Mining fuzzy generalized association rules from quantitative data under fuzzy taxonomic structures

_Tzung-Pei Hong*
Department of Electrical Engineering
National University of Kaohsiung
Kaohsiung, 811, Taiwan, R.O.C.
E-mail: tphong@nuk.edu.tw
Fax: +886+7+5919138

_Kuei-Ying Lin_
Institute of Information Engineering
I-Shou University
Kaohsiung, 840, Taiwan, R.O.C.
E-mail: m883331m@isu.edu.tw
Fax: +886+7+6577711

_Shyue-Liang Wang_
Department of Information Management
National University of Kaohsiung
Kaohsiung, 811, Taiwan, R.O.C.
E-mail: slwang@nuk.edu.tw
Fax: +886+7+5919000

* Corresponding author.
Abstract

Due to the increasing use of very large databases and data warehouses, mining useful information and helpful knowledge from transactions has become an important research area. Most conventional data-mining algorithms identify the relationships among transactions using binary values and find rules at a single concept level. Transactions with quantitative values and items with taxonomic relations are, however, commonly seen in real-world applications. Besides, the taxonomic structures may not be crisp. This paper thus proposes a fuzzy data-mining algorithm for extracting fuzzy generalized association rules under given fuzzy taxonomic structures. The proposed algorithm first generates expanded transactions according to given fuzzy taxonomic structures. It then transforms each quantitative value into a fuzzy set in linguistic terms. 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 mining process based on fuzzy counts is then performed to find fuzzy generalized association rules from these items. The algorithm can therefore focus on the most important linguistic terms and reduce its time complexity.

Keywords: data mining, fuzzy set, fuzzy taxonomic structure, generalized association rule, quantitative value.
1. Introduction

The rapid development of computer technology, especially increased capacities and decreased costs of storage media, has led businesses to store huge amounts of external and internal information in large databases at low cost. Mining useful information and helpful knowledge from these large databases has thus evolved into an important research area [2][4]. Years of effort in data mining has produced a variety of efficient techniques. Depending on the types of databases to be processed, mining approaches may be classified as working on transactional databases, temporal databases, relational databases, and multimedia databases, among others. Depending on the classes of knowledge sought, mining approaches may be classified as finding association rules, classification rules, clustering rules, and sequential patterns, among others [4].

Deriving association rules from transaction databases is most commonly seen in data mining [1][2][4]. It discovers relationships among items such that the presence of certain items in a transaction tends to imply the presence of certain other items. Most previous studies concentrated on showing how binary-valued transaction data on a single level of items may be handled. However, transaction data in real-world applications usually consist of quantitative values and items are often organized in a taxonomy, so designing a sophisticated data-mining algorithm able to deal with
quantitative data on multiple levels of items presents a challenge to workers in this
research field.

In the past, Agrawal and his co-workers proposed several mining algorithms for
finding association rules in transaction data based on the concept of large itemsets
[1-2, 13]. They also proposed a method [14] for mining association rules from data
sets using quantitative and categorical attributes. Their proposed method first
determined the number of partitions for each quantitative attribute, and then mapped
all possible values of each attribute onto a set of consecutive integers. Other methods
have also been proposed to handle numeric attributes and to derive association rules.
Fukuda et al. introduced the optimized association-rule problem and permitted
association rules to contain single uninstantiated conditions on the left-hand side [5].
They also proposed schemes for determining conditions under which rule confidence
or support values were maximized. However, their schemes were suitable only for
single optimal regions. Rastogi and Shim extended the approach to more than one
optimal region, and showed that the problem was NP-hard even for cases involving
one uninstantiated numeric attribute [13][14].

Recently, the fuzzy set theory has been used more and more frequently in
intelligent systems because of its simplicity and similarity to human reasoning [11].
Several fuzzy learning algorithms for inducing rules from given sets of data have been
designed and used to good effect with specific domains. Strategies based on decision trees were proposed in [12, 18-19]. Wang et al. proposed a fuzzy version-space learning strategy for managing vague information [17]. Hong et al. also proposed a fuzzy mining algorithm for managing quantitative data [6].

In [10], we proposed a data-mining algorithm able to deal with quantitative data under a crisp taxonomic structure. In that approach, each item definitely belongs to only one ancestor in the taxonomic structure. The taxonomic structures may, however, not be crisp in real-world applications. An item may belong to different classes in different views. This paper thus proposes a new fuzzy data-mining algorithm for extracting fuzzy generalized association rules under given fuzzy taxonomic structures. The proposed algorithm first generates expanded transactions according to given fuzzy taxonomic structures. It then transforms each quantitative value into a fuzzy set in linguistic terms. 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 mining process based on fuzzy counts is then performed to find fuzzy generalized association rules from these items. The algorithm can therefore focus on the most important linguistic terms and reduce its time complexity. The rules mined are expressed in linguistic terms, which are more natural and understandable for human beings.
The remaining parts of this paper are organized as follows. Data mining with fuzzy taxonomic structures is described in Section 2. A novel data-mining algorithm for quantitative values under fuzzy taxonomic structures is proposed in Section 3. An example to illustrate the proposed algorithm is given in Section 4. Experimental results are shown in Section 5. Conclusions and future works are stated in Section 6.

2. Data mining with a fuzzy taxonomic structure

Previous studies on data mining focused on finding association rules on a single-concept level. Mining multiple-concept-level rules may, however, lead to discovery of more general and important knowledge from data. Relevant taxonomies of data items are thus usually predefined in real-world applications. An item may, however, belong to different classes in different views. When taxonomic structures are not crisp, hierarchical graphs can be used to represent them. Terminal nodes on the hierarchical graphs represent the items actually appearing in transactions; internal nodes represent classes or concepts formed by lower-level nodes. A simple example is given in Figure 1.
Figure 1. An example of fuzzy taxonomic structures

In this example, vegetarian diet falls into two classes: fruit and vegetable. Fruit can be further classified into apple and tomato. Similarly, assume vegetable is divided into tomato and cabbage. Note that tomato belongs to both fruit and vegetable with different membership degrees. It is thought of as fruit with 0.9 membership value and as vegetable with 0.7. The membership value of tomato belonging to vegetarian diet can be calculated using the max-min product combination. Since both fruit and vegetable belong to vegetarian diet with membership value 1, the membership value of tomato belonging to vegetarian diet is then \( \max(min(1, 0.9), min(1, 0.7)) = 0.9 \). Only the terminal items (apple, tomato, cabbage, pork and beef) can appear in transactions. The membership degrees of ancestors for each terminal node are shown in Table 1.
Table 1. The membership degrees of ancestors for each terminal node in this example

<table>
<thead>
<tr>
<th>Terminal node</th>
<th>Membership values of ancestors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple</td>
<td>1/Fruit, 1/Vegetarian-Diet</td>
</tr>
<tr>
<td>Tomato</td>
<td>0.9/Fruit, 0.7/Vegetable, 0.9/Vegetarian-Diet</td>
</tr>
<tr>
<td>Cabbage</td>
<td>1/Vegetable, 1/Vegetarian-Diet</td>
</tr>
<tr>
<td>Pork</td>
<td>1/Meat</td>
</tr>
<tr>
<td>Beef</td>
<td>1/Meat</td>
</tr>
</tbody>
</table>

Wei and Chen proposed a method to find generalized association rules under fuzzy taxonomic structures [16]. The items to be processed in their approach are binary. Their mining process first calculated the membership values of ancestors for each terminal node in the manner mentioned above. It then added the ancestors of items to the given transactions as Srikant and Agrawal’s did [15]. Candidate itemsets were generated and counted by scanning the expanded transaction data. If the number of an itemset appearing in the expanded transactions was larger than a pre-defined threshold value (called the minimum support), the itemset was considered a large itemset. Itemsets containing only single items were first processed. The 1-large itemsets derived were then combined to form candidate itemsets containing two items. This process was repeated until all large itemsets had been found. After that, all possible generalized association rules were induced from the large itemsets. The rules with confidence values larger than a predefined threshold (called the minimum confidence) were thus kept. Uninteresting association rules were then pruned away.
and interesting rules were output. An interesting rule must satisfy at least one of the following three interest requirements:

1. a rule with no ancestor rules (by replacing the items in a rule with their ancestors in the taxonomy) mined out,

2. the support value of a rule being $R$-time larger than the expected support values of its ancestor rules, and

3. the confidence value of a rule being $R$-time larger than the expected confidence values of its ancestor rules.

Wei and Chen’s concepts will be used in our approach to mine fuzzy interesting generalized association rules from quantitative transaction data. The rules mined are expressed in linguistic terms, which are more natural and understandable for human beings.

3. Mining fuzzy generalized association rules from quantitative data under fuzzy taxonomic structure

The proposed generalized mining algorithm integrates fuzzy-set concepts and generalized data mining technologies to find cross-level interesting rules from quantitative data. The quantitative items may be from any level of the given fuzzy
taxonomy. The knowledge derived is represented by fuzzy linguistic terms, and thus easily understandable by human beings.

The proposed fuzzy mining algorithm first calculates the membership values of ancestors for each terminal node as Wei and Chen’s approach did [16]. It then forms expanded transactions and uses membership functions to transform each quantitative value into a fuzzy set in linguistic terms. It adopts an iterative search approach to finding large itemsets. 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 the number of original items. The algorithm therefore focuses on the most important linguistic terms, which reduces its time complexity. A mining process using fuzzy counts is performed to find fuzzy multiple-level association rules. Fuzzy interest requirements are then checked to remove uninteresting rules. Details of the proposed fuzzy mining algorithm are stated below.

**The fuzzy generalized mining algorithm for fuzzy taxonomic structures:**

**INPUT:** A body of $n$ quantitative transaction data, a set of membership functions, a fuzzy taxonomic structure, a minimum support value $\alpha$, a minimum confidence value $\lambda$, and an interest threshold $R$.

**OUTPUT:** A set of fuzzy generalized association rules.

**STEP 1:** Calculate the membership values of ancestors of each terminal node from the
given fuzzy taxonomic structure.

STEP 2: Calculate the quantitative value $v_{ik}$ of each ancestor item $A_k$ in transaction datum $D_i$ ($I=1$ to $n$) as:

$$v_{ik} = \sum_{T_j \in D_i} v_{ij} \cdot \mu_{A_k}(T_j),$$

where $T_j$ is a terminal item appearing in $D_i$, $v_{ij}$ is the quantitative value of $T_j$, and $\mu_{A_k}(T_j)$ is the membership value of item $T_j$ belonging to ancestor $A_k$.

STEP 3: Add ancestors of appearing items to transactions with their quantities calculated from STEP 2.

STEP 4: Transform the quantitative value $v_{ij}$ of each expanded item name $I_j$ (terminal item or ancestor item) appearing in transaction datum $D_i$ ($I=1$ to $n$) into a fuzzy set $f_{ij}$ represented as

$$f_{ij} = \left( \frac{f_{ij1}}{R_{j1}} + \frac{f_{ij2}}{R_{j2}} + \ldots + \frac{f_{ijh_j}}{R_{jh_j}} \right)$$

using the given membership functions, where $h_j$ is the number of fuzzy regions for $I_j$, $R_{jl}$ is the $l$-th fuzzy region of $I_j$, $1 \leq l \leq h_j$, and $f_{ijl}$ is $v_{ij}$’s fuzzy membership value in region $R_{jl}$.

STEP 5: Calculate the scalar cardinality $count_{jl}$ of each fuzzy region $R_{jl}$ in the transaction data as:

$$count_{jl} = \sum_{i=1}^{n} f_{ijl}.$$

STEP 6: Find $\text{max-} count_j = \max_{l=1}^{h_j} \{ count_{jl} \}$, for $j = 1$ to $m$, where $m$ is the number of expanded items. Let $\text{max-} R_j$ be the region with $\text{max-} count_j$ for item $I_j$, which
will be used to represent the fuzzy characteristic of item \(I_j\) in later mining processes.

STEP 7: Check whether the value \(\text{max-count}_j\) of a region \(\text{max}-R_j\), \(j = 1\) to \(m\), is larger than or equal to the predefined minimum support value \(\alpha\). If a region \(\text{max}-R_j\) is equal to or greater than the minimum support value, put it in the large 1-itemsets \((L_1)\). That is,

\[
L_1 = \{\text{max}-R_j \mid \text{max-count}_j \geq \alpha, 1 \leq j \leq m\}.
\]

STEP 8: Generate the candidate set \(C_2\) from \(L_1\). Each 2-itemset in \(C_2\) must not include items with ancestor or descendant relations in the fuzzy taxonomy.

STEP 9: For each newly formed candidate 2-itemset \(s\) with items \((s_1, s_2)\) in \(C_2\):

(a) Calculate the fuzzy value of \(s\) in each transaction datum \(D_i\) as \(f_{is} = f_{is_1} \land f_{is_2}\), where \(f_{is_j}\) is the membership value of \(D_i\) in region \(s_j\). If the minimum operator is used for the intersection, then \(f_{is} = \min(f_{is_1}, f_{is_2})\).

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

\[
\text{count}_s = \sum_{i=1}^{n} f_{is}.
\]

(c) If \(\text{count}_s\) is larger than or equal to the predefined minimum support value \(\alpha\), put \(s\) in the large 2-itemsets \((L_2)\).

STEP 10: IF \(L_2\) is null, then exit the algorithm; otherwise, do the next step.

STEP 11: Set \(r = 2\), where \(r\) is used to represent the number of items stored in the
current large itemsets.

STEP 12: Generate the candidate set $C_{r+1}$ from $L_r$ in a way similar to that in the \textit{apriori} algorithm [1]. That is, the algorithm first joins $L_r$ and $L_r$ assuming that $r$ items in the two itemsets are the same and the other one is different. Store in $C_{r+1}$ itemsets having all their sub-$r$-itemsets in $L_r$.

STEP 13: For each newly formed $(r+1)$-itemset $s$ with items $(s_1, s_2, \ldots, s_{r+1})$ in $C_{r+1}$:

(a) Calculate the fuzzy value of $s$ in each transaction datum $D_i$ as

$$f_{is} = f_{is_1} \Lambda f_{is_2} \Lambda \ldots \Lambda f_{is_{r+1}},$$

where $f_{is_j}$ is the membership value of $D_i$ in region $s_j$. If the minimum operator is used for the intersection, then

$$f_{is} = \min_{j=1}^{r+1} f_{is_j}.$$

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

$$\text{count}_s = \sum_{i=1}^{n} f_{is}.$$

(c) If $\text{count}_s$ is larger than or equal to the predefined minimum support value $\alpha$, put $s$ in $L_{r+1}$.

STEP 14: If $L_{r+1}$ is null, then do the next step; otherwise, set $r=r+1$ and repeat STEPs 12 to 14.

STEP 15: Construct the association rules for all the large $q$-itemset $s$ containing items $(s_1, s_2, \ldots, s_q)$, $q \geq 2$, using the following substeps:
(a) Form all possible association rules thusly:

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

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

\[
\frac{\sum_{i=1}^q f_{it}}{\sum_{i=1}^q (f_{is_1} \Lambda \ldots \Lambda f_{is_{k-1}}, f_{is_{k+1}} \Lambda \ldots \Lambda f_{is_q})}
\]

STEP 16: Keep the rules with confidence values larger than or equal to the predefined confidence threshold \( \lambda \).

STEP 17: Output the rules without ancestor rules (by replacing the items in a rule with their ancestors in the taxonomy) to users as interesting rules.

STEP 18: For each remaining rule \( s \) (represented as \( s_1 \Lambda s_2 \Lambda \ldots \Lambda s_r \rightarrow s_{(r+1)} \)), find the close ancestor rule \( t \) (represented as \( t_1 \Lambda t_2 \Lambda \ldots \Lambda t_r \rightarrow t_{(r+1)} \)) and calculate the support interest measure \( I_{\text{support}}(s) \) of \( s \) as:

\[
I_{\text{support}}(s) = \frac{\text{count}_s}{\prod_{k=1}^{r+1} \text{count}_{is_k} \times \text{count}_s},
\]

and the confidence interest measure \( I_{\text{confidence}}(s) \) of \( s \) as:

\[
I_{\text{confidence}}(s) = \frac{\text{confidence}_s}{\text{count}_{s_{(r+1)}} \times \text{confidence}_{t_{(r+1)}}},
\]

where \( \text{confidence}_s \) and \( \text{confidence}_{t} \) are respectively the confidence values of rules \( s \) and \( t \); output the rules with their support interest measures or confidence interest measures larger than or equal to the predefined interest.
threshold \( R \) to users as interesting rules.

Note that in Step 17, an ancestor of a fuzzy rule is formed by replacing the items in the rule with their ancestors in the fuzzy taxonomy, but the linguistic terms in both the rules must be the same. The rules output from Steps 17 and 18 can then serve as meta-knowledge concerning the given transactions.

4. An example

In this section, an example is given to illustrate the proposed fuzzy generalized mining algorithm. This is a simple example to show how the proposed algorithm generates fuzzy generalized association rules from quantitative transactions under a fuzzy taxonomic structure. The data set includes the six transactions shown in Table 2.

<table>
<thead>
<tr>
<th>Transaction ID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(Apple, 3) (Tomato, 4) (Beef, 2)</td>
</tr>
<tr>
<td>2</td>
<td>(Tomato, 7) (Cabbage, 7) (Beef, 7)</td>
</tr>
<tr>
<td>3</td>
<td>(Tomato, 2) (Cabbage, 10) (Beef, 5)</td>
</tr>
<tr>
<td>4</td>
<td>(Cabbage, 9) (Beef, 10)</td>
</tr>
<tr>
<td>5</td>
<td>(Apple, 7) (Pork, 8)</td>
</tr>
<tr>
<td>6</td>
<td>(Apple, 2) (Tomato, 8)</td>
</tr>
</tbody>
</table>

Each transaction includes a transaction ID and some purchased items. Each item is represented by a tuple (item name, item amount). For example, the fourth
transaction consists of nine units of cabbage and ten units of beef. Assume the predefined fuzzy taxonomy is as shown in Figure 1. For convenience, the simple symbols in Table 3 are used to represent the items and groups.

Table 3. Items and groups are represented by simple symbols for convenience

<table>
<thead>
<tr>
<th>Items</th>
<th>Symbol</th>
<th>Groups</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple</td>
<td>A</td>
<td>Fruit</td>
<td>$T_1$</td>
</tr>
<tr>
<td>Tomato</td>
<td>B</td>
<td>Vegetable</td>
<td>$T_2$</td>
</tr>
<tr>
<td>Cabbage</td>
<td>C</td>
<td>Vegetarian Diet</td>
<td>$T_3$</td>
</tr>
<tr>
<td>Pork</td>
<td>D</td>
<td>Meat</td>
<td>$T_4$</td>
</tr>
<tr>
<td>Beef</td>
<td>E</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Also assume that the fuzzy membership functions are the same for all the items and are as shown in Figure 2.

![Figure 2. The membership functions used in this example](image)

In the example, amounts are represented by three fuzzy regions: Low, Middle and High. Thus, three fuzzy membership values are produced for each item amount.
according to the predefined membership functions. For the transaction data in Table 2, all the expanded transactions are shown in Table 4.

<table>
<thead>
<tr>
<th>Transaction ID</th>
<th>Expanded Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(A, 3)(B, 4)(E, 2)(T₁, 6.6)(T₂, 2.8)(T₃, 6.6)(T₄, 2)</td>
</tr>
<tr>
<td>2</td>
<td>(B, 7)(C, 7)(E, 7)(T₁, 6.3)(T₂, 11.9)(T₃, 13.3)(T₄, 7)</td>
</tr>
<tr>
<td>3</td>
<td>(B, 2)(C, 10)(E, 5)(T₁, 1.8)(T₂, 11.4)(T₃, 11.8)(T₄, 5)</td>
</tr>
<tr>
<td>4</td>
<td>(C, 9)(E, 10)(T₂, 9)(T₃, 9)(T₄, 10)</td>
</tr>
<tr>
<td>5</td>
<td>(A, 7)(D, 8)(T₁, 7)(T₃, 7)(T₄, 8)</td>
</tr>
<tr>
<td>6</td>
<td>(A, 2)(B, 8)(T₁, 9.2)(T₂, 5.6)(T₃, 9.2)</td>
</tr>
</tbody>
</table>

The quantitative values of the expanded items are represented using fuzzy sets, which are shown in Table 5, where the notation item.term is called a fuzzy region.

<table>
<thead>
<tr>
<th>TID</th>
<th>Fuzzy set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Low</td>
</tr>
<tr>
<td>2</td>
<td>Middle</td>
</tr>
<tr>
<td>3</td>
<td>Low</td>
</tr>
<tr>
<td>4</td>
<td>Low</td>
</tr>
<tr>
<td>5</td>
<td>Low</td>
</tr>
<tr>
<td>6</td>
<td>Low</td>
</tr>
</tbody>
</table>

The fuzzy region with the highest count among the three possible regions for each item is found. Thus, "Low" is chosen for A, "Middle" is chosen for B, D, E, T₁
and $T_4$, and "High" is chosen $C$, $T_2$ and $T_3$.

The count of each region selected is then checked against the predefined minimum support value $\alpha$. Assume in this example, $\alpha$ is set at 1.5. $L_1$ is shown in Table 6.

Table 6. The set of large 1-itemsets in this example

<table>
<thead>
<tr>
<th>Itemset</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.Middle</td>
<td>2.2</td>
</tr>
<tr>
<td>C.High</td>
<td>1.6</td>
</tr>
<tr>
<td>E.Middle</td>
<td>2.0</td>
</tr>
<tr>
<td>$T_1$.Middle</td>
<td>3.14</td>
</tr>
<tr>
<td>$T_2$.High</td>
<td>2.6</td>
</tr>
<tr>
<td>$T_3$.High</td>
<td>3.56</td>
</tr>
<tr>
<td>$T_4$.Middle</td>
<td>2.6</td>
</tr>
</tbody>
</table>

Similarly, $L_2$ includes $(E.Middle, T_2.High)$, $(E.Middle, T_3.High)$, $(T_1. Middle, T_4.Middle), (T_2.High, T_4.Middle)$ and $(T_3.High, T_4.Middle)$. $L_3$ is null.

Assume the given confidence threshold $\lambda$ is set at 0.7. The following three rules are thus kept:

If $E = Middle$, then $T_2 = High$, with a support value of 1.8 and a confidence value of 0.9;
If $E = Middle$, then $T_3 = High$, with a support value of 1.92 and a confidence value of 0.96;
If $T_4 = Middle$, then $T_3 = High$, with a support value of 2.12 and a confidence value of 0.82

Since the rule “If $T_4 = Middle$, then $T_3 = High$” have no ancestor rules mined
out, it is thus output as an interesting rule. Assume the given interest threshold $R$ is set at 3. Since the support and the confidence interest measures of the remaining two rules are less than 3, they are thus not interesting rules.

5. Experimental Results

The section reports on experiments made to show the effect of the proposed approach. They were implemented in C on a Pentium-III 700 Personal Computer. The number of levels was set at 3. There were 64 purchased items (terminal nodes) on level 3, 16 generalized items on level 2, and 4 generalized items on level 1. Each non-terminal node had four branches. Only the terminal nodes could appear in transactions. Totally 10000 transactions with an average of 12 purchased items in each transaction are generated in each run. The relationships between numbers of rules mined and minimum support values for minimum confidence value set at 0.7 are shown in Figure 3.
Figure 3. The relationships between numbers of rules mined and minimum supports

From Figure 3, it is easily seen that numbers of rules mined decrease along with increase of minimum support values. This is quite consistent with our intuition.

6. Conclusions and future works

In this paper, we have proposed a fuzzy generalized mining algorithm for processing transaction data with quantitative values and discovering interesting generalized association rules among them. The rules thus mined out exhibit quantitative regularity under given fuzzy taxonomic structures and can be used to provide suggestions to appropriate supervisors. Compared to conventional crisp-set mining methods for quantitative data, our approach gets smoother mining results due to its fuzzy membership characteristics.
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 membership functions are known in advance. In [7-9], we proposed some fuzzy learning methods to automatically derive membership functions. We will therefore attempt to dynamically adjust the membership functions in the proposed fuzzy generalized mining algorithm to avoid the bottleneck of membership function acquisition. We will also attempt to design specific data-mining models for various problem domains.

Acknowledgment

The authors would like to thank the anonymous referees for their very constructive comments. This research was supported by the National Science Council of the Republic of China under contract NSC 90-2213-E-390-001.

References


perspective,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No. 6, 1993, pp. 914-925.


