Available online at www.sciencedirect.com

SCIENCE DIRECT®

Applied Soft Computing 5 (2004) 1–10

Applied Soft Computing

www.elsevier.com/locate/asoc

A fuzzy AprioriTid mining algorithm with reduced computational time☆

Tzung-Pei Hong [a,*], Chan-Sheng Kuo [b], Shyue-Liang Wang [c]

[a] *Department of Electrical Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan, ROC*
[b] *Department of Management Information Systems, National Chengchi University, Taipei 116, Taiwan, ROC*
[c] *Department of Computer Science, New York Institute of Technology, NY 10023, USA*

Received 18 November 2002; received in revised form 11 March 2004; accepted 16 March 2004

Abstract

Due to the increasing use of very large databases and data warehouses, mining useful information and helpful knowledge from transactions is evolving into an important research area. Most of conventional data mining algorithms identify the relation among transactions with binary values. Transactions with quantitative values are, however, commonly seen in real world applications. In the past, we proposed a fuzzy mining algorithm based on the *Apriori* approach to explore interesting knowledge from the transactions with quantitative values. This paper proposes another new fuzzy mining algorithm based on the *AprioriTid* approach to find fuzzy association rules from given quantitative transactions. Each item uses only the linguistic term with the maximum cardinality in later mining processes, thus making the number of fuzzy regions to be processed the same as that of the original items. The algorithm therefore focuses on the most important linguistic terms for reduced time complexity. Experimental results from the data in a supermarket of a department store show the feasibility of the proposed mining algorithm.

Keywords: Data mining; Fuzzy set; Association rule; Transaction; Quantitative value

1. Introduction

In data mining researches, inducing association rules from transaction data is the most commonly seen [10,18]. Most of the previous research works can, however, only handle transaction data with attributes of binary values. In real-world applications, transaction data are usually composed of quantitative values. Designing a sophisticated data-mining algorithm to deal with different types of data turns a challenge in this research topic.

Fuzzy set theory is being used more and more frequently in intelligent systems because of its simplicity and similarity to human reasoning [17]. Several fuzzy learning algorithms for inducing rules from given sets of data have been designed and used to good effect with specific domains [5,7–9,11,13–15,20]. Using fuzzy sets in data mining has also been developed in recent years [6,16,21].

In [16], 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

☆ This is a modified and expanded version of the paper “A fuzzy AprioriTid mining algorithm with reduced computational time,” presented at The Tenth IEEE International Conference on Fuzzy Systems.

* Corresponding author.
E-mail addresses: tphong@nuk.edu.tw (T.-P. Hong), cskuo@nccu.edu.tw (C.-S. Kuo), slwang@nyit.edu (S.-L. Wang).

1568-4946/$ – see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2004.03.009

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. This paper proposes another new fuzzy mining algorithm based on the AprioriTid approach [4] to find fuzzy association rules from given quantitative transactions. It is capable of transforming quantitative values in transactions into linguistic terms, then filtering them, and finding association rules. Each item uses only the linguistic term with the maximum cardinality (highest count) in later mining processes, thus making the number of fuzzy regions to be processed the same as that of the original items. The algorithm therefore focuses on the most important linguistic terms for reduced time complexity. Experimental results from the data in a supermarket of a department store show the feasibility of the proposed mining algorithm.

The remaining parts of this paper are organized as follows. Related research is reviewed in Section 2. The proposed fuzzy AprioriTid data-mining algorithm is described in Section 3. An example is given to illustrate the proposed algorithm in Section 4. Experiments to demonstrate the performance of the proposed data-mining algorithm are stated in Section 5. Conclusions and future work are finally given in Section 6.

2. Related research

As mentioned above, the goal of data mining is to discover the important associations among items such that the presence of some items in a transaction will imply the presence of some other items. For achieving this purpose, Agrawal and his co-workers proposed several mining algorithms based on the concept of large itemsets to find association rules from transactions [1–4]. They decomposed the mining process into two phases. In the first phase, candidate itemsets are generated and counted by scanning the transactions. If the number of an itemset appearing in the transactions is larger than a pre-defined threshold value (called minimum support), the itemset is thought of as a large itemset. Itemsets with only one item are first processed. The large itemsets with one item are then combined to form candidate itemsets of two items. This process is repeated until all large itemsets are found. In the second phase, the desired association rules are induced from the large itemsets found in the first phase. All the possible combination ways of association rules for each large itemset are formed, and the ones with their calculated confidence values larger than a predefined threshold (called minimum confidence) are output as desired association rules.

In addition to proposing methods for mining association rules from transactions of binary values, Srikant and Agrawal also proposed a method to mine association rules from those with quantitative and categorical attributes [19]. 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 the large itemsets whose support values are greater than the user-specified minimum support. These large itemsets are then processed to generate association rules, and the interesting rules are output from the viewpoint of users.

Fuzzy set theory was first proposed by Zadeh [22]. 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. Recently, fuzzy sets have also been used in data mining to increase its flexibility. Hong et al. proposed a fuzzy mining algorithm to mine fuzzy rules from quantitative data [16]. They transformed each quantitative item into a fuzzy set and used fuzzy operations to find fuzzy rules. Cai et al. proposed weighted mining to reflect different importance to different items [6]. Each item was attached a numerical weight given by users. Weighted supports and weighted confidences were then defined to determine interesting association rules. Yue et al. then extended their concepts to fuzzy item vectors [21]. This paper proposes another new fuzzy mining algorithm based on the AprioriTid approach [4] to find fuzzy association rules from given quantitative transactions.

3. The proposed fuzzy data-mining algorithm

The role of fuzzy sets helps transform quantitative values into linguistic terms, thus reducing possible itemsets in the mining process. They are used in the AprioriTid data-mining algorithm to discover useful association rules from 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;
$D^{(i)}$: the ith transaction datum, $1 \le i \le n$;
A_j: the jth attribute, $1 \le j \le m$;
$|A_j|$: the number of fuzzy regions for A_j;
R_{jk}: the kth fuzzy region of A_j, $1 \le k \le |A_j|$;
$v_j^{(i)}$: 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 $i = 1$–n;
$\mathrm{count}_j^{\max}$: the maximum count value among count_{jk} values, $k = 1$ to $|A_j|$;
$R_j^{\max}$: the fuzzy region of A_j with $\mathrm{count}_j^{\max}$
α: the predefined minimum support level;
λ: the predefined minimum confidence value;
C_r: the set of candidate itemsets with r attributes (items);
$\bar{C}_r$: the temporary set for recording the fuzzy values of r-items in each data;
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 with linguistic terms using membership functions. The algorithm then calculates the scalar cardinality of each linguistic term on all the transaction data using the temporary set $\bar{C}_r$. 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 mining process based on fuzzy counts is then performed to find fuzzy association rules. The detail of the proposed mining algorithm is described as follows.

The 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 associate rules.

STEP 1. Transform the quantitative value $v_j^{(i)}$ of each transaction datum $D^{(i)}$, $i = 1$–n, for each attribute A_j, $j = 1$–m, into a fuzzy set y represented as $(f_{j_1}^{(i)}/R_{j_1} + f_{j_2}^{(i)}/R_{j_2} + \cdots + f_{j_l}^{(i)}/R_{j_l})$ using the given membership functions, where R_{jk} is the kth 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. Build a temporary set $\bar{C}_1$ including all the pairs $(R_{jk}, f_{jk}^{(i)})$ of each data, where $1 \le i \le n$, $1 \le j \le m$, $1 \le k \le |Aj|$, and $f_{jk}^{(i)} \ne 0$.

STEP 3. For each region R_{jk} stored in $\bar{C}_1$, calculate its scalar cardinality for all the transactions from $\bar{C}_1$:

$$\mathrm{count}_{jk} = \sum_{i=1}^{n} f_{jk}^{(i)}.$$

STEP 4. Find $\mathrm{count}_j^{\max} = \mathrm{Max}_{k=1}^{|A_j|}(\mathrm{count}_{jk})$, for $j = 1$–m, where $|A_j|$ is the number of fuzzy regions for A_j. Let $R_j^{\max}$ be the region with $\mathrm{count}_j^{\max}$ for attribute A_j. $R_j^{\max}$ will be used to represent this attribute in later mining processing.

STEP 5. Check whether the $\mathrm{count}_j^{\max}$ of each $R_j^{\max}$, $j = 1$–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 one-itemsets (L_1). That is,

$$L_1 = \{R_j^{\max} | \mathrm{count}_j^{\max} \ge \alpha, 1 \le j \le m\}.$$

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

STEP 7. Generate the candidate set C_{r+1} from L_r. Restated, the algorithm 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. Store in C_{r+1} the itemsets which have all their sub-r-itemsets in L_r.

STEP 8. Build an empty temporary set $\bar{C}_{r+1}$.

STEP 9. Do the following substeps for each newly formed $(r+1)$-itemset s with items $(s_1, s_2, \ldots, s_{r+1})$ in C_{r+1}:

(a) For each transaction datum $D^{(i)}$, calculate its fuzzy value on s as $f_s^{(i)} = f_{s_1}^{(i)} \Lambda f_{s_2}^{(i)} \Lambda \cdots \Lambda f_{s_{r+1}}^{(i)}$ using $\bar{C}_r$, where $f_{s_j}^{(i)}$ is the fuzzy membership value of $D^{(i)}$ in region s_j. If the minimum operator is used for the intersection, then $f_s^{(i)} = \mathrm{Min}_{j=1}^{r+1} f_{s_j}^{(i)}$.

(b) Store the pair (s, $f_s^{(i)}$) of Case i in $\bar{C}_{r+1}$, where $1 \le i \le n$, $f_s^{(i)} \ne 0$.

(c) Set $\mathrm{count}_s = \sum_{i=1}^{n} f_s^{(i)}$ using $\bar{C}_{r+1}$.

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

STEP 10. IF L_{r+1} is null, then do the next step; otherwise, set $r = r + 1$ and repeat STEPs 7–10.

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

(a) Form all possible association rules as follows:

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

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

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

STEP 12. Output the rules with confidence values larger than or equal to the predefined confidence threshold λ.

After STEP 12, the rules constructed are output and can act as the meta-knowledge for the given transactions.

4. 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.

Each case consists of five course scores: statistics (denoted ST), database (denoted DB), object-oriented programming (denoted OOP), 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 Fig. 1.

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

Case no.	ST	DB	OOP	DS	MIS
1	86	77	86	71	68
2	61	79	89	77	80
3	84	89	86	79	89
4	73	86	79	84	62
5	70	89	87	72	79
6	65	77	86	61	87
7	67	87	75	71	80
8	86	63	64	84	86
9	75	65	79	87	88
10	79	63	63	85	89

In this example, triangular membership functions are used to represent fuzzy sets due to their simplicity, easy comprehension, and computational efficiency. They are usually assigned by experts as in most applications. They can also be derived through automatic adjustment [12]. In addition to triangular membership functions, other types such as the Gaussian can be used in the proposed algorithm, which is independent of the types of membership functions.

From Fig. 1, 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 ST score in Case 1 as an example. The score "86" is converted into a fuzzy set (0.0/Low + 0.0/Middle + 0.7/High) using

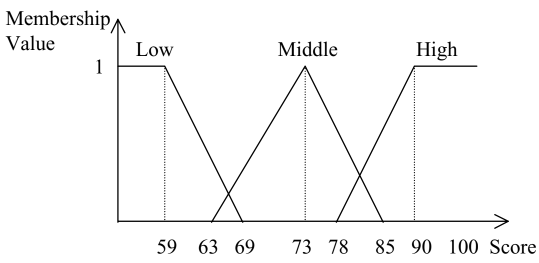

Fig. 1. The membership function used in this example.

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

Case no.	ST			DB			OOP			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	0.5	0.1	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	0.9	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	0.0	0.7	0.0	0.0	0.0	0.7	0.8	0.0	0.0	0.0	0.0	0.8
7	0.2	0.4	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	0.6	0.0	0.0	0.5	0.1	0.0	0.0	0.1	0.5	0.0	0.0	0.7
9	0.0	0.8	0.0	0.4	0.2	0.0	0.0	0.5	0.1	0.0	0.0	0.8	0.0	0.0	0.8
10	0.0	0.5	0.1	0.6	0.0	0.0	0.6	0.0	0.0	0.0	0.0	0.6	0.0	0.0	0.9

Table 3
The temporary set $\bar{C}_1$ for this example

Case	Set-of-itemsets
1	{(ST.High, 0.7), (DB.Middle, 0.7), (OOP.High, 0.7), (DS.Middle, 0.8), (MIS.Low, 0.1), (MIS.Middle, 0.5)}
2	{(ST.Low, 0.8), (DB.Middle, 0.5), (DB.High, 0.1), (OOP.High, 0.9), (DS.Middle, 0.7), (MIS.Middle, 0.4), (MIS.High, 0.2)}
3	{(ST.Middle, 0.1), (ST.High, 0.5), (DB.High, 0.9), (OOP.High, 0.7), (DS.Middle, 0.5), (DS.High, 0.1), (MIS.High, 0.9)}
4	{(ST.Middle, 1.0), (DB.High, 0.7), (OOP.Middle, 0.5), (OOP.High, 0.1), (DS.Middle, 0.1), (DS.High, 0.5),(MIS.Low, 0.7)}
5	{(ST.Middle, 0.7), (DB.High, 0.9), (OOP.High, 0.8), (DS.Middle, 0.9), (MIS.Middle, 0.5), (MIS.High, 0.1)}
6	{(ST.Low, 0.4), (ST.Middle, 0.2), (DB.Middle, 0.7), (OOP.High, 0.7), (DS.Low, 0.8), (MIS.High, 0.8)}
7	{(ST.Low, 0.2), (ST.Middle, 0.4), (DB.High, 0.8), (OOP.Middle, 0.8), (DS.Middle, 0.8), (MIS.Middle, 0.4), (MIS.High, 0.2)}
8	{(ST.High, 0.7), (DB.Low, 0.6), (OOP.Low, 0.5), (OOP.Middle, 0.1), (DS.Middle, 0.1), (DS.High, 0.5), (MIS.High, 0.7)}
9	{(ST.Middle, 0.8), (DB.Low, 0.4), (DB.Middle, 0.2), (OOP.Middle, 0.5), (OOP.High, 0.1), (DS.High, 0.8), (MIS.High, 0.8)}
10	{(ST.Middle, 0.5), (ST.High, 0.1), (DB.Low, 0.6), (OOP.Low, 0.6), (DS.High, 0.6), (MIS.High, 0.9)}

the given membership functions. This step is repeated for the other cases and courses, and the results are shown in Table 2.

STEP 2. Build a temporary set $\bar{C}_1$ including all the pairs $(R_{jk}, f_{jk}^{(i)})$ of each data. The results are shown in Table 3.

STEP 3. For each attribute region, calculate its scalar cardinality for all the transactions from $\bar{C}_1$ as the *count* value. Take the region *ST.Low* as an example. Its scalar cardinality $= (0.8 + 0.4 + 0.2) = 1.4$. Repeating this step for the other regions, the results are shown in Table 4.

STEP 4. Find the region with the highest count among the three possible regions for each attribute. Take the course *ST* as an example. The count is 1.4 for *Low*, 3.7 for *Middle*, and 2.0 for *High*. Since the count for *Middle* is the highest among the three counts, the region *Middle* is thus used to represent the course *ST* in later mining process. This step is repeated for the other regions. "*High*" is thus chosen for DB, OOP and MIS, and "*Middle*" is chosen for ST and DS. The number of items chosen is thus the same as that of the original attributes, meaning the

Table 4
The set of one-itemsets with their counts for this example

Itemset	Count
ST.Low	1.4
ST.Middle	3.7
ST.High	2.0
DB.Low	1.6
DB.Middle	2.1
DB.High	3.4
OOP.Low	1.1
OOP.Middle	1.9
OOP.High	4.0
...	...
MIS.High	4.6

Table 5
The set of large one-itemsets L_1 for this example

Itemset	Count
ST.Middle	3.7
DB.High	3.4
OOP.High	4.0
DS.Middle	3.9
MIS.High	4.6

algorithm will focus on the important items, and the time complexity could thus be reduced.

STEP 5. For each region selected in STEP 4, check whether its count is larger than or equal to the predefined minimum support value α. The minimum support value is usually assigned by users according to the distribution of frequencies of items. It will have a strong impact on the numbers of large itemsets and association rules. The numbers of association rules decreased along with the increase in minimum support values (later experiments will show this).

Assume in this example, α is set at 2.0. Since the count values of ST.Middle, DB.Middle, OOP.High, DS.Middle, and MIS.High are all larger than 2.0, these items are put in L_1 (Table 5).

STEP 6. Set $r = 1$.

STEP 7. Generate the candidate set C_{r+1} from L_r. C_2 is first generated from L_1 as follows: (ST.Middle, DB.High), (ST.Middle, OOP.High), (ST.Middle, DS.Middle), (ST.Middle, MIS.High), (DB.High, OOP.High), (DB.High, DS.Middle), (DB.High, MIS.High), (OOP.High, DS.Middle), (OOP.High, MIS.High), and (DS.Middle, MIS.High).

STEP 8. Build an empty temporary set $\bar{C}_{r+1}$. $\bar{C}_2$ is thus built.

STEP 9. For each newly formed candidate itemset s in C_2, do the following substeps.

(a) For each transaction data, calculate its fuzzy membership value for this itemset from $\bar{C}_1$. Here, the minimum operator is used for the intersection. Take the candidate itemset (ST.Middle, DB.High) as an example. Only cases 3, 4, 5 and 7 contain both the items ST.Middle and DB.High in $\bar{C}_1$. The derived fuzzy membership functions are shown in Table 6.

The results for the other two-itemsets can be derived in a similar way.

(b) Store the pair $(s, f_s^{(i)})$ of Case i in $\bar{C}_2$, where $f_s^{(i)} \neq 0$. Results are shown in Table 7.

(c) Set counts $= \sum_{i=1}^{n} f_s^{(i)}$ using $\bar{C}_2$. The scalar cardinality (count) of each candidate itemset in C_2 is thus calculated. Results for this example are shown in Table 8.

(d) Check whether these counts are larger than or equal to the predefined minimum support value 2.0. Two itemsets, (DB.High, DS.Middle) and (OOP.High, DS.Middle), are thus kept in L_2 (Table 9).

Table 6
The membership values for (ST.Middle, DB.High)

Case	ST.Middle	DB.High	(ST.Middle, DB.High)
1	0.0	0.0	0.0
2	0.0	0.1	0.0
3	0.1	0.9	0.1
4	1.0	0.7	0.7
5	0.7	0.9	0.7
6	0.2	0.0	0.0
7	0.4	0.8	0.4
8	0.0	0.0	0.0
9	0.8	0.0	0.0
10	0.5	0.0	0.0

STEP 10. IF L_{r+1} is null, then do the next step; otherwise, set $r = r + 1$ and repeat STEPs 7–10. Since L_2 is not null in the example, $r = r + 1 = 2$. STEPs 7–10 are then repeated to find L_3. C_3 is first generated from L_2, and only the itemset (DB.High, OOP.High, DS.Middle) is formed. Its count is calculated as 1.5, smaller than 2.0. It is thus not put in L_3. Since L_3 is an empty set, STEP 11 begins.

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

(a) Form all possible association rules. The following four possible association rules are then formed from the large two-itemsets (DB.High, DS.Middle) and (OOP.High, DS.Middle):

If DB = High, then DS = Middle;
If DS = Middle, then DB = High;

Table 7
The temporary set $\bar{C}_2$ for this example

Case	Set-of-itemsets
1	{(OOP.High, DS.Middle, 0.7)}
2	{(DB.High, OOP.High, 0.1), (DB.High, DS.Middle, 0.1), (DB.High, MIS.High, 0.1), (OOP.High, DS.Middle, 0.7), (OOP.High, MIS.High, 0.2), (DS.Middle, MIS.High, 0.2)}
3	{(ST.Middle, DB.High, 0.1), (ST.Middle, OOP.High, 0.1), (ST.Middle, DS.Middle, 0.1), (ST.Middle, MIS.High, 0.1), (DB.High, OOP.High, 0.7), (DB.High, DS.Middle, 0.5), (DB.High, MIS.High, 0.9), (OOP.High, DS.Middle, 0.5), (OOP.High, MIS.High, 0.7), (DS.Middle, MIS.High, 0.5)}
4	{(ST.Middle, DB.High, 0.7), (ST.Middle, OOP.High, 0.1), (ST.Middle, DS.Middle, 0.1), (DB.High, OOP.High, 0.1), (DB.High, DS.Middle, 0.1), (OOP.High, DS.Middle, 0.1)}
5	{(ST.Middle, DB.High, 0.7), (ST.Middle, OOP.High, 0.7), (ST.Middle, DS.Middle, 0.7), (ST.Middle, MIS.High, 0.1), (DB.High, OOP.High, 0.8), (DB.High, DS.Middle, 0.9), (DB.High, MIS.High, 0.1), (OOP.High, DS.Middle, 0.8), (OOP.High, MIS.High, 0.1), (DS.Middle, MIS.High, 0.1)}
6	{(ST.Middle, OOP.High, 0.2), (ST.Middle, MIS.High, 0.2), (OOP.High, MIS.High, 0.7)}
7	{(ST.Middle, DB.High, 0.4), (ST.Middle, DS.Middle, 0.4), (ST.Middle, MIS.High, 0.2), (DB.High, DS.Middle, 0.8), (DB.High, MIS.High, 0.2), (DS.Middle, MIS.High, 0.2)}
8	{(DS.Middle, MIS.High, 0.1)}
9	{(ST.Middle, OOP.High, 0.1), (ST.Middle, MIS.High, 0.8), (OOP.High, MIS.High, 0.1)}
10	{(ST.Middle, MIS.High, 0.5)}

If OOP = High, then DS = Middle;
If DS = Middle, then OOP = High.

(b) Calculate the confidence values of the above association rules. Assume the given confidence threshold λ is 0.70. Take the first association rule as an example. Its confidence value is calculated as:

$$\frac{\sum_{i=1}^{10}(\text{DB.High} \cap \text{DS.Middle})}{\sum_{i=1}^{10}(\text{DB.High})} = \frac{2.4}{3.4} = 0.71.$$

The confidence values of the other three rules are shown below.

"If DS = Middle, then DB = High" has a confidence value of 0.62;

"If OOP = High, then DS = Middle" has a confidence value of 0.70;

"If DS = Middle, then OOP = High" has a confidence value of 0.72.

Table 8
The counts of the fuzzy itemsets in C_2

Itemset	Count
(ST.Middle, DB.High)	1.9
(ST.Middle, OOP.High)	1.2
(ST.Middle, DS.Middle)	1.3
(ST.Middle, MIS.High)	1.9
(DB.High, OOP.High)	1.7
(DB.High, DS.Middle)	2.4
(DB.High, MIS.High)	1.3
(OOP.High, DS.Middle)	2.8
(OOP.High, MIS.High)	1.8
(DS.Middle, MIS.High)	1.1

STEP 12. Check whether the confidence values of the above association rules are larger than or equal to the predefined confidence threshold λ. Since the confidence λ was set at 0.70 in this example, the following three rules are thus output to users:

1. If the score of database is high, then the score of data structure is middle, with a confidence value of 0.71.
2. If the score of object-oriented programming is high, then the score of data structure is middle, with a confidence value of 0.70.
3. If the score of data structure is middle, then the score of object-oriented programming is high, with a confidence value of 0.72.

Table 9
The itemsets and their fuzzy counts in L_2

Itemset	Count
(DB.High, DS.Middle)	2.4
(OOP.High, DS.Middle)	2.8

After STEP 12, the three rules above are thus output as meta-knowledge concerning the given transactions.

5. Experiments

A 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 relationship between numbers of large itemsets and minimum support values for $\lambda = 0.3$ are shown in Fig. 2.

From Fig. 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 one-itemsets was also smoother than that of the numbers of large two-itemsets, meaning that the minimum support value had a larger influence on itemsets with more items.

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

From Fig. 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.

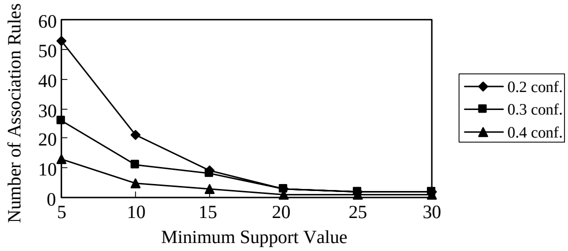

Fig. 3. The relationship between numbers of association rules and minimum support values.

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

From Fig. 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 then made to measure the accuracy of the proposed approach. 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

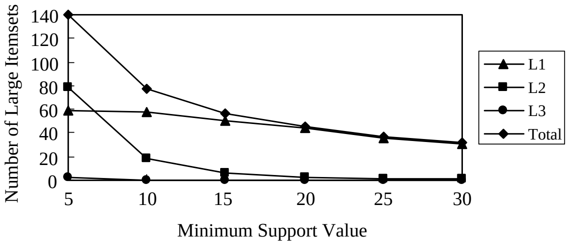

Fig. 2. The relationship between numbers of large itemsets and minimum support values.

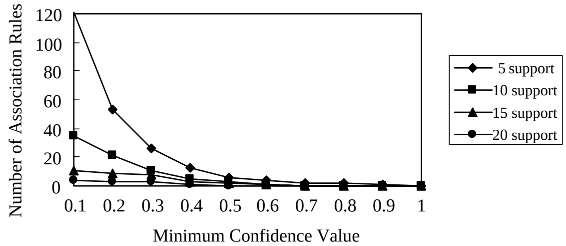

Fig. 4. The relationship between numbers of association rules and minimum confidence values.

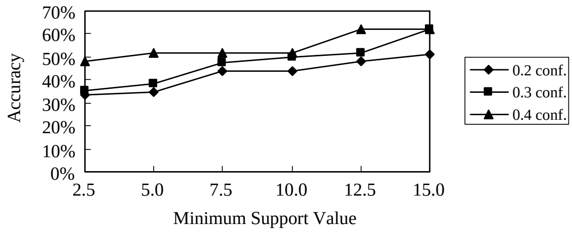

Fig. 5. The relationship between accuracy and minimum support values for various confidence values.

testing. Results for different minimum support values and confidence values are shown in Fig. 5.

From Fig. 5, it is easily seen that the accuracy increased along with an increase in minimum support values, meaning that a large minimum support value yielded a higher accuracy than a small minimum support value. It is also easily seen that the mining algorithm running 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.

6. Conclusion and future work

In this paper, we have proposed a fuzzy data-mining algorithm based on the AprioriTid approach to process transaction data with quantitative values and discover fuzzy association rules among them. Each item uses only the linguistic term with the maximum cardinality in the mining processes, thus making the number of fuzzy regions to be processed the same as that of the original items. The algorithm therefore focuses on the most important linguistic terms for reduced time complexity. The rules mined out exhibit quantitative regularity in 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 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 [13,15], 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 the acquisition of membership functions. We will also attempt to design different data-mining models for different problem domains.

Acknowledgements

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 NSC 92-2213-E-390-001.

References

[1] R. Agrawal, T. Imielinksi, 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, A. Swami, Database mining: a performance perspective, IEEE Trans. Knowledge Data Eng. 5 (6) (1993) 914–925.
[3] R. Agrawal, R. Srikant, 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, R. Srikant, Fast algorithm for mining association rules, The International Conference on Very Large Data Bases, 1994, pp. 487–499.
[5] A.F. Blishun, Fuzzy learning models in expert systems, Fuzzy Sets Syst. 22 (1987) 57–70.
[6] C.H. Cai, W.C. Fu, C.H. Cheng, W.W. Kwong, Mining association rules with weighted items, The International Database Engineering and Applications Symposium, 1998, pp. 68–77.
[7] L.M. de Campos, S. Moral, Learning rules for a fuzzy inference model, Fuzzy Sets Syst. 59 (1993) 247–257.
[8] R.L.P. Chang, T. Pavliddis, Fuzzy decision tree algorithms, IEEE Trans. Syst. Man Cybernetics 7 (1977) 28–35.
[9] M. Delgado, A. Gonzalez, An inductive learning procedure to identify fuzzy systems, Fuzzy Sets Syst. 55 (1993) 121–132.
[10] W.J. Frawley, G. Piatetsky-Shapiro, C.J. Matheus, Knowledge discovery in databases: an overview, The AAAI Workshop on Knowledge Discovery in Databases, 1991, pp. 1–27.
[11] A. Gonzalez, A learning methodology in uncertain and imprecise environments, Int. J. Intell. Syst. 10 (1995) 57–371.
[12] T.P. Hong, C.H. Chen, Y.L. Wu, Y.C. Lee, Using divide-and-conquer GA strategy in fuzzy data mining, The

Ninth IEEE Symposium on Computers and Communications, 2004.

[13] T.P. Hong, J.B. Chen, Finding relevant attributes and membership functions, Fuzzy Sets Syst. 103 (3) (1999) 389–404.

[14] T.P. Hong, J.B. Chen, Processing individual fuzzy attributes for fuzzy rule induction, Fuzzy Sets Syst. 112 (1) (2000) 127–140.

[15] T.P. Hong, C.Y. Lee, Induction of fuzzy rules and membership functions from training examples, Fuzzy Sets Syst. 84 (1996) 33–47.

[16] T.P. Hong, C.S. Kuo, S.C. Chi, Mining association rules from quantitative data, Intell. Data Anal. 3 (5) (1999) 363–376.

[17] A. Kandel, Fuzzy Expert Systems, CRC Press, Boca Raton, 1992, pp. 8–19.

[18] H. Mannila, Methods and problems in data mining, The International Conference on Database Theory, 1997.

[19] R. Srikant, 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.

[20] C.H. Wang, J.F. Liu, T.P. Hong, S.S. Tseng, A fuzzy inductive learning strategy for modular rules, Fuzzy Sets Syst. 103 (1) (1999) 91–105.

[21] S. Yue, E. Tsang, D. Yeung, D. Shi, Mining fuzzy association rules with weighted items, The IEEE International Conference on Systems, Man and Cybernetics, 2000, pp. 1906–1911.

[22] L.A. Zadeh, Fuzzy sets, Inform. Control 8 (3) (1965) 338–353.