Mining association rules from quantitative data

Article *in* Intelligent Data Analysis · November 1999

Impact Factor: 0.61 · DOI: 10.1016/S1088-467X(99)00028-1

3 authors, including:

Tzung-Pei Hong
National University of Kaohsiung
561 PUBLICATIONS **4,319** CITATIONS

Mining Association Rules from Quantitative Data*

Tzung-Pei Hong[†**], *Chan-Sheng Kuo*[‡], *and Sheng-Chai Chi*[‡]

[†]Department of Information Management
[‡]Graduate School of Management Science
I-Shou University
Kaohsiung, 84008, Taiwan, R.O.C.
e-mail: tphong@csa500.isu.edu.tw

ABSTRACT

Data mining is the process of extracting desirable knowledge or interesting patterns from existing databases for specific purposes. Most conventional data-mining algorithms identify the relationships among transactions using binary values, however, transactions with quantitative values are commonly seen in real-world applications. This paper thus proposes a new data-mining algorithm for extracting interesting knowledge from transactions stored as quantitative values. The proposed algorithm integrates fuzzy set concepts and the *apriori* mining algorithm to find interesting fuzzy association rules in given transaction data sets. Experiments with student grades at I-Shou University were also made to verify the performance of the proposed algorithm.

Keywords: data mining, fuzzy set, association rule, transaction, quantitative value.

---

* This is a modified and expanded version of the paper "A data mining algorithm for transaction data with quantitative values," presented at The Eighth International Fuzzy Systems Association World Congress, 1999.

**Corresponding author.

1. INTRODUCTION

Knowledge discovery in databases (KDD) has become a process of considerable interest in recent years as the amounts of data in many databases have grown tremendously large. KDD means the application of nontrivial procedures for identifying effective, coherent, potentially useful, and previously unknown patterns in large databases [13]. The KDD process generally consists of the following three phases [12, 25].

(1) **Pre-processing:** This consists of all the actions taken before the actual data analysis process starts [12]. Famili *et al* think that it may be performed on the data for the following reasons: solving data problems that may prevent us from performing any type of analysis on the data, understanding the nature of the data, performing a more meaningful data analysis, and extracting more meaningful knowledge from a given set of data.

(2) **Data mining:** This involves applying specific algorithms for extracting patterns or rules from data sets in a particular representation.

(3) **Post-processing:** This translates discovered patterns into forms acceptable

for human beings. It may also make possible visualization of extracted patterns.

Due to the importance of data mining to KDD, many researchers in database and machine learning fields are primarily interested in this new research topic because it offers opportunities to discover useful information and important relevant patterns in large databases, thus helping decision-makers easily analyze the data and make good decisions regarding the domains concerned.

Data-mining is most commonly used in attempts to induce association rules from transaction data. Most previous studies have only shown, however, how binary valued transaction data may be handled. Transaction data in real-world applications do not usually consist of quantitative values, so designing a sophisticated data-mining algorithm able to deal with various types of data presents a challenge to workers in this research field.

Fuzzy set theory is being used more and more frequently in intelligent systems because of its simplicity and similarity to human reasoning [22]. The theory has been applied in fields such as manufacturing, engineering, diagnosis, economics, among others [16, 22, 24, 36]. Several fuzzy learning algorithms for inducing rules from

given sets of data have been designed and used to good effect with specific domains [5-6, 8, 11, 15, 17-21, 30, 32-33]. Strategies based on decision trees [9] were proposed in [10, 26-27, 30, 34-35], and Wang et al. proposed a fuzzy version space learning strategy for managing vague information [32].

This paper integrates fuzzy-set concepts with the *apriori* mining algorithm [4] and uses the result to find interesting itemsets and fuzzy association rules in transaction data with quantitative values. A new mining algorithm, called the Fuzzy Transaction Data-mining Algorithm (FTDA) is proposed. It transforms quantitative values in transactions into linguistic terms, then filters them to find association rules by modifying the *apriori* mining algorithm [4] .

The remaining parts of this paper are organized as follows. Agrawal et al.'s mining algorithms are reviewed in Section 2. Fuzzy-set concepts are introduced in Section 3. The proposed data-mining algorithm for quantitative values is described in Section 4. An example is given to illustrate the proposed algorithm in Section 5. Experiments to demonstrate the performance of the proposed data-mining algorithm are stated in Section 6. Conclusions and proposal of future work are given in Section 7.

2. Related Works

The goal of data mining is to discover important associations among items such that the presence of some items in a transaction will imply the presence of some other items. To achieve this purpose, Agrawal and his co-workers proposed several mining algorithms based on the concept of large itemsets to find association rules in transaction data [1-4]. They divided the mining process into two phases. In the first phase, candidate itemsets were generated and counted by scanning the transaction data. If the number of an itemset appearing in the transactions was larger than a pre-defined threshold value (called minimum support), the itemset was considered a large itemset. Itemsets containing only one item were processed first. Large itemsets containing only single items were then combined to form candidate itemsets containing two items. This process was repeated until all large itemsets had been found. In the second phase, association rules were induced from the large itemsets found in the first phase. All possible association combinations for each large itemset were formed, and those with calculated confidence values larger than a predefined threshold (called minimum confidence) were output as association rules.

In addition to proposing methods for mining association rules from transactions of binary values, Agrawal et al. also proposed a method [31] for mining association rules from those with quantitative and categorical attributes. Their proposed method first determines the number of partitions for each quantitative attribute, and then maps all possible values of each attribute into a set of consecutive integers. It then finds large itemsets whose support values are greater than the user-specified minimum-support levels. These large itemsets are then processed to generate association rules, and rules of interest to users are output.

Some other methods were also proposed to handle numeric attributes and to derive association rules. Fukuda *et al* introduced the optimized association-rule problem and permitted association rules to contain single uninstantiated conditions on the left-hand side [14]. They also proposed schemes to determine the conditions such that the confidence or support values of the rules are maximized. However, their schemes were only suitable for a single optimal region. Rastogi and Shim thus extended the problem for more than one optimal regions, and showed that the problem was NP-hard even for the case of one uninstantiated numeric attribute [28, 29]. Some works also used fuzzy set theory and data mining technology to solve classification problems [7, 23].

In this paper, we use fuzzy set concepts to mine association rules from transactions with quantitative attributes. The mined rules are expressed in linguistic terms, which are more natural and understandable for human beings.

3. Review of Fuzzy Set Concepts

Fuzzy set theory was first proposed by Zadeh and Goguen in 1965 [37]. Fuzzy set theory is primarily concerned with quantifying and reasoning using natural language in which words can have ambiguous meanings. This can be thought of as an extension of traditional crisp sets, in which each element must either be in or not in a set.

Formally, the process by which individuals from a universal set *X* are determined to be either members or non-members of a crisp set can be defined by a *characteristic or discrimination function* [37]. For a given crisp set *A*, this function assigns a value $\mu_A(x)$ to every $x \in X$ such that

$$\mu_A(x) = \begin{cases} 1 & \text{if and only if } x \in A \\ 0 & \text{if and only if } x \notin A. \end{cases}$$

Thus, the function maps elements of the universal set to the set containing 0 and 1. This kind of function can be generalized such that the values assigned to the elements of the universal set fall within specified ranges, referred to as the membership grades of these elements in the set. Larger values denote higher degrees of set membership. Such a function is called the membership function, $\mu_A(x)$, by which a fuzzy set A is usually defined. This function is represented by

$$\mu_A : X \to [0,1],$$

where [0, 1] denotes the interval of real numbers from 0 to 1, inclusive. The function can also be generalized to any real interval instead of [0,1].

A special notation is often used in the literature to represent fuzzy sets. Assume that x_1 to x_n are the elements in fuzzy set A, and μ_1 to μ_n are, respectively, their grades of membership in A. A is then usually represented as follows:

$$A = \mu_1 / x_1 + \mu_2 / x_2 + ... + \mu_n / x_n.$$

An *α-cut* of a fuzzy set A is a crisp set A_α that contains all elements in the universal set X with membership grades in A greater than or equal to a specified value of α. This definition can be written as:

$$A_\alpha = \{x \in X \mid \mu_A(x) \geq \alpha\}.$$

The *scalar cardinality* of a fuzzy set A defined on a finite universal set X is the summation of the membership grades of all the elements of X in A. Thus,

$$|A| = \sum_{x \in X} \mu_A(x) .$$

Among operations on fuzzy sets are the basic and commonly used *complementation*, *union* and *intersection,* as proposed by Zadeh.

(1) The complementation of a fuzzy set A is denoted by $\neg A$, and the membership function of $\neg A$ is given by:

$$\mu_{\neg A}(x) = 1 - \mu_{\neg A}(x), \qquad \forall x \in X .$$

(2) The intersection of two fuzzy sets A and B is denoted by $A \cap B$, and the membership function of $A \cap B$ is given by:

$$\mu_{A \cap B}(x) = \min\left\{ \mu_A(x), \mu_B(x) \right\}, \qquad \forall x \in X .$$

(3) The union of fuzzy sets A and B is denoted by $A \cup B$, and the membership function of $A \cup B$ is given by:

$$\mu_{A \cup B}(x) = \max\left\{ \mu_A(x), \mu_B(x) \right\}, \qquad \forall x \in X .$$

4. The fuzzy data-mining algorithm for quantitative values

In this section, the fuzzy concepts are used in the *apriori* data-mining algorithm to discover useful association rules among quantitative values. The following notation is used in this paper.

n: the total amount of transaction data;

m: the total number of attributes;

$D^{(i)}$ *: the i-th transaction datum,* $1 \le i \le n$;

A_j *: the j-th attribute,* $1 \le j \le m$;

$|A_j|$ *: the number of fuzzy regions for* A_j;

R_{jk} *: the k-th fuzzy region of* A_j, $1 \le k \le |A_j|$;

$v_j^{(i)}$: *the quantitative value of* A_j *in* $D^{(i)}$;

$f_j^{(i)}$: *the fuzzy set converted from* $v_j^{(i)}$;

$f_{jk}^{(i)}$: *the membership value of* $v_j^{(i)}$ *in Region* R_{jk} ;

$count_{jk}$: *the summation of* $f_{jk}^{(i)}$, *i=1 to n;*

$count_j^{\max}$: *the maximum count value among* $count_{jk}$ *values, k=1 to* $|A_j|$;

$R_j^{\max}$: *the fuzzy region of* A_j *with* $count_j^{\max}$;

α : *the predefined minimum support level;*

λ : *the predefined minimum confidence value;*

C_r: *the set of candidate itemsets with r attributes (items);*

L_r : *the set of large itemsets with r attributes (items).*

The proposed fuzzy mining algorithm first uses membership functions to transform each quantitative value into a fuzzy set in linguistic terms. The algorithm then calculates the scalar cardinalities of all linguistic terms in the transaction data. Each attribute uses only the linguistic term with the maximum cardinality in later mining processes, thus keeping the number of items the same as that of the original attributes. The algorithm is therefore focused on the most important linguistic terms, which reduces its time complexity. The mining process using fuzzy counts is then performed to find fuzzy association rules. Details of the proposed mining algorithm

are described below.

The FTDA algorithm:

INPUT: A body of *n* transaction data, each with *m* attribute values, a set of membership functions, a predefined minimum support value α, and a predefined confidence value λ.

OUTPUT: A set of fuzzy association rules.

STEP 1: Transform the quantitative value $v_j^{(i)}$ of each transaction datum $D^{(i)}$, *i*=1 to *n,* for each attribute A_j, *j*=1 to *m*, into a fuzzy set $f_j^{(i)}$ represented as $\left(\frac{f_{j_1}^{(i)}}{R_{j_1}} + \frac{f_{j_2}^{(i)}}{R_{j_2}} + ... + \frac{f_{j_l}^{(i)}}{R_{j_l}} \right)$ using the given membership functions, where R_{jk} is the *k*-th fuzzy region of attribute A_j, $f_{jk}^{(i)}$ is $v_j^{(i)}$'s fuzzy membership value in region R_{jk}, and *l* (=$|A_j|$) is the number of fuzzy regions for A_j.

STEP 2: Calculate the scalar cardinality of each attribute region R_{jk} in the transaction data:

$$count_{jk} = \sum_{i=1}^{n} f_{jk}^{(i)} .$$

STEP 3: Find $count_j^{\max} = \underset{k=1}{\overset{|A_j|}{MAX}}(count_{jk})$, for *j*= 1 to *m,* where $|A_j|$ is the number of fuzzy regions for A_j. Let $R_j^{\max}$ be the region with $count_j^{\max}$ for attribute A_j. $R_j^{\max}$ will be used to represent this attribute in later mining processing.

STEP 4: Check whether the $count_j^{\max}$ of each $R_j^{\max}$, *j*=1 to *m*, is larger than or equal to

the predefined minimum support value α . If $R_j^{\max}$ is equal to or greater than the minimum support value, put it in the set of large 1-itemsets (L_1). That is,

$$L_1 = \left\{ R_j^{\max} \middle| Count_j^{\max} \geq \alpha, 1 \leq j \leq m \right\}.$$

STEP 5: Set r=1, where r is used to represent the number of items kept in the current large itemsets.

STEP 6: Generate the candidate set C_{r+1} from L_r in a way similar to that in the *apriori* algorithm [4]. That is, the algorithm first joins L_r and L_r assuming that r-1 items in the two itemsets are the same and the other one is different. It then keeps in C_{r+1} the itemsets, which have all their sub-itemsets of r items existing in L_r.

STEP 7: Do the following substeps for each newly formed (r+1)-itemset s with items $(s_1, s_2, ..., s_{r+1})$ in C_{r+1}:

(a) Calculate the fuzzy value of each transaction data $D^{(i)}$ in s as $f_s^{(i)} = f_{s_1}^{(i)} \Lambda f_{s_2}^{(i)} \Lambda \ldots \Lambda f_{s_{r+1}}^{(i)}$, where $f_{s_j}^{(i)}$ is the membership value of $D^{(i)}$ in region s_j. If the minimum operator is used for the intersection, then $f_s^{(i)} = \underset{j=1}{\overset{r+1}{Min}} f_{s_j}^{(i)}$.

(b) Calculate the scalar cardinality of s in the transaction data as:

$$count_s = \sum_{i=1}^{n} f_s^{(i)} .$$

(c) If $count_s$ is larger than or equal to the predefined minimum support value α, put s in L_{r+1}.

STEP 8: IF L_{r+1} is null, then do the next step; otherwise, set r=r+1 and repeat STEPs 6 to 8.

STEP 9: Construct the association rules for all large q-itemset s with items $(s_1, s_2, ..., s_q)$, $q \geq 2$, using the following substeps:

(a) Form all possible association rules as follows:

$$s_1 \Lambda ... \Lambda s_{k-1} \Lambda s_{k+1} \Lambda ... \Lambda s_q \rightarrow s_k, k=1 \text{ to } q.$$

(b) Calculate the confidence values of all association rules using:

$$\frac{\sum_{i=1}^{n} f_s^{(i)}}{\sum_{i=1}^{n} (f_{s_1}^{(i)} \Lambda ... \Lambda f_{s_{k-1}}^{(i)}, f_{s_{k+1}}^{(i)} \Lambda ... \Lambda f_{s_q}^{(i)})}.$$

STEP 10: Output the rules with confidence values larger than or equal to the predefined confidence threshold λ.

After STEP 10, the rules output can serve as meta-knowledge concerning the given transactions. Since each attribute uses only the linguistic term with the maximum cardinality in the mining process, the number of items is thus the same as that of the original attributes. Instead, if in Step 4, all the regions with support values larger than the threshold are considered, the algorithm will generate more rules, but

will take much more time than the proposed method. Trade-off thus exists between the rule completeness and the time complexity.

5. An Example

In this section, an example is given to illustrate the proposed data-mining algorithm. This is a simple example to show how the proposed algorithm can be used to generate association rules for course grades according to historical data concerning students' course scores. The data set includes 10 transactions, as shown in Table 1.

Table 1:The set of students' course scores in the example

Case No.	OOP	DB	ST	DS	MIS
1	86	77	86	71	68
2	61	87	89	77	80
3	84	89	86	79	89
4	73	86	79	84	62
5	70	85	87	72	79
6	65	67	86	61	87
7	71	87	75	71	80
8	86	69	64	84	88
9	75	65	86	86	79
10	83	68	65	85	89

Each case consists of five course scores: Object-Oriented Programming (denoted OOP), Database (denoted DB), Statistics (denoted ST), Data Structure (denoted DS),

and Management Information System (denoted MIS). Each course is thought of as an attribute in the mining process. Assume the fuzzy membership functions for the course scores are as shown in Figure 1.

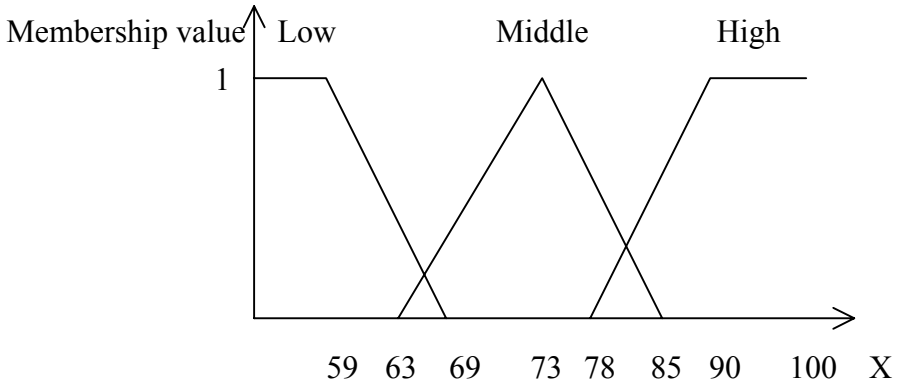

Figure 1: The membership function used in this example

In this example, each attribute has three fuzzy regions: *Low*, *Middle*, and *High*. Thus, three fuzzy membership values are produced for each course score according to the predefined membership functions. For the transaction data in Table 1, the proposed mining algorithm proceeds as follows.

STEP 1: Transform the quantitative values of each transaction datum into fuzzy sets. Take the OOP score in Case 1 as an example. The score "86" is converted into a fuzzy set $(\frac{0.0}{Low}+\frac{0.0}{Middle}+\frac{0.7}{High})$ using the given membership functions. This step is repeated for the other cases and courses, and the results are shown in Table 2.

Table 2: The fuzzy sets transformed from the data in Table 1

Case No.	**OOP**			**DB**			**ST**			**DS**			**MIS**		
	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H
1	0.0	0.0	0.7	0.0	0.7	0.0	0.0	0.0	0.7	0.0	0.8	0.0	0.1	0.5	0.0
2	0.8	0.0	0.0	0.0	0.0	0.8	0.0	0.0	0.9	0.0	0.7	0.0	0.0	0.4	0.2
3	0.0	0.1	0.5	0.0	0.0	0.9	0.0	0.0	0.7	0.0	0.5	0.1	0.0	0.0	0.9
4	0.0	1.0	0.0	0.0	0.0	0.7	0.0	0.5	0.1	0.0	0.1	0.5	0.7	0.0	0.0
5	0.0	0.7	0.0	0.0	0.0	0.6	0.0	0.0	0.8	0.0	0.9	0.0	0.0	0.5	0.1
6	0.4	0.2	0.0	0.2	0.4	0.0	0.0	0.0	0.7	0.8	0.0	0.0	0.0	0.0	0.8
7	0.0	0.8	0.0	0.0	0.0	0.8	0.0	0.8	0.0	0.0	0.8	0.0	0.0	0.4	0.2
8	0.0	0.0	0.7	0.0	0.6	0.0	0.5	0.1	0.0	0.0	0.1	0.5	0.0	0.0	0.8
9	0.0	0.8	0.0	0.4	0.2	0.0	0.0	0.0	0.7	0.0	0.0	0.7	0.0	0.5	0.1
10	0.0	0.2	0.4	0.1	0.5	0.0	0.4	0.2	0.0	0.0	0.0	0.6	0.0	0.0	0.9
Count	1.2	3.8	2.3	0.7	1.7	3.8	0.9	1.6	4.6	0.8	3.9	2.4	0.8	2.3	4.0

STEP 2: Calculate the scalar cardinality of each attribute region in the transactions as the *count* value. Take the region *OOP.Low* as an example. Its scalar cardinality = (0.0 + 0.8 + 0.0 + 0.0 + … + 0.0) = 1.2. This step is repeated for the other regions, and the results are shown in the bottom line of Table 2.

STEP 3: Find the region with the highest count among the three possible regions for each attribute. Take the course *OOP* as an example. The count is 1.2 for *Low*, 3.8 for *Middle*, and 2.3 for *High*. Since the count for *Middle* is the highest among the three counts, the region *Middle* is thus used to represent the course *OOP* in later mining processing. This step is repeated for the other regions, and "*High*" is chosen

for DB, ST and MIS, and "*Middle*" is chosen for OOP and DS. The number of items chosen are thus the same as that of the original attributes, meaning the algorithm is focused on the important items, and the time complexity could be reduced.

STEP 4: Check whether the count of any region (item) kept in STEP 3 is larger than or equal to the predefined minimum support value α. Assume in this example, α is set at 2.5. Since the count values of OOP.Middle, DB.High, ST.High, DS.Middle and MIS.High are all larger than 2.5, these items are put in L_1 (Table 3).

Table 3: The set of large 1-itemsets L_1 for this example

Itemset	count
OOP.Middle	3.8
DB.High	3.8
ST.High	4.6
DS.Middle	3.9
MIS.High	4.0

STEP 5: Set r=1.

STEP 6: Generate the candidate set C_{r+1} from L_r. C_2 is first generated from L_1 as follows: (OOP.Middle, DB.High), (OOP.Middle, ST.High), (OOP.Middle, DS.Middle), (OOP.Middle, MIS.High), (DB.High, ST.High), (DB.High, DS.Middle), (DB.High, MIS.High), (ST.High, DS.Middle), (ST.High, MIS.High), and (DS.Middle, MIS.High).

STEP 7: Do the following substeps for each newly formed candidate itemset.

(a) Calculate the fuzzy membership value of each transaction datum. Here, the minimum operator is used for the intersection. Take (OOP.Middle, DB.High) as an example. The derived membership value for Case 1 is calculated as: *min*(0.0, 0.0)=0.0. The results for the other cases are shown in Table 4.

Table 4: The membership values for OOP.Middle Λ *DB.High*

Case	OOP.Middle	DB.High	OOP.Middle $\cap$ DB.High
1	0.0	0.0	0.0
2	0.0	0.8	0.0
3	0.1	0.9	0.1
4	1.0	0.7	0.7
5	0.7	0.6	0.6
6	0.2	0.0	0.0
7	0.8	0.8	0.8
8	0.0	0.0	0.0
9	0.8	0.0	0.0
10	0.2	0.0	0.0

The results for the other 2-itemsets can be derived in similar fashion.

(b) Calculate the scalar cardinality (count) of each candidate 2-itemset in the transaction data. Results for this example are shown in Table 5.

Table 5: *The fuzzy counts of the itemsets in C_2*

Itemset	count
(OOP.Middle, DB.High)	2.2
(OOP.Middle, ST.High)	1.8
(OOP.Middle, DS.Middle)	1.7
(OOP.Middle, MIS.High)	0.9
(DB.High, ST.High)	2.2
(DB.High,DS.Middle)	2.7
(DB.High, MIS.High)	1.4
(ST.High, DS.Middle)	2.8
(ST.High, MIS.High)	1.8
(DS.Middle, MIS.High)	1.1

(c) Check whether these counts are larger than or equal to the predefined minimum support value 2.5. Two itemsets, (DB.High, DS.Middle) and (ST.High, DS.Middle), are thus kept in L_2 (Table 6).

Table 6: The itemsets and their fuzzy counts in L_2

Itemset	count
(DB.High, and DS.Middle)	2.7
(ST.High, and DS.Middle)	2.8

STEP 8: IF L_{r+1} is null, then do the next step; otherwise, set r=r+1 and repeat STEPs 6 to 8. Since L_2 is not null in the example above, r=r+1=2. STEPs 6 to 8 are then repeated to find L_3. C_3 is first generated from L_2, and only the itemset (DB.High, ST.High, DS.Middle) is formed. Its count is calculated as 1.9, smaller than 2.5. It is not put in L_3, and L_3 is thus an empty set. STEP 9 then begins.

STEP 9: Construct the association rules for each large itemset using the following substeps.

(a) Form all possible association rules. The following four association rules are possible:

If DB = High, then DS = Middle;

If DS = Middle, then DB = High;

If ST = High, then DS = Middle;

If DS = Middle, then ST = High.

(b) Calculate the confidence factors for the above association rules. Assume the given confidence threshold λ is 0.7. Take the first association rule as an example. The fuzzy count of DB.High $\cap$ DS.Middle is calculated as shown in Table 7.

Table 7: The fuzzy counts for DB.High ∩ DS.Middle

Case	DB.High	DS.Middle	DB.High ∩ DS.Middle
1	0.0	0.8	0.0
2	0.8	0.7	0.7
3	0.9	0.5	0.5
4	0.7	0.1	0.1
5	0.6	0.9	0.6
6	0.0	0.0	0.0
7	0.8	0.8	0.8
8	0.0	0.1	0.0
9	0.0	0.0	0.0
10	0.0	0.0	0.0
Count	3.8	3.9	2.7

The confidence factor for the association rule "*If DB = High, then DS = Middle"* is then:

$$\frac{\sum_{i=1}^{10}(DB.High \cap DS.Middle)}{\sum_{i=1}^{10}(DB.High)} = \frac{2.7}{3.8} = 0.71.$$

Results for the other three rules are shown below.

"If DS = Middle, then DB = High" has a confidence factor of 0.69;

"If ST = High, then DS = Middle" has a confidence factor of 0.61;

"If DS = Middle, then ST = High" has a confidence factor of 0.72.

STEP 10: Check whether the confidence factors of the above association rules

are larger than or equal to the predefined confidence threshold λ. Since the confidence λ was set at 0.7 in this example, the following two rules are thus output to users:

1. If the Database score is high, then the Data Structure score is middle, with a confidence factor of 0.71;

2. If the Data Structure score is middle, then the Statistics score is high, with a confidence factor of 0.72.

The two rules above are thus output as meta-knowledge concerning the given transactions.

6. Experimental Results

Student score data from the Department of Information Management at I-Shou University, Taiwan, were used to show the feasibility of the proposed mining algorithm. A total of 260 transactions were included in the data set. Each transaction consisted of scores that a student had gotten. Execution of the mining algorithm was performed on a Pentium-PC.

In the experiments, we set α =70 and λ =0.6. A total of 31 rules were mined out. Three rule mined out are shown below as an example.

1. *If the Management Information Systems score is middle, then the Business Data Communication score is middle, with a confidence factor of 0.73.*

2. *If the Operation Systems score is middle, then the System Analysis and Design score is High, with a confidence factor of 0.70.*

3. *If the Operations Research score is low, then the managerial mathematics score is low, with a confidence factor of 0.61.*

Experiments were also made to show the relationships between numbers of large itemsets and minimum support values. Results are shown in Figure 2, where λ =0.6.

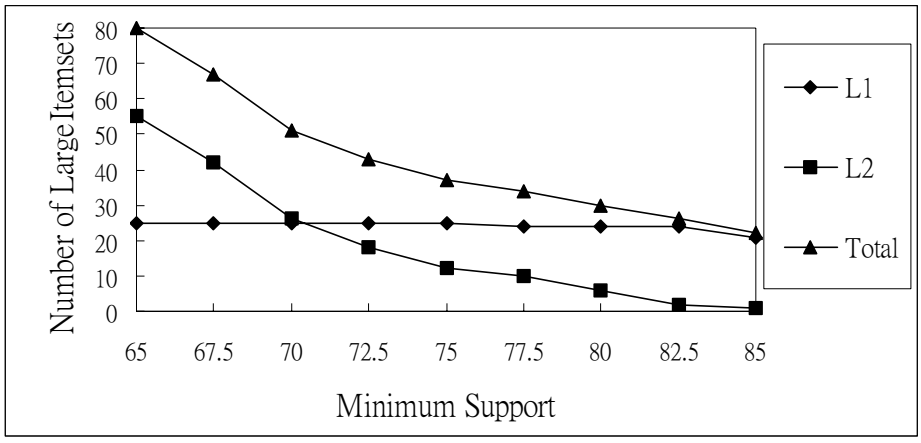

Figure 2. The relationship between numbers of large itemsets and minimum support values.

From Figure 2, it is easily seen that the numbers of large itemsets decreased along with an increase in minimum support values. This is quite consistent with our intuition. The curve of the numbers of large 1-itemsets was also smoother than that of the numbers of large 2-itemsets, meaning that the minimum support value had a larger influence on itemsets with more items.

Experiments were then made to show the relationships between numbers of association rules and minimum support values along with different minimum confidence values. Results are shown in Figure 3.

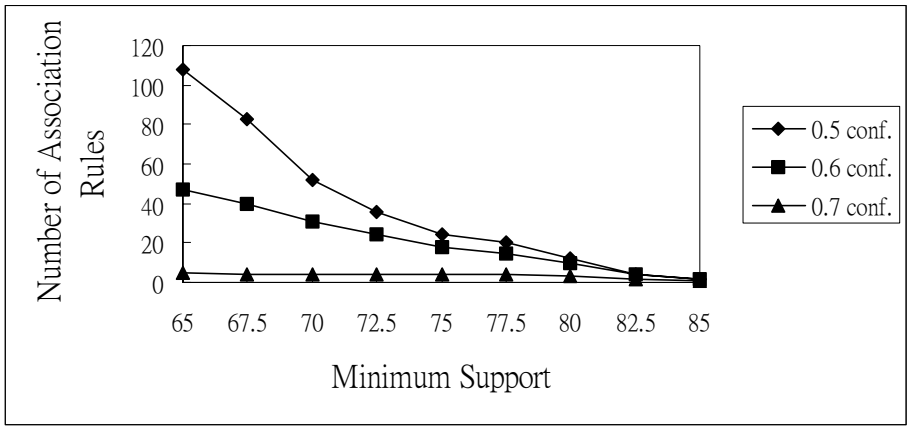

Figure 3. The relationship between numbers of association rules and minimum support values.

From Figure 3, it is easily seen that the numbers of association rules decreased along with the increase in minimum support values. This is also quite consistent with our intuition. Also, the curve of numbers of association rules with larger minimum

confidence values was smoother than that of those with smaller minimum confidence values, meaning that the minimum support value had a large effect on the number of association rules derived from small minimum confidence values.

The relationship between numbers of association rules and minimum confidence values along with various minimum support values is shown in Figure 4.

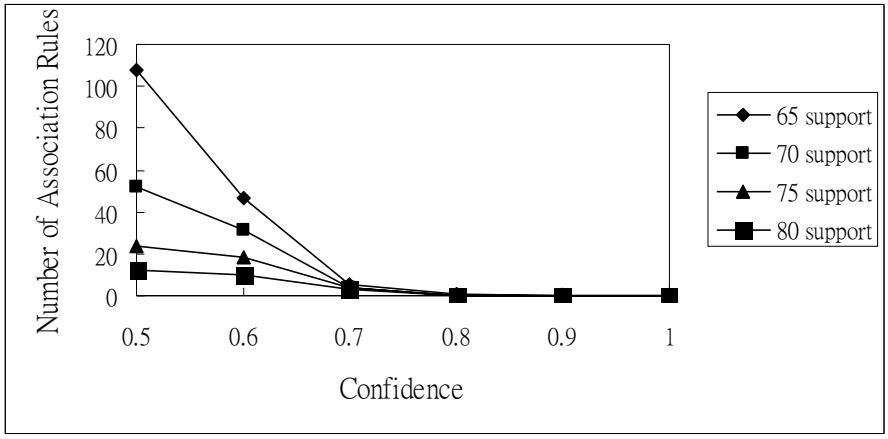

Figure 4. The relationship between numbers of association rules and minimum confidence values.

From Figure 4, it is easily seen that the numbers of association rules decreased along with an increase in minimum confidence values. This is also quite consistent with our intuition. The curve of numbers of association rules with larger minimum support values was smoother than that for smaller minimum support values, meaning that the minimum confidence value had a larger effect on the number of association rules when smaller minimum support values were used. All of the various curves

however converged to 0 as the minimum confidence value approached 1.

Experiments were made to measure the accuracy of the proposed mining algorithm. Results for different amounts of transaction data are shown in Figure 5.

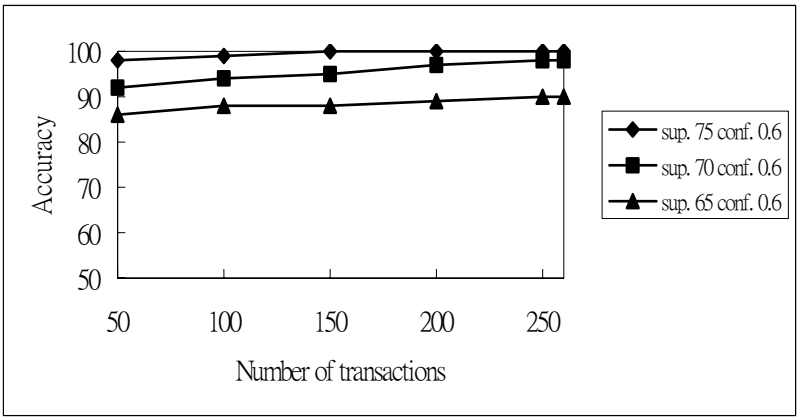

Figure 5: The relationship between accuracy and numbers of transactions.

From Figure 5, it is easily seen that the accuracy increased along with an increase in the number of transactions, meaning that a larger set of sampling data yielded a better mining result. A large minimum support value also yielded a higher accuracy than a small minimum support value.

In this experiment, the association rules mined can actually be used to help the faculty in the Department of Information Management evaluate course programs and

understand the students' learning interests and capacities in the courses.

7. Conclusion and future work

In this paper, we have proposed a generalized data-mining algorithm, called FTDA, which can process transaction data with quantitative values and discover interesting patterns among them. The rules thus mined exhibit quantitative regularity for large databases and can be used to provide some suggestions to appropriate supervisors. The proposed algorithm can also solve conventional transaction-data problems by using degraded membership functions. Experimental results with the students' scores in the Department of Information Management at I-Shou University, Taiwan, show the feasibility of the proposed mining algorithm.

When compared with the traditional crisp-set mining methods for quantitative data, our approach can get smooth mining results due to the fuzzy membership characteristics. Also, when compared with the fuzzy mining methods, which take all the fuzzy regions into consideration, our method can get a good time complexity. Trade-off exists between the rule completeness and the time complexity.

Although the proposed method works well in data mining for quantitative values,

it is just a beginning. There is still much work to be done in this field. Our method assumes that the membership functions are known in advance. In [17-19], we also proposed some fuzzy learning methods to automatically derive the membership functions. In the future, we will attempt to dynamically adjust the membership functions in the proposed mining algorithm to avoid the bottleneck of membership function acquisition. We will also attempt to design specific data-mining models for various problem domains.

Acknowledgment

The authors would like to thank the anonymous referees for their very constructive comments. This research was supported by the National Science Council of the Republic of China under contract NSC89-2213-E-214-003.

References

[1] R. Agrawal, T. Imielinksi and A. Swami, “Mining association rules between sets of items in large database,“ *The 1993 ACM SIGMOD Conference,* Washington DC, USA, 1993.

[2] R. Agrawal, T. Imielinksi and A. Swami, “Database mining: a performance perspective,” *IEEE Transactions on Knowledge and Data Engineering,* Vol. 5, No. 6, 1993, pp. 914-925.

[3] R. Agrawal, R. Srikant and Q. Vu, “Mining association rules with item constraints,” *The Third International Conference on Knowledge Discovery in Databases and Data Mining,* Newport Beach, California, August 1997.

[4] R. Agrawal and R. Srikant, “Fast algorithm for mining association rules,” *The International Conference on Very Large Databases*, 1994, pp. 487-499.

[5] A. F. Blishun, “Fuzzy learning models in expert systems,” *Fuzzy Sets and Systems,* Vol. 22, 1987, pp. 57-70.

[6] L. M. de Campos and S. Moral, “Learning rules for a fuzzy inference model,” *Fuzzy Sets and Systems*, Vol. 59, 1993, pp. 247-257.

[7] K. C. C. Chan and W. H. Au, “Mining fuzzy association rules,” *The 6th ACM International Conference on Information and Knowledge Management*, 1997.

[8] R. L. P. Chang and T. Pavliddis, “Fuzzy decision tree algorithms,” *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 7, 1977, pp. 28-35.

[9] C. Clair, C. Liu and N. Pissinou, “Attribute weighting: a method of applying domain knowledge in the decision tree process,” *The Seventh International Conference on Information and Knowledge Management*, 1998, pp. 259-266.

[10] P. Clark and T. Niblett, “The CN2 induction algorithm,” *Machine Learning*, Vol. 3, 1989, pp. 261-283.

[11] M. Delgado and A. Gonzalez, “An inductive learning procedure to identify fuzzy systems,” *Fuzzy Sets and Systems*, Vol. 55, 1993, pp. 121-132.

[12] A. Famili, W. M. Shen, R. Weber and E. Simoudis, "Data preprocessing and intelligent data analysis," *Intelligent Data Analysis*, Vol. 1, No. 1, 1997.

[13] W. J. Frawley, G. Piatetsky-Shapiro and C. J. Matheus, “Knowledge discovery in databases: an overview,” *The AAAI Workshop on Knowledge Discovery in Databases*, 1991, pp. 1-27.

[14] T. Fukuda, Y. Morimoto, S. Morishita and T. Tokuyama, "Mining optimized association rules for numeric attributes," *The ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, June 1996, pp. 182-191.

[15] A.Gonzalez, “A learning methodology in uncertain and imprecise environments,” *International Journal of Intelligent Systems*, Vol. 10, 1995, pp. 57-371.

[16] I. Graham and P. L. Jones, *Expert Systems – Knowledge, Uncertainty and Decision,* Chapman and Computing, Boston, 1988, pp.117-158.

[17] T. P. Hong and J. B. Chen, "Finding relevant attributes and membership functions," *Fuzzy Sets and Systems,* Vol.103, No. 3, 1999, pp. 389-404.

[18] T. P. Hong and J. B. Chen, "Processing individual fuzzy attributes for fuzzy rule induction," accepted and to appear in *Fuzzy Sets and Systems.*

[19] T. P. Hong and C. Y. Lee, "Induction of fuzzy rules and membership functions from training examples," *Fuzzy Sets and Systems***,** Vol. 84, 1996, pp. 33-47.

[20] T. P. Hong and S. S. Tseng, "A generalized version space learning algorithm for noisy and uncertain data," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 9, No. 2, 1997, pp. 336-340.

[21] R. H. Hou, T. P. Hong, S. S. Tseng and S. Y. Kuo, "A new probabilistic induction method, " *Journal of Automatic Reasoning*, Vol. 18, 1997, pp. 5-24.

[22] A. Kandel, *Fuzzy Expert Systems*, CRC Press, Boca Raton, 1992, pp. 8-19.

[23] C. M. Kuok, A. W. C. Fu and M. H. Wong, "Mining fuzzy association rules in databases," *The ACM SIGMOD Record*, Vol. 27, No. 1, 1998, pp. 41-46.

[24] E. H. Mamdani, "Applications of fuzzy algorithms for control of simple dynamic plants, " *IEEE Proceedings*, 1974, pp. 1585-1588.

[25] H. Mannila, "Methods and problems in data mining," *The International*

Conference on Database Theory, 1997.

[26] J. R. Quinlan, “Decision tree as probabilistic classifier,” *The Fourth International Machine Learning Workshop,* Morgan Kaufmann, San Mateo, CA, 1987, pp. 31-37.

[27] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA, 1993.

[28] R. Rastogi and K. Shim, "Mining optimized association rules with categorical and numeric attributes," *The 14th IEEE International Conference on Data Engineering*, Orlando, 1998, pp. 503-512.

[29] R. Rastogi and K. Shim, "Mining optimized support rules for numeric attributes," *The 15th IEEE International Conference on Data Engineering,* Sydney, Australia, 1999, pp. 206-215.

[30] J. Rives, “FID3: fuzzy induction decision tree,” *The First International symposium on Uncertainty, Modeling and Analysis*, 1990, pp. 457-462.

[31] R. Srikant and R. Agrawal, “Mining quantitative association rules in large relational tables,” *The 1996 ACM SIGMOD International Conference on Management of Data*, Monreal, Canada, June 1996, pp. 1-12.

[32] C. H. Wang, T. P. Hong and S. S. Tseng, “Inductive learning from fuzzy examples,” *The fifth IEEE International Conference on Fuzzy Systems*, New

Orleans, 1996, pp. 13-18.

[33] C. H. Wang, J. F. Liu, T. P. Hong and S. S. Tseng, “A fuzzy inductive learning strategy for modular rules,” *Fuzzy Sets and Systems*, Vol.103, No. 1, 1999, pp. 91-105.

[34] R.Weber, “Fuzzy-ID3: a class of methods for automatic knowledge acquisition,” *The second International Conference on Fuzzy Logic and Neural Networks*, Iizuka, Japan, 1992, pp. 265-268.

[35] Y. Yuan and M. J. Shaw, “Induction of fuzzy decision trees,” *Fuzzy Sets and Systems,* 69, 1995, pp. 125-139.

[36] L. A. Zadeh, “Fuzzy logic,” *IEEE Computer*, 1988, pp. 83-93.

[37] L. A. Zadeh, “Fuzzy sets,” *Information and Control*, Vol. 8, No. 3, 1965, pp. 338-353.