Journal of Nonlinear and Convex Analysis
Volume 14, Number 1, 2013, 193–207

Yokohama Publishers
ISSN 1880-5221 ONLINE JOURNAL

FUZZY DATA MINING FOR QUANTITATIVE TRANSACTIONS WITH FP-GROWTH

CHIEN-HUA WANG, WEI-HSUAN LEE, AND CHIN-TZONG PANG*

Abstract. Association rules mining is to find associations efficiently among the different items of a transaction database. In order to help decision-makers conduct sound and timely solutions, we apply the grid partition method to decide a membership function of quantitative value for each transaction item. We then propose a fuzzy FP-growth algorithm to deal with the process of data mining. In addition, different thresholds are utilized to intervene the experiment. The result shows that the fuzzy FP-growth algorithm is more efficient than others.

1. Introduction

Data mining is useful information retrieval for efficient problem solving in enterprises. By effectively helping decision-makers retrieve desirable knowledge, it has become a predominant issue in recent years [14]. That data mining is widely adopted is to induce association rules ($X \to Y$) from transaction data, where one existing (X) appears, other items (Y) are likely to appear as well [6]. For instance, when a customer purchases bread, one might also get milk along with it. Accordingly, association rules can assist decision makers to scoop the possible items that are likely to be purchased together by consumers in the hopes to facilitate planning marketing strategies [4].

In the conventional association rule algorithm, scanning database takes enormous time particularly when one uses the Apriori algorithm. It occurs that it usually affects the efficiency in data mining. To solve the drawback aforementioned, Han *et al.* [7] proposed a mining method, called frequent-pattern growth (FP-growth). FP-growth has no need to generate candidate itemsets and is considered to be more efficient. FP-growth is constructed by reading the data set one transaction at a time and mapping each transaction onto a path in a frequent-pattern tree (FP-tree). Since different transactions can have several identical items at the same time, their paths may overlap with each other. The more paths overlap with one another, the more compression we can achieve by using the FP-tree structure. If the size of the FP-tree is small enough to fit into the main memory, we can extract frequent itemsets directly from the structure in memory instead of making repeated passes over the data on disks [16]. Therefore, without generating candidate itemsets, one only needs is to scan the database twice.

In addition, in terms of decision making, one has to take users' perception and cognitive uncertainty of subjective decisions into consideration. Zadeh [17] proposed

2010 *Mathematics Subject Classification.* 68Q87.

Key words and phrases. Data Mining, association rules, FP-growth algorithm.

Corresponding author. This work was supported in part by the National Science Council of the Republic of China.

the fuzzy set theory to deal with cognitive uncertainty of vagueness and ambiguity which has been widely used in many applications and has good effects. Thus, it is combined with many methods to generate new learning algorithms. For example, Lin *et al.* [13] proposed a fuzzy mining algorithm which used compressed fuzzy frequent pattern tree (CFFP tree) to extract and analyze quantitative data. And Papadimitriou and Mavroudi [15] proposed a learning algorithm which applied fuzzy frequent-pattern tree (FFP-tree) to find fuzzy association rules. Their method was based on FP-tree using only the local frequent fuzzy 1-itemsets kept in each transaction for mining. The fuzzy grids which were close to but below the predefined minimum support threshold would make no contributions to the mining. It turned out it lost the purpose of incorporating fuzzy set theory processing.

This paper proposes a fuzzy frequent-pattern growth (FFP-growth) method, which treats each item from a transaction database as a linguistic variable, and each linguistic variable is partitioned based on its linguistic value. By doing this, the natural language can be utilized to fully explain fuzzy association rules. There are two phases in the proposed method. One is to find frequent 1-itemset by scanning the database once, and the other is to establish a membership function of FP-tree (MFFP-tree) by scanning the database twice. Next, one conditional pattern base and one conditional membership function of FP-tree (CMFFP-tree) will be extracted from each node in a membership function of FP-tree to generate the fuzzy association rules.

The remaining parts of this paper are organized as below: The association rule and grid partition method are briefly introduced in Section 2. Notations and the algorithm are introduced in Section 3. An example is given to illustrate the proposed algorithm in Section 4. Experimental results to demonstrate the performance of the proposed fuzzy data mining algorithm and comparison with other methods are stated in Section 5. And a conclusion is given in Section 6.

2. Review of mining association rules and grid partition method

The goal of data mining is to explore, analyze knowledge, and to discover meaningful patterns [4]. Agrawal and Srikant has proposed the Apriori algorithm to find association rules in transaction data. They divide the mining process into two major subtasks: frequent itemset generation and rule generation [1, 2, 3], and integrate fuzzy set concept with the Apriori algorithm. Chan *et al.* proposes an F-APACS algorithm to mine fuzzy association rules [5]. Kuok *et al.* proposes a fuzzy mining method to find fuzzy association rules in a numerical database [12]. Hong *et al.* further uses a fuzzy mining algorithm to mine fuzzy rules for quantitative data [8]. Hu *et al.* then utilizes this concept to propose a learning algorithm to mine association and classification rules [9, 10, 11]. In spite of its popularity, the Apriori algorithm may suffer from two nontrivial costs in which it may need to generate a huge number of candidate sets, and may need repeatedly scan a database and check a large set of candidates by pattern matching. By using the "Without-the-candidate-generation" method proposed by Han *et al.*, FP-growth, adopts a divide-and-conquer strategy as follows: first, compress the database representing frequent items into a frequent-pattern tree (FP-tree), but retain the itemset association information. Then divide such a compressed database into a set of conditional

databases, each associated with one frequent item. Lastly, mine each such database separately [7].

In practice, the FP-growth method transforms the problem of finding long frequent patterns to looking for shorter ones recursively and then concatenating the suffix. It uses the least frequent items as suffixes and offers good selectivity. The method substantially reduces the search costs. When the data are large, it is sometimes unrealistic to construct a main memory-based FP-tree. An interesting alternative is to first partition the database into a set of projected databases, and then construct an FP-tree and mine it in each projected database. A study on the performance of the FP-growth method shows that it is efficient and scalable for mining both long and short frequent patterns, and is about an order of magnitude faster than that of the Apriori algorithm. Thus, this paper adopts FP-growth and grid partition methods to mine fuzzy association rules.

By dividing each linguistic variable [18, 19, 20] with its different linguistic values, $K_1 \times K_2 \times \cdots \times K_{d-1}$ fuzzy grids with d dimensions in the pattern space can be obtained. In particular, this paper views a fuzzy grid as a fuzzy concept. For example, there are nine fuzzy grids in Fig.1, since $K_1 = K_2 = 3$. The fuzzy grid (A_{11}, A_{23}) corresponds to the region depicted in Fig.1. In Fig.1, if (A_{11}, A_{23}) is a frequent fuzzy grid, then it is a useful fuzzy concept, and the degrees of relevance of each pattern with respect to (A_{11}, A_{23}) can be further computed.

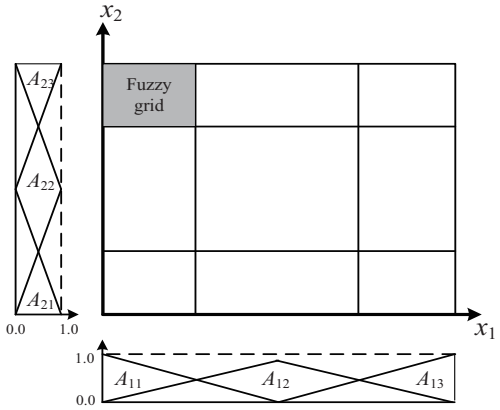

Fig.1. Fuzzy partitions for x_1 and x_2.

In this method, symmetric triangle-shaped linguistic values are used for simplicity. When a linguistic value is not yet determined if it is frequent, it is called a candidate 1-dim fuzzy grid so that a quantitative variable x_k can be divided into K partitions ($K = 2, 3, \cdots$). In addition, $A^K_{x_k,l_i}$ stands for a candidate 1-dim fuzzy grid, and $\mu^K_{x_k,l_i}(x)$ can be defined as the following:

$$\mu^K_{x_k,l_i}(x) = \max\{1 - |x - a^K_{l_i}|/b^K, 0\},$$

where $a^K_{l_i} = m_i + ((m_a - m_i)(i-1)/(K-1)), b^K = (m_a - m_i)/(K-1)$, and m_a and m_i are the maximum and minimum values of the domain interval of x_k; l_i is the ith linguistic value of K linguistic values defined in linguistic variable x_k, respectively. (A_{11}, A_{23}) is called a candidate 2-dim fuzzy grid that can be generated by using A_{11}

and A_{23}. In other words, candidate 1-dim fuzzy grids can be further employed to generate the other candidate or frequent fuzzy grids with higher dimensions [11].

3. Notations and the proposed algorithm

The proposed construction algorithm for building a membership function FP-tree (MFFP-tree) from the quantitative database is described in this section. The notations used in the proposed algorithm is firstly stated below.

Notation

n : the number of transaction data;

m : number of items used to describe each transaction data, where $1 \leq m$;

x_k : kth item, where $1 \leq k \leq m$;

K : the number of linguistic values in each quantitative item of transaction database, where $K \geq 2$;

t_p : pth transaction data, where $1 \leq p \leq n$;

$A^K_{x_k,l_i}$: ith linguistic value of K linguistic values defined in linguistic variable x_k, where $1 \leq k \leq m, 1 \leq i \leq K$;

$\mu^K_{x_k,l_i}(\cdot)$: the membership function of $A^K_{x_k,l_i}$;

$q^{x_k}_{t_p}$: the quantitative value of the item x_k for pth transaction data;

$count^K_{x_k,l_i}$: the summation of the values $\mu^K_{x_k,l_i}(q^{x_k}_{t_p})$ for $p = 1, 2, \cdots n$;

$count^{max}_{x_k}$: the maximum values of $count^K_{x_k,l_i}$ for $i = 1, 2, \cdots K$;

$G^{max}_{x_k}$: the fuzzy grid of x_k with $count^{max}_{x_k}$;

α : the user-specified minimum fuzzy support (min FS) value;

β : the user-specified minimum fuzzy confidence (min FC) value;

L_r : the set of frequent patterns with r length.

The proposed algorithm:

INPUT: A body of n training data, each linguistic variable with K linguistic values, a user-specified minimum fuzzy support (min FS) value α and a user-specified minimum fuzzy confidence (min FC) value β.

OUTPUT: A set of fuzzy association rules.

Step 1. Using grid partition method to transform the quantitative item $A^K_{x_k,l_i}$ for each item x_k of each transaction datum t_p $(p = 1, 2, \ldots, n)$ into a fuzzy grid $A^K_{x_k,l_i}$ represented as:

$$\left(\frac{\mu^K_{x_k,l_1}(q^{x_k}_{t_p})}{A^K_{x_k,l_1}} + \frac{\mu^K_{x_k,l_2}(q^{x_k}_{t_p})}{A^K_{x_k,l_2}} + \cdots + \frac{\mu^K_{x_k,l_K}(q^{x_k}_{t_p})}{A^K_{x_k,l_K}}\right)$$

using the given membership functions $\mu^K_{x_k,l_i}(\cdot)$, where $A^K_{x_k,l_i}$ is the ith fuzzy grid (linguistic term) of K linguistic values defined in linguistic variable x_k of t_pth transaction data.

Step 2. Scan database and calculate the scalar cardinality of fuzzy grid $A^K_{x_k,l_i}$ for each item x_k in the training data: $count^K_{x_k,l_i} = \sum_{p=1}^{n} \mu^K_{x_k,l_i}(q^{x_k}_{t_p})$.

Step 3. Search $count^{max}_{x_k}$, where $count^{max}_{x_k} = \max_{1\le i\le K} count^K_{x_k,l_i}$. Let $G^{max}_{x_k}$ be the fuzzy grid with $count^{max}_{x_k}$ for item x_k. $G^{max}_{x_k}$ will be used to represent this attribute in later mining processing.

Step 4. Collect the fuzzy grid $G^{max}_{x_k}$ of all item x_k and check whether each $G^{max}_{x_k}$ is larger than or equal to the predefined minimum support value. If $G^{max}_{x_k}$ satisfies this condition, put it in the set of L_1, where $L_1 = \{G^{max}_{x_k} : count^{max}_{x_k} \ge \alpha, 1 \le k \le n\}$.

Step 5. If L_1 is not null, then proceed with the next step. Otherwise, exit the algorithm.

Step 6. This step begins to proceed with FP-Growth. Use patterns of L_1 and establish a descending data table called Header Table.

Step 7. According to Header Table, rebuild new fuzzy set table which is sorted by the original fuzzy sets table.

Step 8. Initially set the root node of the fuzzy FP-tree as {ROOT}, and then scan the database to build the membership function FP-tree tuple by tuple which is based on each 1-dim fuzzy grid in the new fuzzy sets table, and link the node of fuzzy grids with the tree.

Step 9. Mine the patterns of Header Table ascendingly. Then set up the conditional pattern base of each node in a membership function of the FP-tree. Next, build the conditional membership function FP-tree.

Step 10. Repeatedly mine conditional membership function of the FP-tree, and gradually increase the frequency pattern base. If the conditional membership function of the FP-tree contains one path, all patterns can be listed.

Step 11. The following substeps are done for corresponding frequent patterns.

(a) Calculate the fuzzy value $N^{t_p}_s$ in pattern $s : (s_1, s_2, \ldots, s_{r+1})$ defined as

$$N^{t_p}_s = \mu^K_{x_{i_1},l_{i_1}}(q^{x_{i_1}}_{t_p}) \wedge \mu^K_{x_{i_2},l_{i_2}}(q^{x_{i_2}}_{t_p}) \wedge \cdots \wedge \mu^K_{x_{i_{r+1}},l_{i_{r+1}}}(q^{x_{i_{r+1}}}_{t_p}),$$

where $s_k = (x_{i_k}, l_{i_k})$, x_{i_k} is i_k th item, l_{i_k} is the i_k th linguistic value of K linguistic values defined in linguistic variable x_{i_k}, $\mu^K_{x_{i_k}l_{i_k}}(\cdot)$ is the membership function of $A^K_{x_{i_k},l_{i_k}}$ for $k = 1, 2, \ldots, r+1$ and for training data t_p.

(b) The fuzzy support $count_s$ of frequent pattern s is calculated, $count_s = \sum_{p=1}^{n} N_s^{t_p}$.

(c) If the fuzzy support $count_s$ is larger than or equal to the user-specified minimum support value, put the frequent pattern s in $L_r (r \geq 2)$.

Step 12. Construct effective association rules for each frequent pattern $(s_1, s_2, \cdots, s_l)$, $l \geq 2$ using the following.

(a) List all possible frequent patterns as follows: $s_1 \wedge s_2 \wedge \cdots \wedge s_k \longrightarrow s_k, k = 1, 2, \cdots, l$.

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

$$\frac{\sum_{p=1}^{n} N_s^{t_p}}{\sum_{p=1}^{n} (N_{s_1}^{t_p} \wedge N_{s_2}^{t_p} \wedge \cdots \wedge N_{s_l}^{t_p})}.$$

Step 13. Output the association rules with confidence values larger than or equal to the user-specified minimum fuzzy confidence value β.

4. An example

In this section, an example is given to illustrate the proposed fuzzy FP-growth algorithm. This example shows how the proposed algorithm can be used to generate fuzzy association rules from a set of transactions. The data set includes five transactions, as shown in Table 1. Each transaction is composed of a transaction identifier and items purchased. There are five items A, B, C, D and E to be purchased. Each item is represented by a tuple (item name, item amount). For instance, the first transaction consists of three units of A, eight units of B, four units of D and eleven units of E. In addition, the assumptions of the membership functions for the item quantities are shown in Fig.2.

Table 1: The data set used in this example

TID	ITEMS
1	(A,3), (B,8), (D,4), (E,11)
2	(B,7), (C,2), (D,3)
3	(A,2), (D,7), (E,6)
4	(A,3), (B,2), (E,5)
5	(B,5), (C,3), (D,8)

In this example, each attribute for x_k has three fuzzy grids: $Small$, $Middle$, and $Large$. Thus, three fuzzy membership values are produced for the quantity of each item according to the predefined membership functions. Additionally, assume that the predefined minimum fuzzy support value and minimum fuzzy confidence value are 1.2 and 0.55, respectively. For the transaction data in Table 1, the proposed mining algorithm proceeds as follows.

Step 1. The quantitative values of the items in each transaction are represented by fuzzy sets. Take the first item in Transaction 1 as an example. The amount "3" of A is used as the given membership function (Fig.2) and applied grid partition method

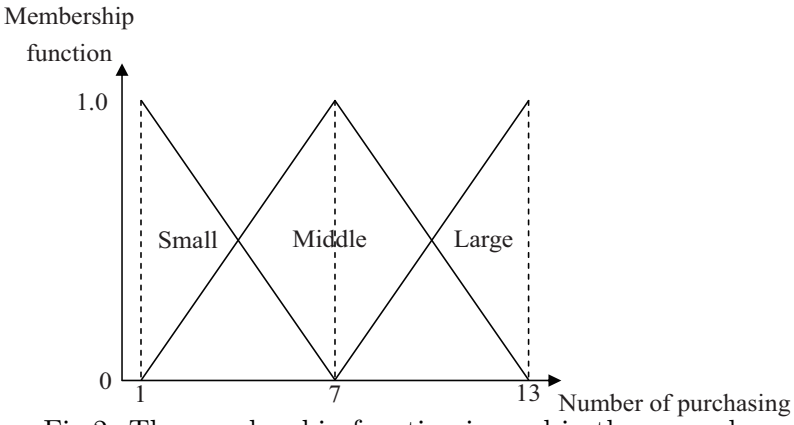

Fig.2. The membership function is used in the example.

to form the fuzzy set $(0.667/A.Small + 0.333/A.Middle)$. The step is repeated for the other items, and the results are shown in Table 2.

Step 2. Scanning database and the scalar cardinality of each fuzzy grid in the

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

TID	Fuzzy Sets
1	$(\frac{0.667}{A.Samll} + \frac{0.333}{A.Middle}), (\frac{0.833}{B.Middle} + \frac{0.167}{B.Large}), (\frac{0.5}{D.Samll} + \frac{0.5}{D.Middle})(\frac{0.333}{E.Middle} + \frac{0.667}{E.Large})$
2	$(\frac{1.0}{B.Middle}), (\frac{0.833}{C.Samll} + \frac{0.167}{C.Middle}), (\frac{0.667}{D.Samll} + \frac{0.333}{D.Middle})$
3	$(\frac{0.833}{A.Samll} + \frac{0.167}{A.Middle}), (\frac{1.0}{D.Middle}), (\frac{0.167}{E.Samll} + \frac{0.833}{E.Middle})$
4	$(\frac{0.667}{A.Samll} + \frac{0.333}{A.Middle}), (\frac{0.833}{B.Samll} + \frac{0.167}{B.Middle}), (\frac{0.333}{E.Samll} + \frac{0.667}{E.Middle})$
5	$(\frac{0.333}{B.Samll} + \frac{0.667}{B.Middle}), (\frac{0.667}{C.Samll} + \frac{0.333}{C.Middle}), (\frac{0.833}{D.Middle} + \frac{0.167}{D.Large})$

transactions are calculated as the count value. Take the fuzzy grid $A.Small$ as an example. Its scalar cardinality is $(0.667 + 0 + 0.833 + 0.667 + 0\ ) = 2.167$. The step is repeated for the other grids, and the results are shown in Table 3.

Table 3: The counts of the fuzzy grids

Item	Count	Item	Count	Item	Count
$A.Samll$	2.167	$A.Middle$	0.833	$A.Large$	0
$B.Samll$	1.167	$B.Middle$	2.667	$B.Large$	0.167
$C.Samll$	1.5	$C.Middle$	0.5	$C.Large$	0
$D.Samll$	1.167	$D.Middle$	2.667	$D.Large$	0.167
$E.Samll$	0.5	$E.Middle$	1.833	$E.Large$	0.667

Step 3. The fuzzy grid with the maximum count among the three possible grids for each item is found. Take item B as an example. Its count is 1.167 for $Small$, 2.667 for $Middle$, and 0.167 for $Large$. Since the count for $Middle$ is the maximum among the three counts, the grid "$Middle$" is used to represent the item B in later building membership function FP-tree process. This step is repeated for the other items. Thus, "$Small$" is chosen for A, "$Small$" is chosen for C, "$Middle$" is chosen for D and "$Middle$" is chosen for E, shown as Table 4.

Step 4. Collect all maximum grids of item and check whether the count of each

Table 4: The set of fuzzy grids with maximum for each item

Item	Count
$A.Samll$	2.167
$B.Middle$	2.667
$C.Samll$	1.5
$D.Middle$	2.667
$E.Middle$	1.833

grid is larger than or equal to the predefined minimum fuzzy support value. Here, minimum fuzzy support value is 1.2, so the counts values of $A.Small$, $B.Middle$, $C.Small$, $D.Middle$ and $E.Middle$ are all larger than 1.2, these frequent 1-dim patterns are placed in L_1.
Step 5. Since L_1 is not null, the next step is then done.
Step 6. Use L_1 of frequent 1-dim patterns and establish a descending data table called Header Table, shown as Table 5.

Table 5: Header Table

1-dim pattern	Count
$B.Middle$	2.667
$D.Middle$	2.667
$A.Samll$	2.167
$E.Middle$	1.833
$C.Samll$	1.5

Step 7. According to Header Table, rebuild new fuzzy set table which sorts the original fuzzy set table, shown as Table 6.

Table 6: The new fuzzy sets from Table 2

TID	Fuzzy Sets
1	$\frac{0.833}{B.Middle} + \frac{0.5}{D.Middle} + \frac{0.667}{A.Small} + \frac{0.333}{E.Middle}$
2	$\frac{1.0}{B.Middle} + \frac{0.333}{D.Middle} + \frac{0.833}{C.Small}$
3	$\frac{1.0}{D.Middle} + \frac{0.833}{A.Small} + \frac{0.833}{E.Middle}$
4	$\frac{0.167}{B.Middle} + \frac{0.667}{A.Small} + \frac{0.667}{E.Middle}$
5	$\frac{0.667}{B.Middle} + \frac{0.833}{D.Middle} + \frac{0.667}{C.Small}$

Step 8. The root of the membership function FP-tree is initially set as {ROOT}. Next, scan the transaction database to build the membership function FP-tree tuple by tuple, which is according to each 1-dim pattern in Table 6 to establish. Next, link the node of fuzzy grids with the tree and take the first transaction as an example. The content is $\frac{0.833}{B.Middle}$, $\frac{0.5}{D.Middle}$, $\frac{0.667}{A.Small}$ and $\frac{0.333}{E.Middle}$. The $B.Middle$ is a linked node with $D.Low$. The same procedure is processed for the connection of $A.Small$ and $E.Middle$. Finally, complete results are shown in Fig.3.
Step 9. Mine the patterns of Header Table ascendingly. And set up the conditional pattern base of each node in a membership function of the FP-tree (MFFP-tree). Next, build conditional membership function of the FP-tree (CMFFP-tree).
Step 10. Use 1-dim pattern of Header table which is the lowest for the membership function FP-tree to mine. The 1-dim pattern of this example is $C.Small$. Next,

descendingly and repeatedly mine conditional membership function of the FP-tree, and gradually increase the frequency pattern base. If the conditional membership function of the FP-tree contains one path, all patterns can be listed, and results are shown in Table 7.

Table 7: Mining the FFP-tree by creating conditional pattern bases.

Item	Conditional Pattern-base	Conditional MFFP-tree	Frequent patterns generated
$C.Small$	$\{\{B.Middle:1\}, \{D.Middle:0.333\}\}$; $\{\{B.Middle:0.667\}, \{D.Middle:0.833\}\}$	$\langle B.Middle:1.667, D.Middle:1.166\rangle$	$\{(D.Middle),(C.Small)\}$; $\{(B.Middle),(C.Small)\}$; $\{(B.Middle),(D.Middle),(C.Small)\}$.
$E.Middle$	$\{\{B.Middle:0.833\}, \{D.Middle:0.5\}, \{A.Small:0.667\}\}$; $\{\{D.Middle:1.0\}, \{A.Small:0.833\}\}$; $\{\{B.Middle:0.167\}, \{A.Small:0.667\}\}$	$\langle B.Middle:0.833, D.Middle:0.5, A.Small:0.667\rangle$; $\langle B.Middle:0.167, A.Small:0.667, E.Middle:0.667\rangle$; $\langle D.Middle:1.0, A.Small:0.833, E.Middle:0.833\rangle$	$\{(A.Small),(E.Middle)\}$; $\{(D.Middle),(E.Middle)\}$; $\{(B.Middle),(E.Middle)\}$; $\{(D.Middle),(A.Small),(E.Middle)\}$; $\{(B.Middle),(A.Small),(E.Middle)\}$; $\{(B.Middle),(D.Middle),(E.Middle)\}$; $\{(B.Middle),(D.Middle),(A.Small),(E.Middle)\}$.
$A.Small$	$\{\{B.Middle:0.833\}, \{D.Middle:0.5\}\}$; $\{\{B.Middle:0.167\}\}$; $\{\{D.Middle:1.0\}\}$	$\langle B.Middle:0.833, D.Middle:0.5\rangle$; $\langle B.Middle:0.167\rangle$; $\langle D.Middle:1.0\rangle$;	$\{(D.Middle),(A.Small)\}$; $\{(B.Middle),(A.Small)\}$; $\{(B.Middle),(D.Middle),(A.Small)\}$.
$D.Middle$	$\{\{B.Middle:2.5\}\}$	$\langle B.Middle:2.5\rangle$	$\{(B.Middle),(D.Middle)\}$.

Step 11. The following substeps are done for corresponding frequent patterns.

(a) The generated frequent patterns are calculated. Here, the minimum operator is used for the intersection. Take $(B.Middle, C.Small)$ as an example. The fuzzy support value for TID 1 is calculated as: $\min\{0.833, 0\} = 0$. Results for the other transactions are shown in Table 8.

Table 8: The generate frequent pattern of $(B.Middle, C.Small)$ in each transaction

TID	$B.Middle$	$C.Small$	$B.Middle, C.Small$
1	0.833	0.0	0.0
2	1.0	0.833	0.833
3	0.0	0.0	0.0
4	0.167	0.0	0.0
5	0.667	0.667	0.667

(b) Each frequent pattern is calculated and the results of it are shown in Table 9.

Table 9: The count value for each frequent pattern

Frequent pattern	Count value
$(D.Middle, C.Small)$	1.000
$(B.Middle, C.Small)$	1.500
$(B.Middle, D.Middle, C.Small)$	1.000
$(A.Small, E.Middle)$	1.833
$(D.Middle, E.Middle)$	1.166
$(B.Middle, E.Middle)$	0.500
$(D.Middle, A.Small, E.Middle)$	1.166
$(B.Middle, A.Small, E.Middle)$	0.500
$(B.Middle, D.Middle, E.Middle)$	0.333
$(B.Middle, D.Middle, A.Small, E.Middle)$	0.333
$(D.Middle, A.Small)$	1.333
$(B.Middle, A.Small)$	0.834
$(B.Middle, D.Middle, A.Small)$	0.500
$(B.Middle, D.Middle)$	0.150

(c) Calculate the fuzzy confidence for the above frequent patterns. Take the fourth frequent pattern as an example. It is calculated as

$$(A.Small, E.Middle) = \frac{(A.Small, E.Middle)}{(A.Small)} = 0.846.$$

Since the fuzzy confidence of $(A.Small, E.Middle)$ is larger than the min fuzzy confidence value 0.58, this frequent pattern is an effective rule. In fact, the following four frequent patterns form association rules and are an output to users.

1. $(C.Small, B.Middle)$ with fuzzy confidence is 1.0: If a small number of C is bought, then a middle of B is bought with a confidence of 1.0.
2. $(A.Small, E.Middle)$ with fuzzy confidence is 0.846: If a small number of A is bought, then a middle of E is bought with a confidence of 0.846
3. $(E.Middle, A.Small)$ with fuzzy confidence is 1.0: If a middle number of E is bought, then a small of A is bought with a confidence of 1.0.
4. $(A.Small, D.Middle)$ with fuzzy confidence is 0.615: If a small number of A is bought, then a middle of D is bought with a confidence of 0.615. The four rules above are the output as meta-knowledge concerning the given transaction.

5. Experimental results

This section reports experiments made and the performance of the proposed approach and performed in VB on a Intel Core 2 Quad PC and 4G memory. A transaction database from a supermarket of the retail business in Kinmen, Taiwan, is used to show the feasibility. A total of 5562 transactions are included in the data set. Each transaction records the purchasing information of a customer. Besides, the assumptions of the membership functions for the quantities of items in the transaction database are shown in Fig.4. The relation between numbers of frequent length patterns and various minimum FS values are shown in Fig.5.

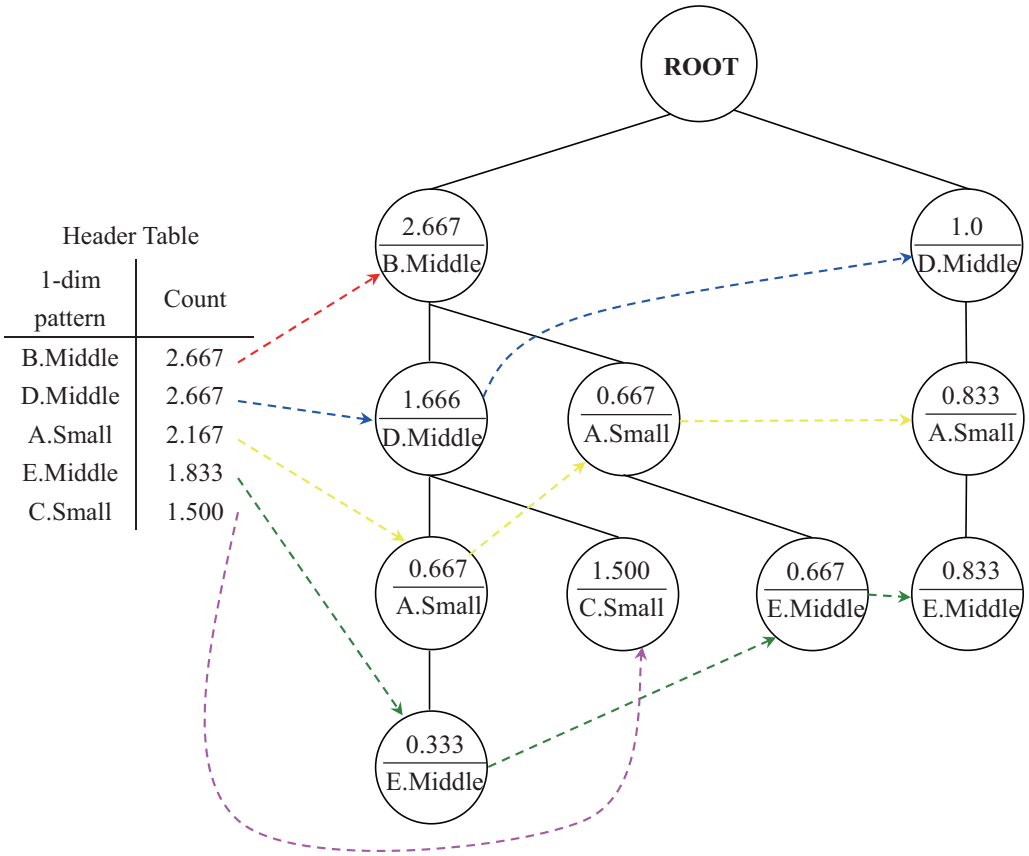

Fig.3. Membership function of FP-tree.

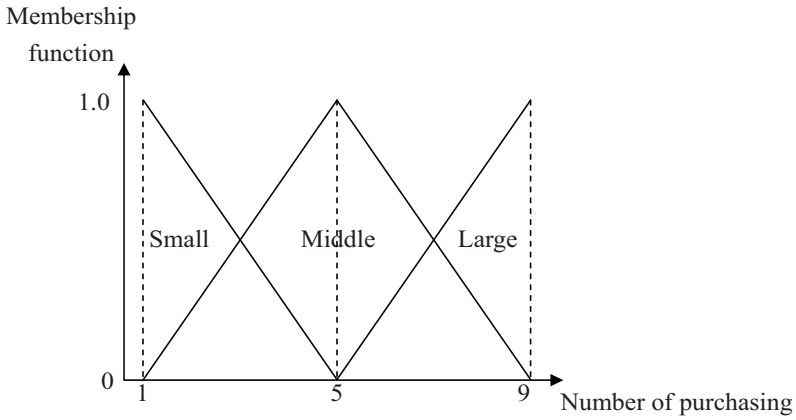

Fig.4. The membership function is used in the example.

From Fig.5, it is obvious that the numbers of frequent length patterns decreases along with an increase in minimum fuzzy support values. This result quite corresponds with our thoughts. The curve of the numbers of frequent length-1 patterns is also smoother than that of the numbers of frequent length-2 patterns, meaning that the minimum fuzzy support value has a larger influence on patterns. In addition,

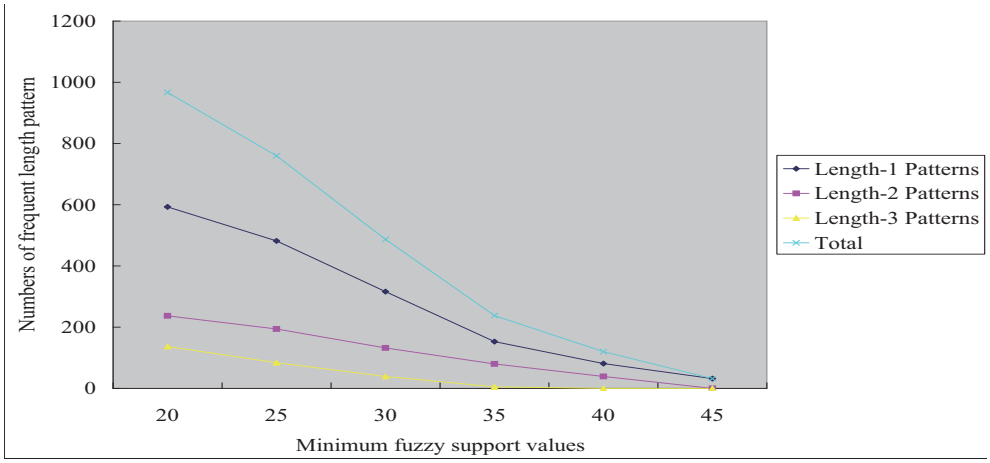

Fig.5. The relation between numbers of length patterns and minimum fuzzy support values.

appropriate minimum fuzzy support values can avoid unnecessary frequent length patterns and meaningless patterns.

Additionally, experiments are made to show the relation between numbers of association rules and minimum fuzzy support values along with different minimum fuzzy confidence values. Results are shown in Fig.6.

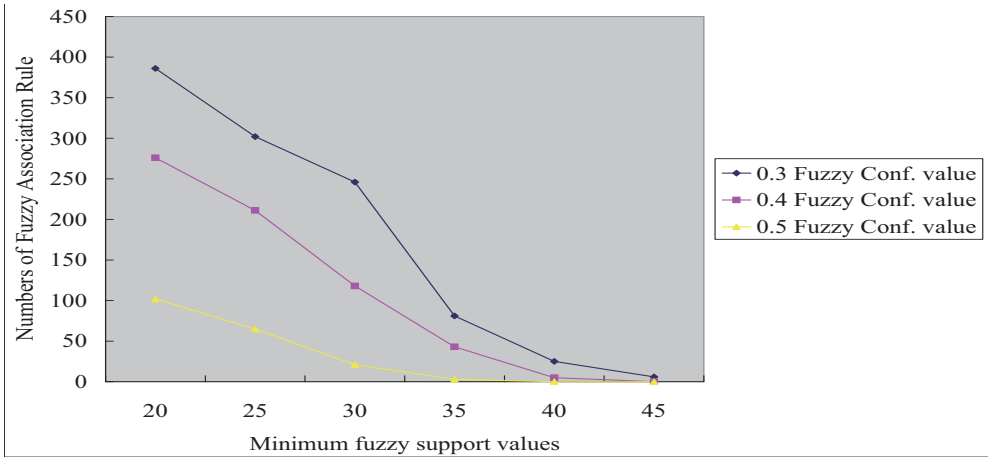

Fig.6. The relation between numbers of fuzzy association rules and minimum fuzzy support values.

According to Fig.6, the numbers of association rules decreases along with the increase in minimum fuzzy support values. Moreover, the curve of numbers of fuzzy association rules with larger minimum fuzzy confidence values is smoother than that of those with smaller minimum fuzzy confidence values, meaning that the minimum fuzzy support value has a large influence on the number of association rules derived from the small minimum confidence values.

In the measure of accuracy, the data set is divided into two parts: training data and testing data, and the proposed method is performed on the training data to

induce the rules. And then the rules are tested on the testing data to measure the percentage of correct predictions. In each performed, 3708 cases are selected at random for training and the remaining 1854 cases are used for testing. Results for different minimum fuzzy support values and minimum fuzzy confidence values are shown in Fig.7.

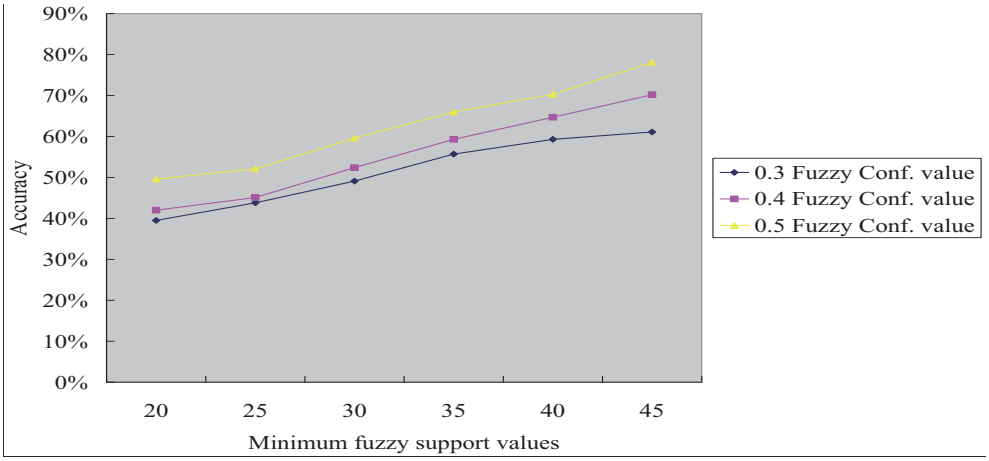

Fig.7. The relation between accuracy and minimum fuzzy support values for fuzzy confidence values.

According to Fig.7, the mining algorithm performed at a higher minimum fuzzy confidence value has a higher accuracy, since the minimum fuzzy confidence value can be thought of as an accuracy threshold for deriving results. And the average accuracy of all the rules are also higher for a larger minimum fuzzy confidence value.

Next, the relation between the execution time and different minimum fuzzy support values are shown in Fig.8. According to Fig.8, the execution time also tends to decrease along with the increase of the numbers of grids since frequent patterns are reduced under many number fuzzy grids.

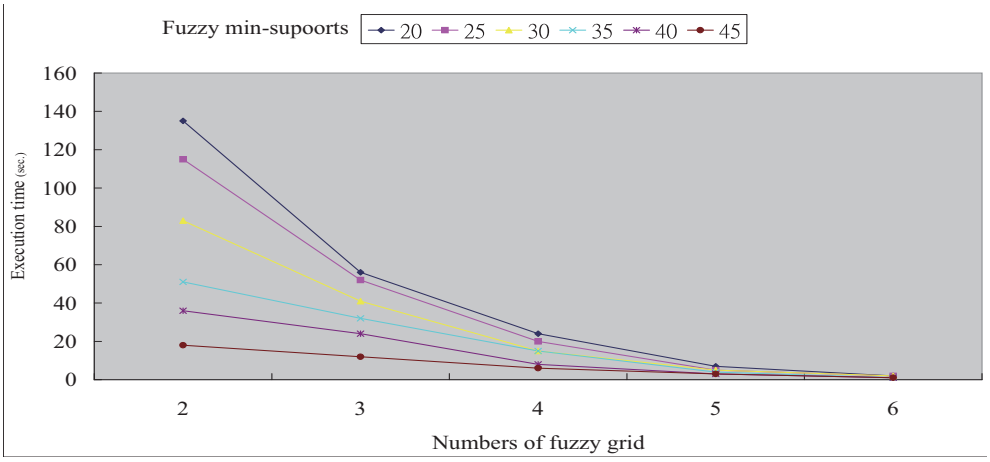

Fig.8. The relation between execution times and number of fuzzy grids.

Finally, the experiment made to compare with that of the proposed method in Lin *et al.* [13] and Padaimitriou [15] in which the accuracy for the minimum fuzzy support value set at 35 and K = 3 is shown in Fig.9.

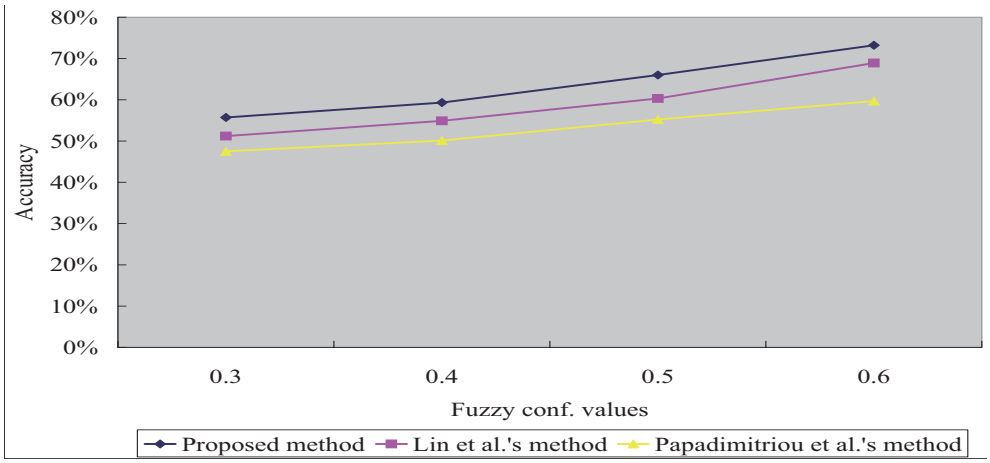

Fig.9. The comparison of the accuracy of three fuzzy data mining algorithms.

According to Fig.9, the accuracy of the proposed method is higher than that of Lin *et al.* [13] and Padaimitriou [15] 's proposed methods for various minimum fuzzy confidence values.

6. Conclusions

In this paper, we have proposed a fuzzy data mining algorithm, which combines fuzzy set theory and FP-Growth to deal with quantitative values, and have found interesting patterns among them. The rules mined represent quantitative regularity for large transaction databases and can be adopted to provide sound suggestions to appropriate supervisors. The proposed algorithm can solve disadvantages found in the Apriori algorithm and enhance the whole efficiency. The experimental results with the data in a supermarket of the retail business show the feasibility of the proposed mining algorithm. When comparing with that of Lin *et al.*'s [13] and Papadimitriou *et al.*'s [15] fuzzy mining method, our approach can get better mining efficiency and the experimental results pronounces the proposed method is more excellent than other mining methods.

References

[1] R. Agrawal, T. Imielinksi and A. Swami, *Mining association rules between sets of items in large database*, in The 1993 ACM SIGMOD Conference, Washington DC, USA, 1993, pp. 207–216.
[2] R. Agrawal, T. Imielinksi and A. Swami, *Database mining: a performance perspective*, IEEE Transactions on Knowledge and Data Engineering **5** (1993), 914–925.
[3] R. Agrawal and R. Srikant, *Fast algorithms for mining association rules*, in Proceedings of 1994 International Conference on Very Large Data Bases, 1994, pp. 487–499.
[4] M. Berry and G. Linoff, *Data Mining Techniques: for marketing, sales, and customer support*, John Wiley & Sons, NY, 1997.
[5] K. C. C. Chen and W. H. Au, *Mining fuzzy association rules*, in The 6th International Conference on Information and knowledge Management, 1997, pp. 209–215.

[6] J. W. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, San Francisco, 2001.
[7] J. Han, J. Pei and Y. Yin, *Mining frequent patterns without candidate generation*, in Proc. ACM SIGMOD Int. Conf. on Management of Data, 2000, pp. 1–12.
[8] T. P. Hong, C. S. Kuo, S. C. Chi, *A data mining algorithm for transaction data with quantitative values*, International Data Analysis, (1999) 363–376.
[9] Y. C. Hu, *Mining association rules at a concept hierarchy using fuzzy partition*, Journal of Information Management **13** (2006), 63–80.
[10] Y. C. Hu, R. S. Chen and G. H. Tzeng, *Finding fuzzy classification rules using data mining techniques*, Pattern Recognition Letters **24** (2003), 509–519.
[11] Y. C. Hu, *Finding useful fuzzy concepts for pattern classification using genetic algorithm*, Information Sciences **175** (2005), 1–19.
[12] C. M. Kuok, A. Fu and M. H. Wong, *Mining fuzzy association rules in database*, SIGMOD Record **27** (1998), 41–46.
[13] C. W. Lin, T. P. Hong and W. H. Lu, *An efficient tree-based fuzzy data mining approach*, International Journal of Fuzzy Systems **12** (2010), 150–157.
[14] S. Myra, *Web usage mining for web site evaluation*, Communications of the ACM **43** (1994), 21–30.
[15] S. Papadimitriou and S. Mavroudi, *The frequent fuzzy patter tree*, in The 9th WSEAS International Conference on Computer, 2005.
[16] P. N. Tan, Michael Steinbach and Vipin Kumar, *Introduction to Data Mining*, Pearson Addison Wesley, Boston, 2005.
[17] L. A. Zadeh, *Fuzzy sets*, Information Control **8** (1965), 338–353.
[18] L. A. Zadeh, *The concept of a linguistic variable and its application to approximate reasoning*, Information Science (part 1) **8** (1975), 199–249.
[19] L. A. Zadeh, *The concept of a linguistic variable and its application to approximate reasoning*, Information Science (part 2) **8** (1975), 301–357.
[20] L. A. Zadeh, *The concept of a linguistic variable and its application to approximate reasoning*, Information Science (part 3) **9** (1976), 43–80.

Manuscript received March 22, 2012
revised August 15, 2012

Chien-Hua Wang
Department of Information Management, Yuan Ze University, Taoyuan, Taiwan, 320
E-mail address: thuck@saturn.yzu.edu.tw

Wei-Hsuan Lee
Department of Information Management, Yuan Ze University, Taoyuan, Taiwan, 320
E-mail address: s969206@mail.yzu.edu.tw

Chin-Tzong Pang
Department of Information Management, Yuan Ze University, Taoyuan, Taiwan, 320
E-mail address: imctpang@saturn.yzu.edu.tw