International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems
Vol. 9, No. 5 (2001) 587–604

TRADE-OFF BETWEEN COMPUTATION TIME AND NUMBER OF RULES FOR FUZZY MINING FROM QUANTITATIVE DATA*

TZUNG-PEI HONG†
Department of Information Management, I-Shou University, Kaohsiung, 84008, Taiwan, R.O.C.
E-mail: tphong@isu.edu.tw

CHAN-SHENG KUO
Department of Management Information Systems, National Chengchi University, Taipei, 11623, Taiwan, R.O.C.
E-mail: chansheng@mis.nccu.edu.tw

SHENG-CHAI CHI
Department of Industrial Management, Huafan University, Taipei, 223, Taiwan, R.O.C.
E-mail: scchi@huafan.hfu.edu.tw

Received August 1999
Revised February 2001

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. Transactions with quantitative values are however commonly seen in real-world applications. We proposed a fuzzy mining algorithm by which each attribute used only the linguistic term with the maximum cardinality in the mining process. The number of items was thus the same as that of the original attributes, making the processing time reduced. The fuzzy association rules derived in this way are not complete. This paper thus modifies it and proposes a new fuzzy data-mining algorithm for extracting interesting knowledge from transactions stored as quantitative values. The proposed algorithm can derive a more complete set of rules but with more computation time than the method proposed. Trade-off thus exists between the computation time and the completeness of rules. Choosing an appropriate learning method thus depends on the requirement of the application domains.

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

1. Introduction

Most enterprises have databases that contain a wealth of potentially accessible information. The unlimited growth of data however inevitably leads to a situation in which accessing desired information from a database becomes difficult. Knowledge discovery in databases (KDD) has thus 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,

*This is a modified and expanded version of the paper "A fuzzy data mining algorithm for quantitative values," presented at The Third International Conference on Knowledge-Based Intelligent Information Engineering Systems, 1999.

†Corresponding author; also at Department of Electrical Engineering, National University of Kaohsiung, Taiwan.

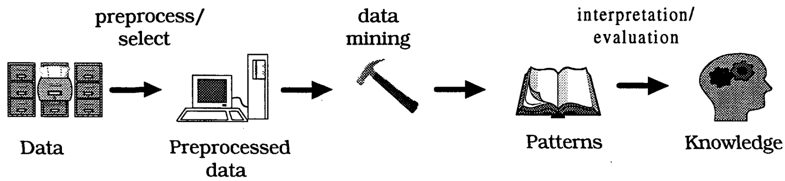

Figure 1: A KDD Process

potentially useful, and previously unknown patterns in large databases [11]. The KDD process [21] is shown in Figure 1

In Figure 1, data mining plays a critical role to the KDD process. It involves applying specific algorithms for extracting patterns or rules from data sets in a particular representation. Due to its importance, 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. For example, there may exist some implicitly useful knowledge in a large database containing millions of records of customers' purchase orders over the last five years. The knowledge can be found out using appropriate data-mining approaches. Questions such as "what are the most important trends in customers' purchase behavior?" can thus be easily answered.

Data mining is most commonly used in attempts to induce association rules from transaction data. An association rule is an expression $X \rightarrow Y$, where X is a set of items and Y is a single item. It means in the set of transactions, if all the items in X exist in a transaction, then Y is also in the transaction with a high probability. For example, assume whenever customers in a supermarket buy bread and butter, they will also buy milk. From the transactions kept in the supermarkets, an association rule such as "*Bread* and *Butter* $\rightarrow$ *Milk*" will be mined out. Most previous studies focused on binary valued transaction data. Transaction data in real-world applications, however, usually consist of quantitative values. 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 [20]. The theory has been applied in fields such as manufacturing, engineering, diagnosis, economics, among others [13, 20, 21, 29]. Several fuzzy learning algorithms for inducing rules from given sets of data have been designed and used to good effect with specific domains [5–7, 10, 12, 14–16, 18-19, 25, 27]. Strategies based on decision trees [8] were proposed in [9, 23–25, 28]. Wang et al. also proposed a fuzzy version space learning strategy for managing vague information [27].

In [17], we proposed a mining approach that integrated fuzzy-set concepts with the *apriori* mining algorithm [4] to find interesting itemsets and fuzzy association rules in transaction data with quantitative values. The term "itemset" was first proposed by Agrawal et al. in their papers [1–4] on data mining, and from then becomes a common usage in this field. It means a set composed of items. Our previously proposed algorithm first used membership functions to transform each quantitative value into a fuzzy set in linguistic terms. It then calculated the scalar cardinalities of all linguistic terms in the transaction data. Each attribute used only the linguistic term with the maximum cardinality in the mining process, thus keeping the number of items the same as that of the original attributes. The algorithm therefore focused on the most important linguistic terms to reduce its computation time. Although this approach could quickly find interesting patterns, some patterns might however be missed since only the linguistic term with the maximum cardinality in each attribute was used in the mining process. This paper thus modifies it and proposes a new fuzzy data-mining algorithm for extracting interesting knowledge from transactions stored as quantitative values. The proposed algorithm considers all the important linguistic terms in the mining process and can thus derive a more complete set of rules than the method proposed in [17] although its computation time increases. Trade-off exists between the computation time and the completeness of rules. Choosing an appropriate learning method thus depends on the requirement of the application domains.

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 flow charts of both the proposed and the previous methods are shown in Section 4 as a comparison. The proposed data-mining algorithm for quantitative values is described in details in Section 5. An example is given to illustrate the proposed algorithm in Section 6. Experiments to demonstrate the performance of the proposed data-mining algorithm are stated in Section 7. Conclusions and proposal of future work are given in Section 8.

2. Review of Agrawal et al.'s Data-Mining Algorithms

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 [26] for mining association rules from those with quantitative attributes. Their proposed method first determined the number of partitions for each quantitative attribute, and then mapped all possible values of each attribute into a set of consecutive integers. It then found large itemsets whose support values were greater than the user-specified minimum-support levels. These large itemsets were then processed to generate association rules, and rules of interest to users were output.

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 in 1965 [30]. 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* [30]. 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:

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

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,

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

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

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

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

(ii) 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:

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

(iii) 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:

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

The above fuzzy operations will be used in the proposed mining algorithm to find linguistic association rules.

4. The Flow Charts of the Proposed and the Previous Methods

The flow charts of the fuzzy data mining approaches proposed in [17] and in this paper are respectively shown in Figures 2 and 3. The difference in these two figures is distinguished by the grey regions. In Figure 2, only the linguistic terms with the maximum count for each attribute are used to form the candidate set. In Figure 3, all the linguistic terms are used. Linguistic terms belonging to the same attribute cannot, however, belong to the same itemset. The computation in Figure 3 is more complex than that in Figure 2 since all possible linguistic terms are used in calculating the large itemsets, but the derived set of association rules in Figure 3 is more complete than that in Figure 2.

5. The Fuzzy Data-Mining Algorithm for Quantitative Values

Notation used in this paper is first stated as follows.

n: the total number of transaction data;
m: the total number of attributes;
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*,* $1 \le k \le |A_j|$*;*
$D^{(i)}$*: the i-th transaction datum,* $1 \le i \le n$*;*
$v_j^{(i)}$*: the quantitative value of* A_j *for* $D^{(i)}$*;*
$f_j^{(i)}$*: 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* $f_{jk}^{(i)}$ *for i=1 to n;*
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 transforms each quantitative value into a fuzzy set of linguistic terms using membership functions. The algorithm then calculates the scalar cardinality of each linguistic term on all the transaction data. The mining process based on fuzzy counts is then performed to find fuzzy association rules. The

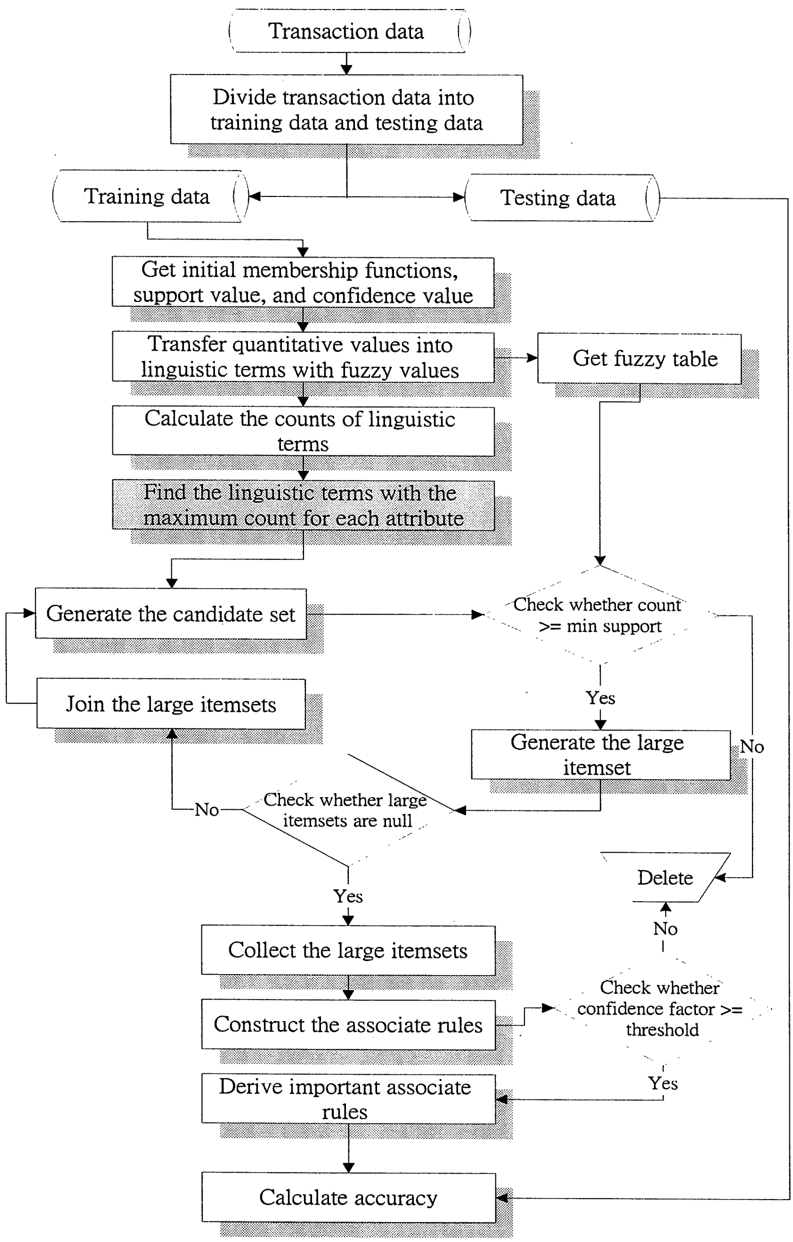

Figure 2: The flow chart of the previously proposed approach

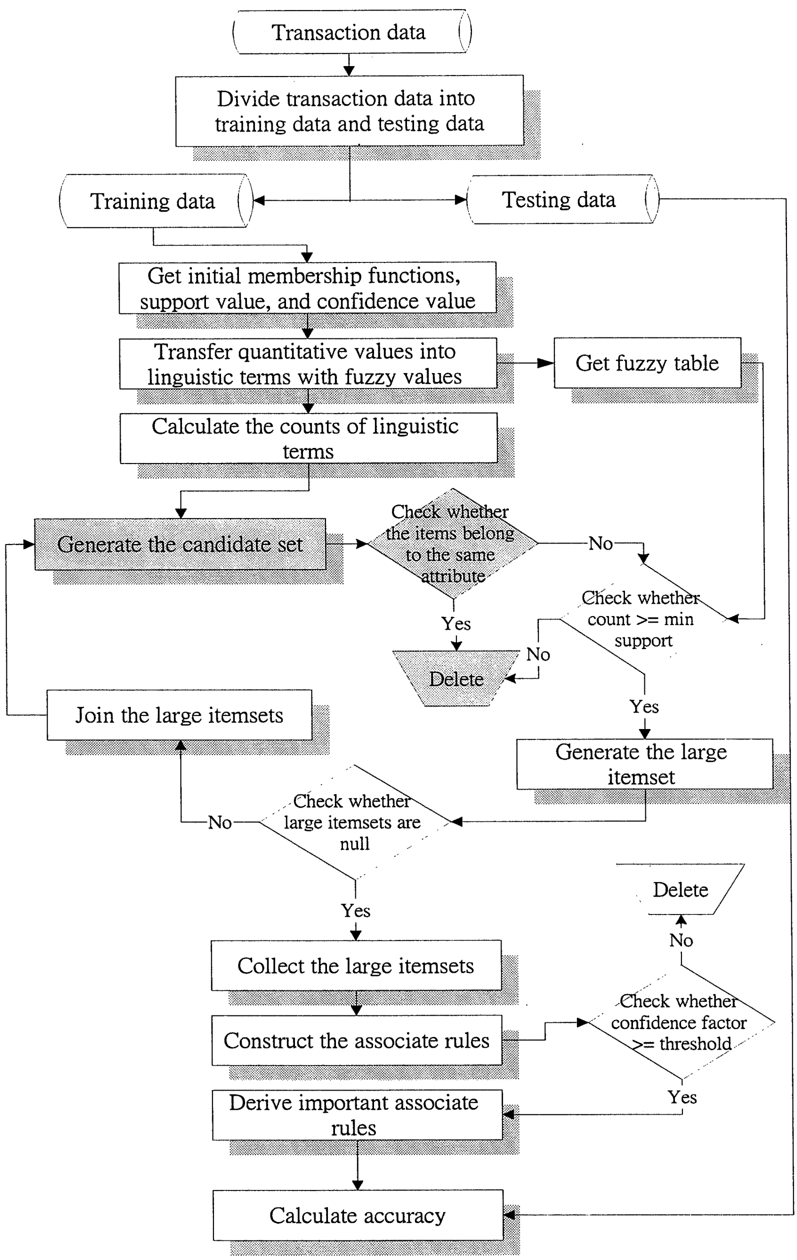

Figure 3: The flow chart of the currently proposed approach

details of the proposed mining algorithm are described as follows. Here we assume the training data are directly fed into the proposed mining algorithm.

The Fuzzy Data Mining Algorithm:

INPUT: A set of n training 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 $\left(\frac{f_{j1}^{(i)}}{R_{j1}} + \frac{f_{j2}^{(i)}}{R_{j2}} + ... + \frac{f_{jl}^{(i)}}{R_{jl}}\right)$ using the given membership functions, where R_{jk} is the k-th fuzzy region (linguistic term) 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 count of each attribute region (linguistic term) R_{jk} in the transaction data:

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

STEP 3: Collect each attribute region (linguistic term) to form the candidate set C_1.

STEP 4: Check whether $count_{jk}$ of each R_{jk} $(1 \le j \le m$ and $1 \le k \le |A_j|)$ is larger than or equal to the predefined minimum support value α. If R_{jk} satisfies the above condition, put it in the set of large 1-itemsets (L_1). That is:

$$L_1 = \{ R_{jk} \mid count_{jk} \ge \alpha, 1 \le j \le m \text{ and } 1 \le k \le |A_j| \}.$$

STEP 5: IF L_1 is not null, then do the next step; otherwise, exit the algorithm.

STEP 6: Set r=1, where r is used to represent the number of items kept in the current large itemsets.

STEP 7: Join the large itemsets L_r to generate the candidate set C_{r+1} in a way similar to that in the apriori algorithm [4] except that two regions (linguistic terms) belonging to the same attribute cannot simultaneously exist in an itemset in C_{r+1}. Restated, the algorithm first joins L_r and L_r under the condition 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 and do not have any two items R_{jp} and R_{jq} $(p \neq q)$ of the same attribute R_j.

STEP 8: 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 $f_s^{(i)} = f_{s_1}^{(i)} \Lambda f_{s_2}^{(i)} \Lambda ... \Lambda f_{s_{r+1}}^{(i)}$, 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 count of s in the transactions 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 9: IF L_{r+1} is null, then do the next step; otherwise, set $r=r+1$ and repeat STEPs 6 to 8.

STEP 10: Collect the large itemsets together.

STEP 11: Construct association rules for each large q-itemset s with items $(s_1, s_2, ..., s_q)$, $q \geq 2$, using the following substeps:

(a) Form each possible association rule 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 12: Output the association rules with confidence values larger than or equal to the predefined confidence threshold λ.

After STEP 12, the rules output can serve as meta-knowledge concerning the given transactions. If testing data exist, they are then used to verify the accuracy of the association rules mined.

6. 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, for simplification, are 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

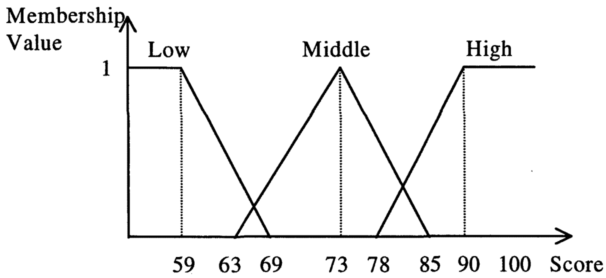

Figure 4: The membership functions used in this example

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 shown in Figure 4.

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. IF the transaction data in Table 1 are all used as the training data, the proposed mining algorithm proceeds as follows.

STEP 1: Transform the quantitative values of each transaction datum into a fuzzy set. 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.

STEP 2: Calculate the scalar cardinality of each attribute region (linguistic term) 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: Each attribute region (linguistic term) is a candidate 1-itemset.

STEP 4: For each region (linguistic term), check whether its count is larger than or equal to the predefined minimum support value α . Assume in this example, α is set at 2. Since the count values of OOP.Middle, OOP.High, DB.High, ST.High, DS.Middle, DS.High, MIS.Middle and MIS.High are all larger than 2, these items are put in L_1 (Table 3).

STEP 5: Since L_1 is not null, the next step is then done.

STEP 6: Set r=1.

STEP 7: Join L_r to generate the candidate set C_{r+1}. C_2 is first generated from L_1 as follows: (OOP.Middle, DB.High), (OOP.Middle, ST.High), …, (DS.High, MIS.High). Note that the itemset (OOP.Middle, OOP.High) is not kept in C_2 since both these two items belong to the same course OOP.

STEP 8: 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 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	2.4	3.8	0.9	1.6	4.6	0.8	3.9	2.4	0.8	2.3	4.0

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

Itemset	support
OOP.Middle	3.8
OOP.High	2.3
DB.High	3.8
ST.High	4.6
DS.Middle	3.9
DS.High	2.4
MIS.Middle	2.3
MIS.High	4.0

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

Case	OOP.Middle	DB.High	OOP.Middle ∩ 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.

(c) Check whether these counts are larger than or equal to the predefined minimum support value 2. Four itemsets, including (OOP.Middle, DB.High), (DB.High, ST.High), (DB.High, DS.Middle) and (ST.High, DS.Middle) , are thus kept in L_2 (Table 5).

Table 5: The itemsets and their fuzzy counts in L_2

OOP.Middle DB.High	2.2
DB.High ST.High	2.2
DB.High DS.Middle	2.7
ST.High DS.Middle	2.8

STEP 9: 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 three itemsets (OOP.Middle DB.High ST.High), (OOP.Middle DB.High DS.Middle) (DB.High ST.High DS.Middle) are formed. Since all their counts are smaller than 2, they are not put in L_3. L_3 is thus an empty set. STEP 10 then begins.

STEP 10: Collect the large itemsets together. Here only L_1 and L_2 exist.

STEP 11: Construct association rules for each large 2-itemset using the following substeps.

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

If OOP.Middle then DB.High;
If DB.High then OOP.Middle;
If DB.High then ST.High;
If ST.High then DB.High;
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 fifth association rule as an example. The fuzzy count of DB.High $\cap$ DS.Middle is calculated as shown in Table 6.

Table 6: The fuzzy count of DB.High $\cap$ DS.Middle

Case	DB.High	DS.Middle	DB.High $\cap$ 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 rules can be similarly derived.

STEP 12: 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:

(i) If the Database score is high, then the Data Structure score is middle, with a confidence factor of 0.71;

(ii) 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.

7. Experimental Results

Part of the customer purchase data from a supermarket of a department store in Kaohsiung, Taiwan, were used to show the feasibility of the proposed mining algorithm. A total of 1508 transactions were included in the data set. Each transaction recorded the purchasing information of a customer. Execution of the mining algorithm was performed on a Pentium-PC. The relationships between numbers of large itemsets and minimum support values for λ =0.3 are shown in Figure 5.

From Figure 5, 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. Also, appropriate minimum support values can avoid too many large itemsets and uninteresting patterns.

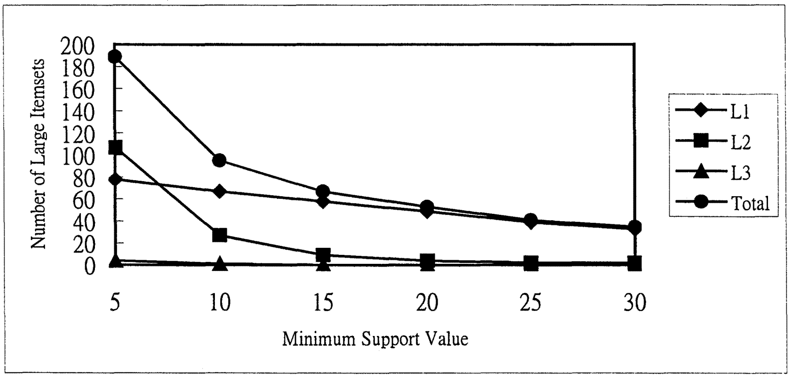

Figure 5. The relationship between numbers of large itemsets and minimum support values using the proposed method.

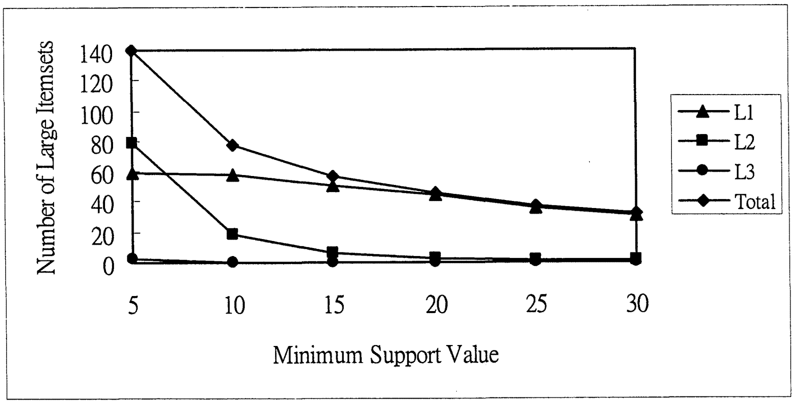

Figure 6: The relationship between numbers of large itemsets and minimum support values using the previous method.

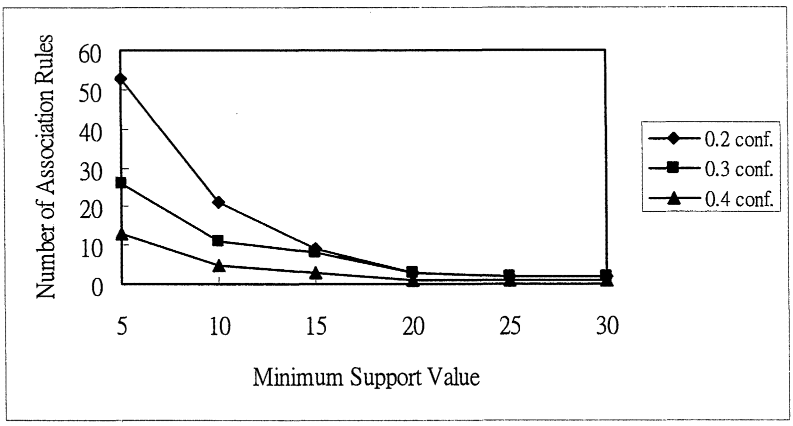

Figure 7. The relationship between numbers of association rules and minimum support values.

The relationships between numbers of large itemsets and minimum support values using the method in [17] are also shown in Figure 6 for comparison. It is obvious from Figures 5 and 6 that the numbers of itemsets by the newly proposed method are larger than those by the method in [17]. Our method can thus find more fuzzy association rules although it needs more computation time than the previous one.

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 7.

From Figure 7, 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 curves for larger minimum confidence values were smoother than

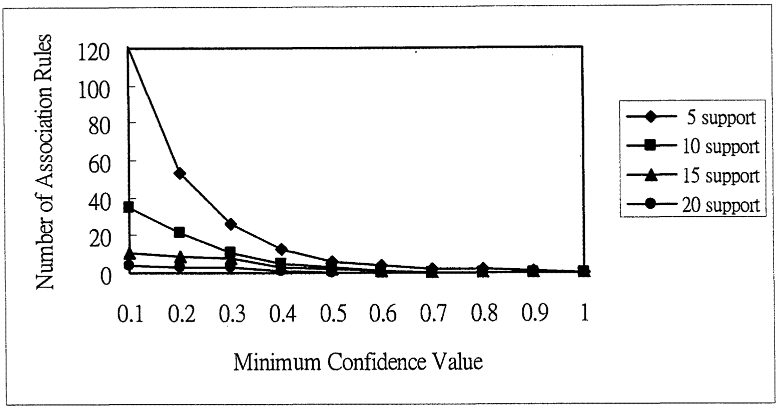

Figure 8. The relationship between numbers of association rules and minimum confidence values.

those for smaller minimum confidence values, meaning that the minimum support value had a large effect on the numbers 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 8.

From Figure 8, it is easily seen that the numbers of association rules decreased along with an increase in minimum confidence values. This is quite consistent with our intuition. The curves for larger minimum support values were smoother than those 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 then made to measure the accuracy of the proposed mining algorithm. The data set was first split into a training set and a test set, and the fuzzy mining algorithm was run on the training set to induce the rules. The rules were then tested on the test set to measure the percentage of correct predictions. In each run, 754 cases were selected at random for training and the remaining 754 cases were used for testing. Results for different minimum support values and confidence values are shown in Figure 9.

From Figure 9, it is easily seen that the mining algorithm run at a higher minimum confidence value had a higher accuracy since the minimum confidence value could be thought of as an accuracy threshold for deriving rules. The average accuracy of the rules was also higher for a larger minimum confidence value.

Experiments were finally made to compare the accuracy of the proposed fuzzy mining algorithm, the previous fuzzy mining algorithm, and the crisp-partition mining method in which the possible values of each attribute were partitioned in a crisp fashion and a traditional mining algorithm was used to mine association rules. The comparison of accuracy for the minimum support value set at 10 is shown in Figure 10.

From Figure 10, it is easily seen that the accuracy of the proposed fuzzy mining algorithm was higher than that of the crisp partition method and that of the previous method in [17] for various minimum confidence values.

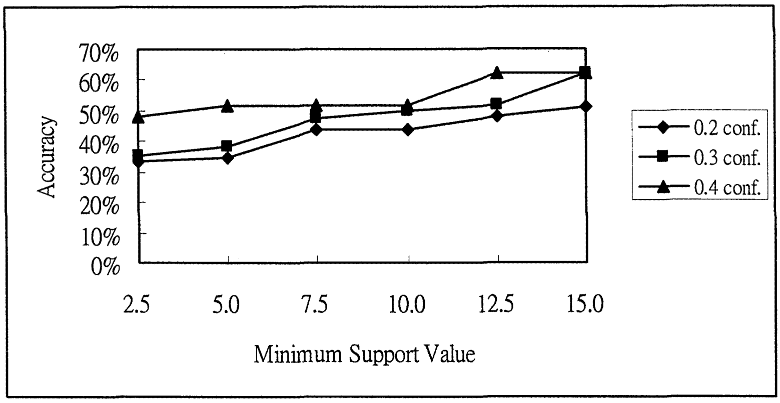

Figure 9: The relationship between accuracy and minimum support values for various confidence values.

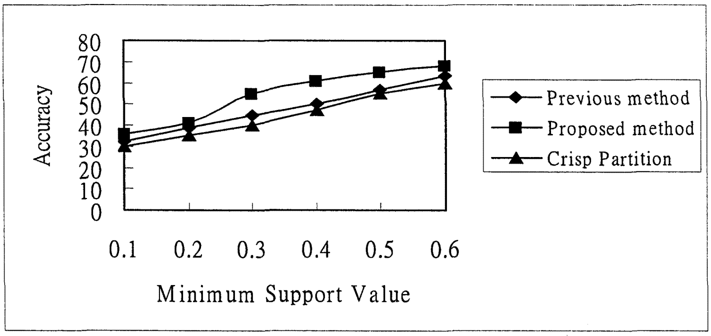

Figure 10: The comparison of the accuracy of three mining algorithms

7. Conclusion and Future Work

In this paper, we have proposed a generalized data-mining algorithm, which can process transaction data with quantitative values and discover interesting patterns among them. The proposed algorithm can derive a more complete set of rules than the method proposed in [17] although it needs more computation time. Trade-off thus exists between the computation time and the completeness of rules. Choosing an appropriate learning method thus depends on the requirement of the application domains. The proposed algorithm can also solve conventional transaction-data problems by using degraded membership functions. Experimental results with the data in a supermarket of a department store show the feasibility of the proposed mining algorithm.

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 [14, 16, 18], 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.

Acknowledgments

The authors would like to thank the anonymous referees for their very constructive comments.

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," T*he International Conference on Very Large Data Bases*, 1994, pp. 487–499.
5. 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. R. L. P. Chang and T. Pavliddis, "Fuzzy decision tree algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 7, 1977, pp. 28–35.
8. 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.
9. P. Clark and T. Niblett, "The CN2 induction algorithm," *Machine Learning*, Vol. 3, 1989, pp. 261–283.
10. M. Delgado and A. Gonzalez, "An inductive learning procedure to identify fuzzy systems," *Fuzzy Sets and Systems*, Vol. 55, 1993, pp. 121–132.
11. 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.
12. A.Gonzalez, "A learning methodology in uncertain and imprecise environments," *International Journal of Intelligent Systems*, Vol. 10, 1995, pp. 57–371.
13. Graham and P. L. Jones, *Expert Systems – Knowledge, Uncertainty and Decision,* Chapman and Computing, Boston, 1988, pp.117–158.
14. 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.
15. T. P. Hong and J. B. Chen, "Processing individual fuzzy attributes for fuzzy rule induction," *Fuzzy Sets and Systems*, Vol. 112, No. 1, 2000, pp. 127–140.
16. 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.
17. T. P. Hong, C. S. Kuo, and S. C. Chi, "Mining association rules from quantitative data," *Intelligent Data Analysis,* Vol. 3, No. 5, 1999, pp. 363–376.
18. 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.
19. 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.
20. Kandel, *Fuzzy Expert Systems*, CRC Press, Boca Raton, 1992, pp.8–19.
21. E. H. Mamdani, "Applications of fuzzy algorithms for control of simple dynamic plants," *IEEE Proceedings*, 1974, pp.1585-1588.
22. H. Mannila, "Methods and problems in data mining," *The International Conference on Database Theory*, 1997.

23. J. R. Quinlan, "Decision tree as probabilistic classifier," *The Fourth International Machine Learning Workshop, Morgan Kaufmann, San Mateo*, CA, 1987, pp. 31–37.
24. J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA, 1993.
25. J. Rives, "FID3: fuzzy induction decision tree," *The First International symposium on Uncertainty, Modeling and Analysis*, 1990, pp. 457–462.
26. 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.
27. 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.
28. 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.
29. 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.
30. H. J. Zimmermann, *Fuzzy Set Theory and Its Applications,* Kluwer Academic Publisher, Boston,1991.