database. After the transformation, a transaction in the new database stores the support and deviation of every r-item in the corresponding transaction of the original database. In the second phase, we use a level-wise approach to iteratively generate candidate r-itemsets of k items, C_k, and then find frequent r-itemsets of k items, L_k. We then use the maximum deviation σ_{dev} to determine FC r-itemsets of k items, L_k^c. To proceed to the next level, we generate candidate set C_{k+1} from L_k and repeat. In the final phase, we generate fuzzy association rules from the FC r-itemsets, L_k^c, obtained in the second phase.

Table 5 The deviations of the items

TID	X	Y	Z
1	0.3	0.3	0.1
2	0.4		0.3

Subroutine: *Sup_Transform* Subroutine. Transform every d-item data (it_i, v_i) in each transaction of D into a set of ($ic_{j,r}$, $\mu_{j,r}$, $d_{j,r}$), where $ic_{j,r}$ is an r-item, $\mu_{j,r}$, is its support, and $d_{j,r}$ is its deviation.
Input: a database, D; a membership function, FS_{f_j}.
Output: D^T, a new database where each transaction contains a set of (b_j, μ_j, d_j).
(1) for each transaction $t \in D$ {
(2) for each d-item a_i=(it_i, v_i){
// the following procedure does not generate an r-item if its support is zero//
(3) if $a_i \in$ *category* d-item, then for each category f_j
create an r-item b_j= (it_i, f_j) with *sup* and *dev* as defined in Definition 4.
(4) if $a_i \in$ *point number* d-item, then for each linguistic term f_j
create an r-item b_j= (it_i, f_j) with *sup* and *dev* as defined in Definition 6.
(5) if $a_i \in$ *interval number* d-item, then for each linguistic term f_j
create an r-item b_j= (it_i, f_j) with *sup* and *dev* as defined in Definition 7.}
(6) store the results as a transaction in the new database D^T;}
(7) return D^T;

Fig. 2 The *Sup_Transform* function

Note that, to improve the performance of the proposed approach, two subroutines *Sup_Transform* and *FreqCertItemsets_gen* should be integrated into one single subroutine, that is, calculate each itemsets' support and deviation directly without using temporary database D^T to avoid the copy of the whole database. However, to clarify the two different datasets used in the traditional and the proposed approach, the *Sup_Transform* subroutine is separated from the *FreqCertItemsets_gen* subroutine in this study. In the following, we present and explain in detail the three subroutines in each phase.

As mentioned previously, there are three types of uncertain data. Accordingly, we need to use the membership functions to calculate the supports for each data type. Figure 2 shows the pseudocode for the *Sup_Transform* subroutine, which is used to calculate the support for each data type. In steps 1–7, we check the data type of every d-item a_i in each transaction, and generate the corresponding r-items. Finally, every d-item a_i is transformed into r-items in the form (b_j, μ_j, d_j), where b_j is an r-item, μ_j is its support, and d_j is its deviation. Steps 3–5 calculate the support and deviation for different data types, such as *categories, point numbers*, and *interval numbers*. With the help of membership functions, the *Sup_Transform* subroutine can transform the original data set into a new form D^T. followings, we introduce the *FreqCertItemsets_gen* subroutine, which mines fuzzy association rules from database D^T.

The Apriori algorithm calculates the counts of candidate itemsets by adding either a one or a zero, depending on whether that particular itemset appears in the transaction or not. The *FreqCertItemsets_gen* subroutine in the proposed algorithm, however, can add a fractional value to the counts of itemsets. Figure 3 shows the pseudocode for the *FreqCertItemsets_gen* subroutine. Step 1 finds the frequent 1-itemsets, L_1. In steps 2–10, L_{k-1} is used to generate candidates C_k in order to find L_k. The *apriori_gen* subroutine [1,2] generates the candidates and then uses the downward closure property to eliminate those that have a non-frequent subset (step 3). Once all the candidates have been generated, the database is scanned (step 4). For each transaction, a *subset* function is used to find all subsets of the transaction that are candidates (step 5), and the counts and deviations for each of those candidates are accumulated (steps 6, 7, and 8). Finally, all candidates satisfying the minimum support constraint constitute the set of frequent r-itemsets (L_k). After filtering out uncertain r-itemsets from L_k, we obtain FC itemsets (L_k^c) (steps 13, 14). We can then use those patterns to generate the

Subroutine: *FreqCertItemsets_gen* Subroutine. Find FC r-itemsets using an iterative, level-wise approach based on candidate generation.
Input: Database D^T; the minimum support σ_{sup} and the maximum deviation σ_{dev}.
Output: L^c, FC r-itemsets in D^T.
Method:
(1) L_1 = find_frequent_1-itemsets(D^T);
(2) for (k=2; $L_{k-1}\neq\varnothing$; k++) {
(3) C_k = apriori_gen(L_{k-1}, *min_sup*, *max_dev*);
(4) for each transaction $t\in D^T$ { //Scan D to compute counts
(5) C^t=subset(C_k, t); //get the subsets of t that are candidates;
(6) for each candidate $c \in C^t$ {
(7) $c.count{=}c.count + Min_{j=1}^{k} sup_j$;
(8) $c.devcount = c.devcount + Max_{j=1}^{k} dev_j$;
(9) }
(10) }
(11) $c.sup= (c.count / |DB|)$;
//where $|DB|$ denotes the transaction numbers of database DB.
(12) $c.dev= (c.devcount / |DB_B|)$;
//where $|DB_B|$ denotes the transaction numbers of the itemsets B in database DB.
(13) $L_k = \{c \in C_t \mid c.sup \geq \sigma_{sup}\}$;
(14) $L_k^c = \{c \in L_k \mid c.dev \leq \sigma_{dev}\}$;
(15) return $L{=}\cup_k L_k$};

Fig. 3 The *FreqCertItemsets_gen* function

Subroutine: *FAR_gen* Subroutine. Generate all fuzzy association rules
Input: L^c, FC itemsets in D^T; a predefined minimum confidence λ.
Output: *FAR*, Fuzzy association rules in D^T.
Method:
(1) For each FC itemset $B{=}X\cup Y$, where $X\cap Y{=}\phi$
(2) For every subset X of B
(3) If the confidence of rule $X\Rightarrow Y$ is no less than the minimum confidence λ
(4) then output the rule
(5) return *FAR*

Fig. 4 The *FAR_gen* function

Fuzzy association rules (FAR). In the next section, we introduce the *FAR_gen* subroutine, which generates fuzzy association rules from FC r-itemsets L_k^c.

Finally, like the Apriori algorithm, we generate fuzzy association rules from the FC r-itemsets (L_k^c) obtained in the second phase. Figure 4 shows the pseudocode for the *FAR_gen* subroutine. Obviously, the procedure can generate all the FARs, satisfying Definition 10.

5.2 An example

An example is given to illustrate the proposed data mining algorithm. This example uses the data set shown in Table 1.

STEP 1 Assume that we have five membership functions (FS_{f_j}), as shown in expressions (1)–(5), and two similarity matrixes, as shown in Tables 2 and 3. Every d-item $\left(it_i, \frac{p_{i1}}{x_{i1}} + \frac{p_{i2}}{x_{i2}} + \cdots + \frac{p_{ik}}{x_{ik}}\right)$ in each transaction of D can be mapped to a set of r-items with multiple supports, and the results are shown in Table 6. Furthermore, by consolidating the multiple

Table 6 The itemsets with multiple supports

TID	Teaching method (TM)	Estimated score (ES)	Teaching skill (TS)
1	(BBI, (0.20, 0.50))	(high, (0.70, 0.50))	(poor, (0.50))
	(CAI, (0.10, 1.00))	(very-high, (0.50, 0.30))	(average, (0.10, 0.50))
	(AVI, (0.04, 0.50))		(good, (0.05, 1.00))
			(very-good, (0.50))
2	(BBI, (1.00))	(middle, (0.90, 0.20))	(very-poor, (0.10))
	(CAI, (0.50))	(high, (0.50, 0.50, 0.27))	(poor, (0.20, 0.50))
	(AVI, (0.20))	(very-high, (0.15))	(average, (0.10, 1.00, 0.15))
			(good, (0.50, 0.30))
			(very-good, (0.15))
3	(BBI, (0.30, 0.50))	(middle, (0.40))	(poor, (0.10))
	(CAI, (0.15, 1.00))	(high, (0.10, 0.90, 0.20))	(average, (0.20, 0.50))
	(AVI, (0.06, 0.50))	(very-high, (0.50, 0.50))	(good, (0.10, 1.00))
			(very-good, (0.50))
4	(BBI, (1.00, 0.10))	(high, (0.20))	(very-poor, (0.15))
	(CAI, (0.50, 0.25))	(very-high, (0.20, 1.00))	(poor, (0.30, 0.50))
	(AVI, (0.20, 0.50))		(average, (0.15, 1.00, 0.10))
			(good, (0.50, 0.20))
			(very-good, (0.10))
5	(BBI, (1.00, 0.10))	(middle, (0.09, 0.40))	(very-poor, (0.10))
	(CAI, (0.50, 0.25))	(high, (0.05, 1.00, 0.45))	(poor, (0.20, 0.50))
	(AVI, (0.40, 0.20))	(very-high, (0.25))	(average, (0.10, 1.00, 0.05))
			(good, (0.50, 0.10))
			(very-good, (0.05))

supports of every r-item into two values, support and deviation, we can build the temporary database D^T shown in Table 7.

STEP 2.1 For each r-item stored in database D^T, calculate its support and check whether the support of each r-item is greater than or equal to the minimum support σ_{sup}. If it is, then insert it in L_1.

Then, calculate its deviation and check whether the deviation of each frequent itemset in L_1 is less than or equal to the maximum deviation σ_{dev}. If it is, put it into the set of L_1^c. For example, let us set σ_{sup} to 0.38 and σ_{dev} to 0.35. Then, we have L_1 and L_1^c, as shown in Table 8.

STEP 2.2 Now, we generate candidate set C_2 from L_1. For example, we can obtain C_2 as follows: (TM.BBI, TM.CAI), (TM.BBI, ES.high), (TM.BBI, ES.very-high), (TM.BBI, TS.good), …, and (ES.very-high, TS.good). After computing their supports, we can determine L_2, as shown in Table 9. Then, filter out the uncertain itemsets from L_2 to obtain the FC itemsets (L_2^c), as shown in Table 10.

STEP 2.3 Since L_2 is not null, we repeat the previous steps to find L_3. Unfortunately, we find that C_3 is empty after pruning; therefore, we stop the iterations.

Table 7 The constructed temporary set D^T

TID	Teaching method (TM)	Estimated score (ES)	Teaching skill (TS)
1	(BBI, 0.35, 0.15)	(high, 0.60, 0.10)	(poor, 0.05, 0.00)
	(CAI, 0.55, 0.45)	(very-high, 0.40, 0.10)	(average, 0.30, 0.20)
	(AVI, 0.27, 0.23)		(good, 0.52, 0.48)
			(very-good, 0.50, 0.00)
2	(BBI, 1.00, 0.00)	(middle, 0.55, 0.35)	(very-poor, 0.10, 0.00)
	(CAI, 0.50, 0.00)	(high, 0.42, 0.11)	(poor, 0.35, 0.15)
	(AVI, 0.20, 0.00)	(very-high, 0.15, 0.00)	(average, 0.42, 0.41)
			(good, 0.40, 0.10)
			(very-good, 0.15, 0.00)
3	(BBI, 0.40, 0.10)	(middle, 0.04, 0.00)	(poor, 0.10, 0.00)
	(CAI, 0.57, 0.43)	(high, 0.40, 0.36)	(average, 0.35, 0.15)
	(AVI, 0.28, 0.22)	(very-high, 0.50, 0.00)	(good, 0.55, 0.45)
			(very-good, 0.50, 0.00)
4	(BBI, 0.55, 0.45)	(high, 0.20, 0.00)	(very-poor, 0.15, 0.00)
	(CAI, 0.38, 0.12)	(very-high, 0.60, 0.40)	(poor, 0.40, 0.10)
	(AVI, 0.35, 0.15)		(average, 0.42, 0.41)
			(good, 0.35, 0.15)
			(very-good, 0.10, 0.00)
5	(BBI, 0.56, 0.44)	(middle, 0.25, 0.16)	(very-poor, 0.10, 0.00)
	(CAI, 0.40, 0.10)	(high, 0.50, 0.39)	(poor, 0.35, 0.15)
	(AVI, 0.40, 0.20)	(very-high, 0.25, 0.00)	(average, 0.38, 0.44)
			(good, 0.30, 0.20)
			(very-good, 0.05, 0.00)

Table 8 L_1 andL_1^c

Frequent itemsets in L_1			FC itemsets in L_1^c		
Itemsets	Support	Deviation	Itemsets	Support	Deviation
TM.BBI	0.572	0.228	TM.BBI	0.572	0.228
TM.CAI	0.480	0.220	TM.CAI	0.480	0.220
ES.high	0.424	0.192	ES.high	0.424	0.192
ES.very-high	0.380	0.100	ES.very-high	0.380	0.100
TS.good	0.424	0.276	TS.good	0.424	0.276

Table 9 L_2 for this example

Itemsets	Support	Deviation
(TM.BBI, TM.CAI)	0.406	0.354
(TM.CAI, ES.high)	0.394	0.300
(TM.CAI, TS.good)	0.424	0.276

Table 10 L_2^c for this example

Itemsets	Support	Deviation
(TM.CAI, ES.high)	0.394	0.300
(TM.CAI, TS.good)	0.424	0.276

Table 11 The generated fuzzy association rules

No.	Rule
1	If TM.CAI, then ES.high; (confidence = 82.08%)
2	If ES.high, then TM.CAI; (confidence = 92.92%)
3	If TM.CAI, then TS.good; (confidence = 88.33%)
4	If TS.good, then TM.CAI; (confidence = 100%)

STEP 3 Construct the FARs from all frequent and certain r-itemsets.

We can generate the FARs from L_2^c. For example, let us set λ to 0.8. Then, we can obtain the FARs, as shown in Table 11.

6 Experiment results

We conducted several experiments to evaluate our approach and the performance of the proposed algorithm. The survey data concerned teaching evaluations of courses at a university in Taiwan. We invite about 600 students to answer questions. For the students' convenience, we list the questionnaire's contents on the web and then record their opinions into database. A total of 490 pieces of valid survey data were collected after eliminating undeliverable ones. The questionnaire's contents can be found in Appendix A. The data set contains a description of each student's opinions about the courses they took in the semester. It combines information from two categories: (1) course, including the course's pace, expected score, (2) lecturer, including lecturer's professional, lecturer's guidance and support, pace of lecturing. Each transaction provided information about the teacher's teaching performance and the student's learning performance. The algorithms were implemented using Sun Java language (J2SDK 1.3.1) and tested on a PC with a single Intel Pentium III 866 MHz processor and 512MB main memory using the Windows XP operating system. Neither multi-threading technology nor parallel computing skills were used in our implemented programs.

There are two approaches for the acquisition of membership functions: (1) querying experts [30,32]; (2) applying machine learning techniques [3,4,29,31]. Most existing papers [30,32] in fuzzy data mining only focus on the design of mining algorithms by assuming that the membership functions are given, because this can streamline the presentation of the paper and enable us to focus on the design of mining algorithms. Due to the same reasons, we assume that the fuzzy functions and matrixes are given by experts. For these experiments, we invited a senior faculty department member to set the values of the five membership functions, matrix $\mathrm{Sim}_{CC\text{-}T}$, and matrix $\mathrm{Sim}_{CC\text{-}L}$.

We performed three experiments. In the first experiment, we investigate how the run time of the UDM algorithm changes as we vary the minimum support value and the database size. In the second experiment, we compare the performances of the proposed UDM algorithm and the traditional fuzzy-Apriori algorithm, which uses the *Min* operator to infer the supports of itemsets [30]. Finally, the third experiment applies the UDM algorithm to discover

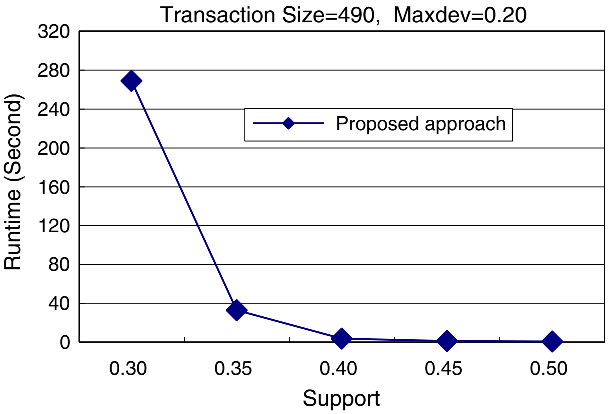

Fig. 5 Run time versus minimum support (proposed approach)

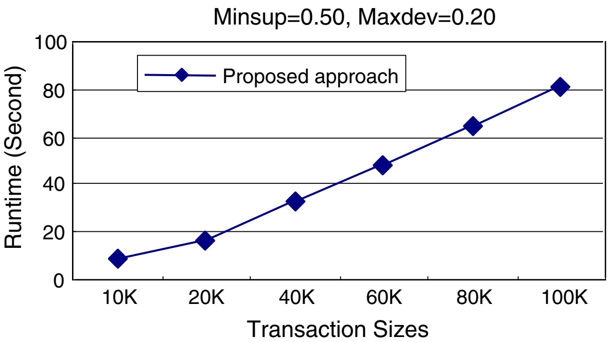

Fig. 6 Run time versus database size (proposed approach)

rules from uncertain data in the real world. The traditional fuzzy-Apriori algorithm discovers both low and high certainty itemsets, however, our algorithm discovers interesting rules only containing high certainty itemsets.

In the first experiment, we are interested in investigating how the run time changes as we vary the minimum support value and the database size. Therefore, we first fixed the database size at 490, maximum deviation at 0.2, and varied the minimum support. In Fig. 5, it is apparent that the run time increases as the minimum support value decreases. This is especially true when the minimum support becomes very small; the run time increases sharply. These results concur with results from previous association mining algorithms [28,46,51]. Next, we set the minimum support at 0.5, maximum deviation at 0.2, and varied the number of transactions by repeatedly duplicating the database until the intended size was reached. From Fig. 6, we found that the run time increases linearly with respect to database size. This linear relationship indicates that the proposed algorithm has a good scalability.

In the second experiment, we focus on studying performance differences between the traditional fuzzy-Apriori algorithm [30] and the proposed UDM algorithm. Note that the traditional fuzzy-Apriori algorithm does not consider the deviation factor, while ours does. Therefore, to make comparison possible, we needed to generate a specific transaction set for the traditional fuzzy-Apriori algorithm. Since the *Sup_Transform* function used in this work can generate a new database, D^T, where each transaction contains a set of (b_j, μ_j, d_j), we

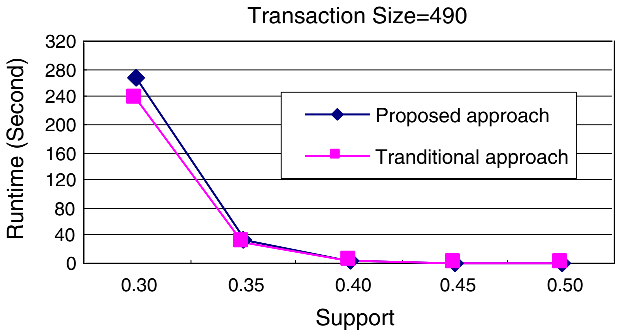

Fig. 7 Run time versus minimum support (comparison)

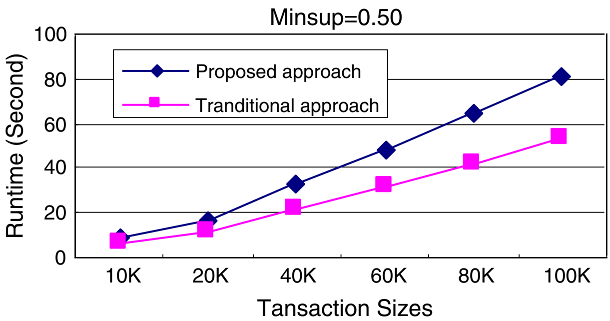

Fig. 8 Run time versus database size (comparison)

eliminated the deviation part, (d_j), from each transaction set of (b_j, μ_j, d_j). We obtained a new transaction set of (b_j, μ_j), which the traditional fuzzy-Apriori algorithm could use directly. Then, we could compare the performances of the traditional fuzzy-Apriori algorithm and the proposed UDM algorithm.

In the first test, we set the database size at 490 and varied the minimum support from 0.30 to 0.50. From the results in Fig. 7, we found that the traditional fuzzy-Apriori algorithm performs slightly better than the proposed algorithm. This result is quite reasonable because the proposed UDM algorithm deals with an extra factor, *deviation*, which requires additional computation.

In the second test, we set the minimum support at 0.5 and varied the number of transactions by repeatedly duplicating the database until the intended size was reached. Figure 8 indicates that the fuzzy-Apriori algorithm has a better run time than the UDM algorithm. The reason for this result is the same as the first test, the additional factor of deviation.

In the third test, we set the database size at 490 and varied the minimum support from 0.30 to 0.50. Figures 9 and 10 show the accumulative numbers of patterns in L_1, L_2, and L_3 for different minimum supports σ_{sup}. Obviously, the traditional fuzzy-Apriori algorithm generates more patterns in L_1, L_2, and L_3 than the UDM algorithm. The results, however, indicate that many patterns generated by the traditional fuzzy-Aprori algorithm are of high uncertainty. Using the new index, deviation, our proposed approach discovers only believable patterns from uncertain data.

The third experiment investigated the number of patterns for different deviation thresholds. In this experiment, we set the database size at 490, the minimum support thresholds

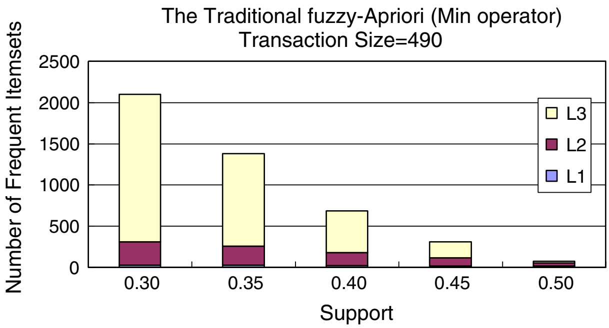

Fig. 9 Patterns versus minimum support (traditional approach)

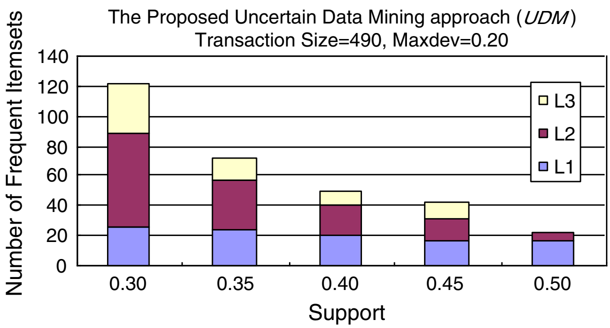

Fig. 10 Patterns versus minimum support (proposed approach)

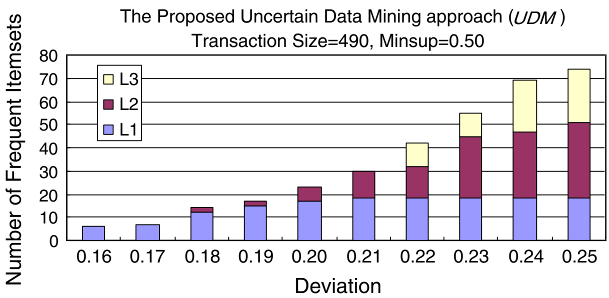

Fig. 11 Patterns versus maximum deviation (proposed approach)

σ_{sup} to 0.50, and varied the maximum deviation from 0.16 to 0.25. Figure 11 shows the accumulative numbers of patterns in L_1, L_2, and L_3 for different maximum deviation thresholds σ_{dev}. The number of patterns increases with the maximum deviation value. This is especially true when the maximum deviation becomes very large. Obviously, with the new deviation index proposed in this work, we can eliminate many highly uncertain patterns and obtain more certain ones from uncertain data. From the above results, our algorithm has been proven through its discovery of interesting rules that contain only high certainty itemsets.

With support value $\sigma_{\text{sup}} = 0.5$ and deviation value $\sigma_{\text{dev}} = 0.2$, the proposed UDM approach discovered L_2^c, as shown in Table 12. One may wonder if there were any more patterns generated by the traditional fuzzy-Apriori approach. Actually, with support values

Table 12 L_2^c in this experiment

Itemsets	Support (%)	Deviation (%)
(PC.good, PL.good)	55.03	19.50
(PC.good, TS.good)	54.30	19.40
(PL.good, TS.good)	53.79	19.60
(GS.good, TS.good)	54.42	19.66
(GS.very-good, LP.very-good)	52.58	17.02
(LP.very-good, TS.very-good)	50.23	17.00

GS guidance and support, *LP* lecturer professional, *PC* pace of the course, *PL* pace of the lectures, *TS* teaching skill.

$\sigma_{\text{sup}} = 0.5$, the traditional fuzzy-Apriori approach generated an extra 27 patterns with a deviation larger than 0.20, such as (PC.good, GS.good).

Since the proposed approach uses deviation thresholds to eliminate highly uncertain patterns, the uncertain L_2 itemset (PC.good, GS.good) could only be discovered by the traditional fuzzy-Apriori approach. From itemset (PC.good, GS.good), we can generate a rule like "if the pace of the course is good, the guidance and support provide by the lecture will be good". Although its support value is larger than the minimum support thresholds σ_{sup}, this rule seems not as reasonable as the rule "if the pace of the course is good, the pace of the lecturer will be good", which is generated by the certain L_2^c itemsets, (PC.good, PL.good), as shown in Table 12. From the above illustration, we know that without the deviation threshold the traditional fuzzy-Apriori approach generates many uncertain rules, whereas the proposed one will not. Therefore, the proposed approach can improve the quality and the accuracy of extracted knowledge. Finally, Table 13 shows some certain rules that can be found by using our approach but not with the traditional method.

Table 13 Association rules generated by L_2^c in this experiment

No	Rules	Support (%)	Confidence (%)
1	(PC.good ⇒ PL.good)	55.03	95.28
2	(PL.good ⇒ PC.good)	55.03	96.45
3	(PC.good ⇒ TS.good)	54.30	94.01
4	(TS.good ⇒ PC.good)	54.30	95.19
5	(PL.good ⇒ TS.good)	53.79	94.27
6	(TS.good ⇒ PL.good)	53.79	94.29
7	(GS.good ⇒ TS.good)	54.42	92.08
8	(TS.good ⇒ GS.good)	54.42	95.41
9	(GS.very-good ⇒ LP.very-good)	94.92	94.92
10	(LP.very-good ⇒ GS.very-good)	94.92	88.60
11	(LP.very-good ⇒ TS.very-good)	84.64	84.64
12	(TS.very-good ⇒ LP.very-good)	84.64	95.98

GS guidance and support, *LP* lecturer professional, *PC* pace of the course, *PL* pace of the lectures, *TS* teaching skill

Please refer to the certain rules generated by L_2^c in Table 13, which were found using our approach. Those rules are meaningful for teaching evaluation and course evaluation. Take Rule#4 and Rule#8 as examples. Rule #4 indicates that if the lecturers own good teaching skills, the pace of the course will be more suitable for students' learning. Rule #8 indicates that if the lecturers own good teaching skills, students will feel comfortable in seeking guidance and support from the lecturers. Our finding suggests that the lecturers with good teaching skills not only make the pace of the course more suitable for students' learning but also make students more considerate in seeking guidance and support.

7 Conclusion

Association rule mining is an extremely popular data mining technique that can discover relationships between data. Association rule mining algorithms have been used in various applications and data sets, due to its practical usefulness; however, no association mining algorithms have been used to discover rules from uncertain data. This is because previous mining algorithms merely focused on a certain data set.

This paper has made several contributions. First, this paper proposed a representation scheme for uncertain data. This representation was based on possibility distributions, since possibility theory establishes a close connection between the concepts of similarity and uncertainty, thereby providing an excellent framework for handling uncertain data. Second, although uncertain data exists in practical databases, no mining algorithms have been developed to discover knowledge from uncertain data. This paper developed an effective mining algorithm that finds fuzzy association rules from uncertain data. Experimental results from the survey data show the feasibility of the proposed mining algorithm.

There are several issues that can be addressed in future research. First, we assumed that the membership functions were known in advance. In the future, we would attempt to adopt other data mining technologies, such as clustering [18,25], to obtain the similarity matrix. Then, experts would not need to assign them in the preprocessing phase, and the membership functions acquisition bottleneck would be avoided. In addition, this paper assumed that experts assigned the maximum deviation threshold. In future research, we would attempt to automatically infer the maximum deviation threshold from the raw data, avoiding the maximum deviation acquisition bottleneck. Finally, the variation of the generated rules as the similarity matrix's parameter settings varied is an interesting issue. In this regard, it would be helpful for future studies to test the sensitivity of the rule mining algorithm.

Appendix A

In this study, students were also allowed to use possibility to answer the seven questions listed in Table 14.

The User can use the possibility to represent each choice. (0–100%)

Table 14 The seven questions in the Teaching Evaluation and Course Evaluation

No.	The questions
1	How suitable was the pace of the course to my learning?
	☐very poor ☐poor ☐average ☐good ☐very good
2	How suitable was the pace of the lectures to my learning?
	☐very poor ☐poor ☐average ☐good ☐very good
3	How comfortable did I feel in seeking guidance and support from my lecturer?
	☐very poor ☐poor ☐average ☐good ☐very good
4	How professional was the lecturer?
	☐very poor ☐poor ☐average ☐good ☐very good
5	How approachable and courteous was the lecturer?
	☐very poor ☐poor ☐average ☐good ☐very good
6	What teaching methods do you think are appropriate for this course?
	☐Black-board instruction ☐Computer-aided instruction ☐Audio-visual instruction
7	What score do you think you will receive in this course this semester? (1–100)

References

1. Agrawal R, Imielinski T, Swami A (1993) Mining association rules between sets of items in large databases. In: Proceedings of ACM SIGMOD, Washington, DC, pp 207–216
2. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. In: Proceedings of the 20th international conference on very large data bases. Santiago, Chile, pp 487–499
3. Alcala-Fdez J, Alcala R, Gacto MJ, Herrera F (2009) Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms. Fuzzy Set Syst 160(7):905–921
4. Arslan A, Kaya M (2001) Determination of fuzzy logic membership functions using genetic algorithms. Fuzzy Set Syst 118(2):297–306
5. Berry M, Linoff G (1997) Data mining techniques: for marketing, sales, and customer support. Wiley, New York
6. Berti-Equille L (2007) Data quality awareness: a case study for cost-optimal association rule mining. Knowl Inf Syst 11(2):191–215
7. Chen YL, Ho CY (2005) A sampling-based method for mining frequent patterns from databases. Lect Notes Artif Int 3614:536–545
8. Chen YL, Huang TCK (2005) Discovering fuzzy time-interval sequential patterns in sequence databases. IEEE Trans Syst Man Cybern B 35(5):959–972
9. Chen YL, Huang TCK (2006) A new approach for discovering fuzzy quantitative sequential patterns in sequence databases. Fuzzy Set Syst 157(12):1641–1661
10. Chen YL, Shen CC (2005) Mining generalized knowledge from ordered data through attribute-oriented induction techniques. Eur J Oper Res 166(1):221–245
11. Chen YL, Tang K, Shen RJ, Hu YH (2005) Market basket analysis in a multiple store environment. Decis Support Syst 40(2):339–354
12. Chen YL, Weng CH (2008) Mining association rules from imprecise ordinal data. Fuzzy Set Syst 159(4):60–474
13. Cheng J, Ke Y, Ng W (2008) A survey on algorithms for mining frequent itemsets over data streams. Knowl Inf Syst 16(1):1–27
14. Cheung DW, Ng VT, Fu AW, Fu YJ (1996) Efficient mining of association rules in distributed databases. IEEE Trans Knowl Data Eng 8(6):911–922
15. Chui C, Kao B, Hung E (2007) Mining frequent itemsets from uncertain data. Lect Notes Comput Sci 4426:47–58
16. Chung SM, Luo C (2008) Efficient mining of maximal frequent itemsets from databases on a cluster of workstations. Knowl Inf Syst 16(3):359–391
17. Conci A, Castro EMMM (2002) Image mining by content. Expert Syst Appl 23(4):377–383

18. De SK, Krishna PR (2004) Clustering web transactions using rough approximation. Fuzzy Set Syst 148(1):131–138
19. Delgado M, Marin N, Sanchez D, Vila MA (2003) Fuzzy association rules: general model and applications. IEEE Trans Fuzzy Syst 11(2):214–225
20. Dempster AP (1967) Upper and lower probabilities induced by a multivalued mapping. Anuals Math Stat 38:325–339
21. Denguir-Rekik A, Mauris G, Montmain J (2006) Propagation of uncertainty by the possibility theory in Choquet integral-based decision making: application to an E-commerce website choice support. IEEE Trans Instrum Meas 55(3):721–728
22. Djouadi Y, Redaoui S, Amroun K (2007) Mining fuzzy association rules from uncertain data. In: IEEE international fuzzy systems conference, London, pp 1-6
23. Dubois D (2006) Possibility theory and statistical reasoning. Comput Stat Data Anal 51(1):47–69
24. Dubois D, Prade H (1988) Representation and combination of uncertainty with belief functions and possibility measures. Comput Intell 4(3):244–264
25. Eick CF, Rouhana A, Bagherjeiran A, Vilalta R (2006) Using clustering to learn distance functions for supervised similarity assessment. Eng Appl Artif Intel 19(4):395–401
26. Han J, Fu Y (1995) Discovery of multiple-level association rules from large databases. In: Proceedings of 1995 international conference on very large databases. Zurmh, Switzerland, pp 420–431
27. Han J, Kamber WM (2001) Data mining: concepts and techniques. Morgan Kaufmann, San Francisco
28. Holt JD, Chung SM (2002) Mining association rules using inverted hashing and pruning. Inform Process Lett 83(4):211–220
29. Hong TP, Chen JB (1999) Finding relevant attributes and membership functions. Fuzzy Set Syst 103(3):389–404
30. Hong TP, Kuo CS, Wang SL (2004) A fuzzy AprioriTid mining algorithm with reduced computational time. Appl Soft Comput 5(1):1–10
31. Hong TP, Lee CY (1996) Induction of fuzzy rules and membership functions from training examples. Fuzzy Set Syst 84(1):33–47
32. Hong TP, Lin KY, Wang SL (2003) Fuzzy data mining for interesting generalized association rules. Fuzzy Set Syst 138(2):255–269
33. Hsu HM, Wang WP (2001) Possibilistic programming in production planning of assemble-to-order environments. Fuzzy Set Syst 119(1):59–70
34. Hsu PY, Chen YL, Ling CC (2004) Algorithms for mining association rules in bag databases. Inf Sci 166(1):31–47
35. Hu YC, Chen RS, Tzeng GH (2003) Discovering fuzzy association rules using fuzzy partition methods. Knowl Based Syst 16(3):137–147
36. Hu YC, Tzeng GH (2003) Elicitation of classification rules by fuzzy data mining. Eng Appl Artif Intel 16(7-8):709–716
37. Hüllermeier E (2003) Possibilistic instance-based learning. Artif Intell 148(1-2):335–383
38. Ke Y, Cheng J, Ng W (2008) An information-theoretic approach to quantitative association rule mining. Knowl Inf Syst 16(2):245–258
39. Lee G, Lee KL, Chen ALP (2001) Efficient graph-Based algorithms for discovering and maintaining association rules in large databases. Knowl Inf Syst 3(3):338–355
40. Leung CW, Chan SC, Chung F (2006) A collaborative filtering framework based on fuzzy association rules and multiple-level similarity. Knowl Inf Syst 10(3):357–381
41. Loiseau Y, Prade H, Boughanem M (2004) Qualitative pattern matching with linguistic terms. AI Commun 17(1):25–34
42. Ordonez C, Ezquerra N, Santana CA (2006) Constraining and summarizing association rules in medical data. Knowl Inf Syst 9(3):259–283
43. Oussalah M, Maaref H, Barret C (2001) New fusion methodology approach and application to mobile robotics: investigation in the framework of possibility theory. Inf Fusion 2(1):31–48
44. Pedrycz W (1998) Fuzzy set technology in knowledge discovery. Fuzzy Set Syst 98(3):279–290
45. Prade H, Testemale C (1984) Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries. Inf Sci 34(22):115–143
46. Rastogi R, Shim K (2001) Mining optimized support rules for numeric attributes. Inf Syst 26(6):425–444
47. Shafer G. (1976) A Mathematical Theory of Evidence. Princeton University Press, Princeton
48. Shu JY, Tsang ECC, Yeung DS (2001) Query fuzzy association rules in relational database. In: IFSA World Congress and 20th NAFIPS international conference. Vancouver, BC, Canada, pp 2989-2993
49. Shyu ML, Haruechaiyasak C, Chen SC, Premaratne K, Mining association rules with uncertain item relationships, in [http://www.cs.fiu.edu/~chens/PDF/SCI02.pdf]
50. Wu X, Zhang C, Zhang S (2005) Database classification for multi-database mining. Inform Syst 30(1):71–88

51. Yun H, Ha D, Hwang B, Ryu K Ho (2003) Mining association rules on significant rare data using relative support. J Syst Software 67(3):181–191
52. Zadeh LA (1965) Fuzzy sets. Inf Control 8:338–353
53. Zadeh LA (1978) Fuzzy sets as a basis for a theory of possibility. Fuzzy Set Syst 1:3–28
54. Zhang S, Wu X, Zhang C, Lu J (2008) Computing the minimum-support for mining frequent patterns. Knowl Inf Syst 15(2):233–257

Author Biographies

Cheng-Hsiung Weng is currently working as an assistant professor in the Department of Management Information Systems at Central Taiwan University of Science and Technology of Taiwan. He received his Ph.D. degree in information management from National Central University, Chung-Li, Taiwan. His current research interests include data mining, knowledge management, decision support system, and software engineering. He has published papers in Fuzzy Sets and Systems, Knowledge-Based Systems, International Journal of Computer Integrated Manufacturing and others.

Yen-Liang Chen is currently working as a Professor in the Department of Information Management at National Central University of Taiwan. He received his Ph.D. degree in computer science from National Tsing Hua University, Hsinchu, Taiwan. His current research interests include data mining, information retrieval, knowledge management and decision making models. He has published papers in Decision Support Systems, Information & Management, IEEE Transactions on Software Engineering, IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on SMC—part A and part B, Information Systems, Operations Research, Journal of Information Science, Information Sciences, Naval Research Logistics, Transportation Research—part B, European Journal of Operational Research, Knowledge-based Systems and many others.