Abstract

Background/Objectives: The objective of this paper is to find the relationships among the time of events in temporal quantitative databases. Methods/Statistical analysis: The fuzzy sets are applied to both quantitative attributes of database and time distance among events. Then, Apriori algorithm with improvement is used to find out all frequent sequences. Final, the rules are pulled out from the sequences. Findings: The finding rules from the proposed algorithm help to predict quantity and occur time of items from other previous items. A rule can be a form “If large number of computers is sold, small number of modems will be bought in a medium period”. In this paper, experiments are made to show performance of the proposed algorithm. Applications/Improvements: The findings are useful to forecast in market basket analysis, stock market analysis.

Keywords: Association Rule, Data Mining, Fuzzy Sets, Fuzzy Time-interval, Temporal Quantitative Database

1. Introduction

Association rule mining problems are an important object in data mining research. Agrawal et al\(^1\) first introduced an association rule-mining problem in transaction databases in 1993 and grew so far\(^2\)–\(^7\).

Fuzzy logic was introduced in 1965\(^8\). It was applied to transform quantitative attributes in database into fuzzy sets by authors\(^9\)–\(^11\). This approach helped to solve unsuitable of converting quantitative attributes to binary values that causes of the sharp boundary problems. Furthermore, the researches of fuzzy association rules are continued\(^12\)–\(^15\).

The time distinct among items in mining association rules were been attentive\(^16\)–\(^23\). The problem was first presented in 2003 by Chen et al\(^22\). The concept of fuzzy sets was used to extend the original work, thus, fuzzy time-interval sequential patterns could be extracted from databases\(^23\). Main idea of the work is brief: Time-interval items were applied by fuzzy sets; Frequently sequential patterns with forms (A, I\(_1\), B, I\(_2\), C), where A, B, C are attributes, I\(_1\), I\(_2\) are linguistic term of time-interval, were mined; The FTI-Apriori algorithm and FTI-Prefix Span algorithm were proposed to mining the fuzzy time-interval sequential patterns.

However, the research in\(^23\) only applied for sequence databases without quantitative attributes, it not used with temporal quantitative databases. An instance of the result of rules like that “after buying a computer, a customer will return to buy a modem in a medium period”.

The main objective of this study is to mine fuzzy time-interval association rules from temporal quantitative databases. An example of the rules is as “if large number of computers is sold, small number of modems will be bought in a medium period”. The rules do not care about objects that have their transactions. In this paper, FTQ algorithm is presented. FTQ stands for Fuzzy Time-interval in temporal Quantitative databases. The algorithm is developed from Apriori algorithm\(^5\), that joining two frequent sequences with length of k to a frequent sequence with length of k+1.

*Author for correspondence
The remaining parts of this paper are organized as follows. Defining the problems is presented in section II. FTQ algorithm is shown in section 3. Section 4 presents results. Conclusions are drawn in section 5.

2. Problem Definition

Definition 1: Let $E=\{e_1, e_2, \ldots, e_u\}$ be set of all attributes, $T_i=\{e_{1}(i), e_{2}(i), \ldots, e_{u}(i)\}$ ($1 \leq i \leq n$) be set of values of $E$ at time $i$, $e_{k}(i)$ is value of $e_{k}$ at time $i$ ($1 \leq k \leq u$), $e_{k}(i)$ is a quantitative value. $D = \{T_1, T_2, \ldots, T_n\}$ is called a temporal quantitative database.

Example 1: An example of temporal quantitative database

$DF = (T, E, FE)$ is called a temporal fuzzy database and fuzzy set $f_{k,j}^{\alpha}$ is called a fuzzy attribute. Each fuzzy set has a membership function $\mu: X \rightarrow [0,1]$.

Example 2: Let fuzzy set be in $[25]^{24}$ with $K=3$ for all attributes in database of example 1, fuzzy membership functions are determined as follows:

$$\mu_{k, a}^{\alpha} (val) = \max \left\{ 1 - \frac{\text{val} - a_{k}^{\alpha}}{b_{k}^{\alpha}}, 0 \right\}$$
(1)

where

$$a_{k}^{\alpha} = m + (ma-mi)(i_m-1)/(K-1)$$

$$b_{k}^{\alpha} = (ma-mi)/(K-1)$$

$\mu_{k,a}^{\alpha}$ is the fuzzy membership function, which links to $i_k$th fuzzy set of $x_a$ attribute; $x_a$ is the $m^{th}$ attribute in $E$.

Table 1. An example of temporal quantitative database

| Time | Values |
|------|--------|
| 1    | $e_{1}(2), e_{5}(5)$ |
| 2    | $e_{1}(5), e_{2}(7), e_{4}(1), e_{6}(6)$ |
| 3    | $e_{1}(4), e_{10}(5)$ |
| 4    | $e_{2}(2), e_{3}(6)$ |
| 5    | $e_{3}(3), e_{1}(1), e_{3}(3)$ |
| 6    | $e_{5}(5)$ |
| 11   | $e_{1}(1), e_{2}(2), e_{4}(4)$ |
| 12   | $e_{5}(5)$ |
| 18   | $e_{3}(2)$, $e_{1}(1)$ |
| 19   | $e_{3}(3)$ |
| 20   | $e_{1}(1)$ |
| 22   | $e_{6}(6)$ |
| 25   | $e_{2}(2)$ |
| 29   | $e_{5}(5)$ |
| 31   | $e_{1}(1)$ |

a. An example of temporal quantitative database

where,

$E=\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}\}$ is a set of attributes; $e_i$ is an attribute; $T_{11}=\{e_{1}(1), e_{2}(2), e_{4}(4)\}$ are values of $E$ at time $11$; $e_{1}(1), e_{4}(2), e_{9}(4)$ represent values of $e_{1}, e_{4}, e_{9}$ attributes at the time.

Definition 2: Let $T$ be a set of transactions, $E$ be a set of attributes and $FE = \{F_{e_{1}}, F_{e_{2}}, \ldots, F_{e_{u}}\}$ be a set of fuzzy sets, which link to each of attributes in $E$, $F_{e_{k}} = \{f_{k,1}^{\alpha}, f_{k,2}^{\alpha}, \ldots, f_{k,h_{k}}^{\alpha}\}$ be fuzzy sets link to $e_{k}$ attribute ($k=1, \ldots, u$), with $f_{k,j}^{\alpha}$ be a $j^{th}$ fuzzy set of attribute $e_{k}$ ($1 \leq j \leq h_{k}$).

Table 2. Temporal fuzzy database

| Time | Fuzzy attributes |
|------|-----------------|
| 1    | $f_{1,2}^{0.5}, f_{1,2}^{0.5}, f_{1,2}^{0.5}$ (1) |
| 2    | $f_{1,1}^{0.5}, f_{1,1}^{0.5}, f_{1,1}^{0.5}$ (1) |
| 3    | $f_{1,2}^{0.5}, f_{1,2}^{0.5}, f_{1,2}^{0.5}$ (1) |
| 4    | $f_{2,3}^{0.667}, f_{2,3}^{0.333}, f_{2,3}^{0.333}$ (1) |
| 5    | $f_{2,3}^{0.667}, f_{2,3}^{0.333}$ (1) |
| 6    | $f_{2,3}^{0.667}, f_{2,3}^{0.333}$ (1) |
| 11   | $f_{2,3}^{0.667}, f_{2,3}^{0.333}$ (1) |
| 12   | $f_{2,3}^{0.667}, f_{2,3}^{0.333}$ (1) |
| 18   | $f_{2,3}^{0.667}, f_{2,3}^{0.333}$ (1) |
| 19   | $f_{2,3}^{0.667}, f_{2,3}^{0.333}$ (1) |
| 20   | $f_{2,3}^{0.667}, f_{2,3}^{0.333}$ (1) |
| 22   | $f_{2,3}^{0.667}, f_{2,3}^{0.333}$ (1) |
| 25   | $f_{2,3}^{0.667}, f_{2,3}^{0.333}$ (1) |
| 29   | $f_{2,3}^{0.667}, f_{2,3}^{0.333}$ (1) |
| 31   | $f_{2,3}^{0.667}, f_{2,3}^{0.333}$ (1) |

b. Temporal fuzzy database
K is the number of partitions in an attribute, \(i_m\) is the \(i^{th}\) fuzzy set, \(m, ma, min, max\) of \(X_j\). Result of transforming the source database into temporal fuzzy database DF as follows:

In the Table 2, \(f_{k,n}^{\text{val}}(val)\) is value of the fuzzy attribute, \(f_{k,n}^{\text{val}}\) is the transformed fuzzy value. An example, \(f_{3,1}^{\text{val}}(0.5)\) at time 1 is mean fuzzy value 0.5 of the first of three of the \(e_i\) attribute.

Definition 3: \(P\) is a fuzzy sequence with a form \(((p_1, t_1), (p_2, t_2), \ldots, (p_p, t_p))\) where \(p_i\) is a fuzzy attribute and \(t_i\) is time of \(p_i\) \((1 \leq i \leq n; t_1 \leq t_i \leq t_n)\). Sequence \(A\) is sorted by alphabet in case of events occurs at the same time. Then the interval-time values between two continuous fuzzy items in a sequence is \(t_i = t_{i+1} - t_i (1 \leq i \leq n-1)\). For example, if \(P = ((p_1, t_1), (p_2, t_2), (p_3, t_3))\), then their interval-time values are \(t_1 = 4 - 1 = 3\) and \(t_2 = 29 - 4 = 25\).

Given the set \(L_T = \{\text{lt}_j\}_{j=1,2,...,p}\) of linguistic terms, we use \(\mu_{\text{lt}}(t_i)\) to denote the membership degree of time-interval value \(t_i\) to linguistic term \(\text{lt}_j\). \(\mu_{\text{lt}}(t_i) : X[0,1]\) for.

Example 3: \(L_T = \{\text{Short}, \text{Medium}, \text{Long}\}\) and fuzzy membership functions for time-interval concept are as follows:

\[
\mu_{\text{Short}}(t_i) = \begin{cases} 
1, & t_i \leq 2 \\
\frac{15 - t_i}{13}, & 2 < t_i < 15 \\
0, & t_i \geq 15
\end{cases}
\]

\[
\mu_{\text{Medium}}(t_i) = \begin{cases} 
0, & t_i \leq 2 \text{ or } t_i \geq 28 \\
\frac{t_i - 2}{13}, & 2 < t_i < 15 \\
\frac{28 - t_i}{13}, & 15 < t_i < 28
\end{cases}
\]

\[
\mu_{\text{Long}}(t_i) = \begin{cases} 
0, & t_i \leq 15 \\
\frac{t_i - 15}{13}, & 15 < t_i < 28 \\
1, & t_i \geq 28
\end{cases}
\]

Definition 4: Let \(FE\) be a set of fuzzy sets, which link to attributes in a temporal quantitative database; \(L_T = \{\text{lt}_j\}_{j=1,2,...,p}\) be linguistic terms. The sequence \(\beta = \{(q_1, \text{lt}_1), (q_2, \text{lt}_2), \ldots, (q_r, \text{lt}_r), (q_e, \text{lt}_e)\}\) is a fuzzy time-interval sequence if \(q_i \in \text{FS} \ ) and \(\text{lt}_j \in L_T\) for \(1 \leq j \leq r-1\) and \(q_e \in \text{FE}\). For instance, \(\beta = (f_{3,1}^{\text{val}}, \text{Short}, f_{3,1}^{\text{val}}, \text{Medium}, f_{3,1}^{\text{val}})\) is a fuzzy time-interval sequence.

Definition 5: A fuzzy time-interval sequence \(a = (p_1, \text{ltp}_1, p_2, \text{ltp}_2, \ldots, p_k, \text{ltp}_k, p_1)\) is a subsequence of a fuzzy time-interval sequence \(\beta = (q_1, \text{ltq}_1, q_2, \text{ltq}_2, \ldots, q_r, \text{ltq}_r, q_e)\) if exist a integer value \(w\) satisfied with \(p_i = q_w\) for \(k=1\). For example, the fuzzy time-interval sequence \((f_{3,1}^{\text{val}}, \text{Short}, f_{3,1}^{\text{val}})\) and \((f_{3,1}^{\text{val}}, \text{Short}, f_{3,1}^{\text{val}}, \text{Medium}, f_{3,1}^{\text{val}})\) are two subsequences of \((f_{3,1}^{\text{val}}, \text{Short}, f_{3,1}^{\text{val}}, \text{Medium}, f_{3,1}^{\text{val}})\).

Definition 6: A fuzzy time-interval association rule has a form \(X \rightarrow Y\), where \(X\) is a fuzzy time-interval sequence, \(Y\) is a fuzzy attribute in DF, \(\text{lt}_e \in L_T\). For example, \(f_{3,1}^{\text{val}} \rightarrow f_{3,1}^{\text{val}}\) (\(\text{Short}\)) is a fuzzy time-interval association rule.

Let \(P = ((p_1, t_1), (p_2, t_2), \ldots, (p_p, t_p))\) be a fuzzy sequence and \(\alpha = (p_1, \text{lt}_1, p_2, \text{lt}_2, \ldots, p_p, \text{lt}_p, p_1)\) be a fuzzy time-interval sequence, \(p_i(t)\) is the value of \(p_i\) at time \(t_i\). The support of \(P\) in \(\alpha\) is defined as follows:

\[
\gamma_P(\alpha) = \begin{cases} 
\min_{1 \leq i \leq r} \{\mu_{\text{lt}}(t_{i+1} - t_i)\} \times \min_{1 \leq i \leq p} \{\mu_{\text{lt}}(t_i)\} & \text{if } r = 1 \\
\min_{1 \leq i \leq r} \{\mu_{\text{lt}}(t_{i+1} - t_i)\} \times \min_{1 \leq i \leq p} \{\mu_{\text{lt}}(t_i)\} & \text{if } r > 1
\end{cases}
\]

Example: Given a fuzzy sequence \(P = (f_{3,1}^{\text{val}}, 6), (f_{3,1}^{\text{val}}, 12), (f_{3,1}^{\text{val}}, 31)\), a fuzzy time-interval sequence \(\alpha = (f_{3,1}^{\text{val}}, \text{Short}, f_{3,1}^{\text{val}}, \text{Medium}, f_{3,1}^{\text{val}})\). The support of \(P\) in \(\alpha\) is \(\min\{\mu_{\text{Short}}(6), \mu_{\text{Medium}}(18)\} \times \min\{0.667,1,1\} = \min\{0.667\} \times \min\{0.667,1\} = 0.667 \times 0.667 = 0.461\) (the fuzzy membership functions in definition 2).

Given a temporal fuzzy database DF with \(N\) transactions, we have definitions:

Support of \(\alpha\) in DF is defined as follows:

\[
\text{Supp}(\alpha) = \sum_{P \in DF} \gamma_P(\alpha) / N
\]

Confidence of rule \(X \rightarrow Y(l_t)\) in DF is defined as follows:

\[
\text{Conf}(X \rightarrow Y(l_t)) = \text{Supp}(X,l_t,Y) / \text{Supp}(X)
\]

Support of rule \(X \rightarrow Y(l_t)\) is degreed

\[
\text{Supp}(X \rightarrow Y(l_t)) = \text{Supp}(X,l_t,Y)
\]

A fuzzy time-interval sequence is called frequently if it has its support more than or equal to \(\text{min\_sup}\) (\(\text{min\_sup}\) is the user’s threshold).

For example: Rule: \(f_{3,1}^{\text{val}} \rightarrow f_{3,1}^{\text{val}}\) (\(\text{Short}\)) \((\text{confidence}=0.568)\) that is mean if \(e_i\) is \(e_i \_2\) and time is \(\text{Short}\) then \(e_i\) is \(e_i \_3\) with its confidence=56.8%.
3. Fuzzy Time-interval in Temporal Quantitative Database Algorithm (FTQ)

3.1 Problem

Given a temporal quantitative database D, and a min_sup, a min_conf value threshold, set LT of linguistic terms, fuzzy membership functions of each of linguistic terms, FE set of fuzzy sets of quantitative attributes in D with their fuzzy membership functions.

The main objective is to determine in database all fuzzy time-interval association rules whose supports are more than or equal to the min_sup and whose confidences are more than or equal to the min_conf.

3.2 FTQ Algorithm

3.2.1. Main Idea

First, we transform the temporal quantitative database D to a temporal fuzzy database DF by using fuzzy sets and their membership functions for quantitative attributes in D. Then, all frequently fuzzy time-interval sequences are found. This process is based on Apriori algorithm: two phases in generating frequently fuzzy time-interval sequences are repeated until all one found. In the first phase, C_1 set of candidates with length of k was built from L_k-1 set of frequently fuzzy time-interval sequences with length of k-1. In the second phase, all frequent sequences in C_k are added into L_k. The process of finding frequently fuzzy time-interval sequences stops when C_k is empty.

Generating C_k is follow:

If k=1: All fuzzy attributes of DF are added into C_1, this is the set of candidates with length of 1.

If k=2: C_2 is result of joining two elements of the L_1 with the LT, denote L_1×LT×L_1. For example, if L_1={p_1, p_2} and LT={Short, Medium, Long}, then C_2 has elements: (p_1, Short, p_2), (p_1, Short, p_2), (p_1, Medium, p_2), (p_1, Long, p_2), (p_1, Short, p_2), (p_1, Medium, p_2), (p_1, Long, p_2), (p_2, Short, p_1), (p_2, Medium, p_1), (p_2, Long, p_1), (p_2, Short, p_1), (p_2, Medium, p_1), (p_2, Long, p_1).

If k>2: If (p_1,l_1,l_2,l_3,...,l_k-1,p_k) and (p_2,l_1,l_2,...,l_k-1,p_k) were two sequences of L_k-1, then α=(p_1,l_1,l_2,l_3,...,l_k-1,l_k,p_2) was generated and added into C_k. In the same way, C_k included all candidates with length of k, is completed.

Then, we computed support of sequences of C_k as follows:

An array of time values is declared. For the first time, all transactions which contain p_1 at time t, are added to the first elements of the array, lst[i][1], (i equivalent i^th sequence contains p_1), each element is a couple of (time, value). Then, transactions, which contain p_2 at t time, are added to the second elements of the array, lst[i][2], if t>lst[i][1].time. The process is continued with transactions t which contain p_m at t time, add to lst[i][m] (3≤m≤k). Result is a list of k elements equivalence a sequences and lst[i][r].time - lst[i][r-1].time (2≤r≤k) is the time-interval of two consecutive elements in the sequence. The support of α is calculated.

Fuzzy time-interval association rules are generated by extracting from frequently fuzzy time-interval sequences with length of more than or equal to two. Then we computed the confidence of the rules by formula (4) and results are all rules which have their confidences more than and equal to min_conf.

3.2.2. 3.2.2 FTQ Algorithm

Input: A temporal quantitative database D, and a min_sup, a min_conf threshold of user, set LT of linguistic terms, fuzzy membership functions of each of linguistic terms, FE set of fuzzy sets of quantitative attributes in D with their fuzzy membership functions.

Output: Fuzzy time-interval association rules have their confidences more than or equal to min_conf and their supports more than or equal to min_sup.

Algorithm as follows:

Transforming D to DF
C_1={attributes of DF}
L_1={α ∈C_1 | Supp(α) ≥ min_sup}
C_2=∅;
for each e_1 ∈ L_1
    for each e_2 ∈ L_1
        for each ltd ∈ LT {
            α = e_1 * ltd * e_2;
            add α to C_2;
        }
for each α ∈ C_2
    α.count = Supp(α);
L_2={α ∈ C_2 | α.count ≥ min_sup}
for (k > 2; L_k-1 ≠ ∅; k++)
{
    C_k=fuzzy_apriori_gen(L_k-1);
}
for each $\alpha \in C_k$
  $a.count=\text{Supp}(\alpha)$;
  $L_k=\{\alpha \in C_k | a.count \geq \minsup\}$
}
return Generating_rules($\cup L_k$);

Supp($\alpha$)//computing support of sequence $\alpha$
{
  $m=0$;
  for each $t_j \in T$
    if ($p_i \in t_j$)
      $m++$;
      $\text{lst}[m][1].time=j$;
      $\text{lst}[m][1].value=p_i(t_j)$; //fuzzy value of $p_i$
in the transaction $t_j$ in the DF
  }
for (i=2; i\leq|\alpha|; i++)
  for each $t_j \in T$
    if ($p_i \in t_j$) and $j \geq \text{lst}[m][i-1].time$
      $\text{lst}[m][i].time=j$;
      $\text{lst}[m][i].value=p_i(t_j)$;
  }
$\text{count}=0$;
for (i=1; i\leq m; i++)
  if (|$\text{lst}[i]$|=|$\alpha$|)
    $s=1$;
    $v=1$;
    for (j=1; j<|$\alpha$|; j++)
      $s=\min(s, \text{lst}[i][j].value)$;
      $v=\min(v, \text{lst}[i][j].value)$;
    $v=\min(v, \text{lst}[i][|\alpha|].value)$;
    $\text{count}=\text{count}+v^s$;
}
return $\text{count}/|DF|$;

fuzzy_apriori_gen($L_{k-1}$)//generating candidates for $C_k$
{
  $C_k=\emptyset$;
  for each $\beta_1 \in L_{k-1}$ // $\beta_1=($p_{1,1}$, \text{plt}_{1,1}$, $p_{2,1}$, $\text{plt}_{2,1}$, ... , $p_{k-2,1}$, $\text{plt}_{k-2,1}$, $p_{k-1,1}$)$
    for each $\beta_2 \in L_{k-1}$ // $\beta_2=($q_{1,1}$, \text{qlt}_{1,1}$, $q_{2,1}$, $\text{qlt}_{2,1}$, ... , $q_{k-2,1}$, $\text{qlt}_{k-2,1}$, $q_{k-1,1}$)$
      $\alpha=\beta_1 \ast \beta_2$
      for (i=2; i\leq k-2; i++)
        if ($p_i \neq q_i$ or $\text{plt}_i \neq \text{qlt}_i$) break;
        $\alpha=\alpha \ast p_i \ast \text{plt}_i$
      if ($i=k-1$ and $a_{k-1}=b_{k-1}$)
        $\alpha=p_i \ast \text{plt}_i \ast \alpha \ast \text{p}_{k-1} \ast \text{qlt}_{k-1} \ast q_k$
        add $\alpha$ to $C_k$;
  }
  return $C_k$;
}
Generating_rules($L$) //Generating rules from set frequently fuzzy time-interval sequences $L$
{
  $R=\emptyset$;
  for each $\alpha \in L$ // $\alpha=($p_{1,1}$, \text{plt}_{1,1}$, $p_{2,1}$, $\text{plt}_{2,1}$, ... , $p_{|p|-1,1}$, $\text{plt}_{|p|-1,1}$, $p_{|p|}$)$
    $r=(p_{1,1}$, $\text{plt}_{1,1}$, $p_{2,1}$, $\text{plt}_{2,1}$, ... , $p_{|p|-1,1}$, $\text{plt}_{|p|-1,1}$, $p_{|p|})$
    if ($\text{Supp}(p)/\text{Supp}(p_{|p|}) \geq \minconf$)
      add $r$ to $R$;
  }
return $R$;
}

where,
The * operator combines elements and linguistic terms values to become a fuzzy time-interval sequence. For example: $p_1 \ast \text{Short} \ast p_2$ become $(p_1, \text{Short}, p_2)$ sequence in case of $\text{Short} \in \text{LT}$ and $p_1, p_2$ elements.

$|p|$ is length of fuzzy time-interval sequence $p$, $p_{|p|}$ is the last item of $p$.

3.2.3. Correctness and completeness of the FTQ algorithm

3.2.3.1. Correctness
By the formula (5), the support of a rule is the support of fuzzy time-interval sequence, which generates the rule; so that all result rules have support not less than min_sup threshold. Confidences of rules are more than or equal to the min_conf cause of formula (4). Thus, the FTQ algorithm is correct.
3.2.3.2. Completeness

Because the FTQ algorithm is developed from Apriori algorithm by building a frequent k-sequence from two (k-1)-sequences, generated frequently fuzzy time-interval sequences are complete.

If it is exist a rule $X \rightarrow y(lt_{i})$ that not in algorithm results but it satisfies both min_sup and min_conf conditions then it is impossible. $\text{Supp}(X \rightarrow y(lt_{i})) = \text{Supp}(X, lt_{i}, y)$ is not less than the min_sup but $(X, lt_{i}, y)$ is not in frequently fuzzy time-interval sequences set. This is not suitable with completeness of frequently fuzzy time-interval sequences of result sets. Therefore, the FTQ algorithm is completeness.

4. Experimental Results

Tested data is Istanbul Stock Exchange that includes seven index stocks of SP, DAX, FTSE, NIKKEI, BOVESPA, MSCE_EU, and MSCI_EM from Jun 5, 2009 to Feb 22, 2011. Tested data is as Table 3 following:

We develop the FTQ algorithm on C# language program. The algorithm is run on computer with Intel Core i5 2.5GHz processor, 4GB RAM, Windows 7.

$LT=$\{Short, Medium, Long\} are linguistic terms of interval-time and fuzzy membership functions for time-interval concept as description in definition 3. Each of quantitative attributes is partition into three parts and their fuzzy membership functions are as formula (1)

Experimental results are as follows:

Table 4 describes relationship among the number of fuzzy time-interval association rules and the min_sup, min_conf

The Figure 1 shows the relationship between the number of result rules and the min_conf threshold with different min_sup.

With different min_conf, number of rules changes equivalent to the min_sup values as Figure 2:

Table 5 and Figure 3 show the time cost of the algorithm in case of minimum confidence is 70%.

Table 3. Tested data

| Database                  | Number of attributes | Number of transactions |
|---------------------------|----------------------|------------------------|
| ISTANBUL STOCK EXCHANGE   | 8                    | 537                    |

g. Tested data

table 4. Relationships among the number of rules and the min_sup, the min_conf

| min_conf | min_sup | 0.65 | 0.70 | 0.75 | 0.80 | 0.85 |
|----------|---------|------|------|------|------|------|
| 0.15     | 1655    | 1481 | 1028 | 501  | 131  |
| 0.17     | 594     | 490  | 291  | 130  | 23   |
| 0.20     | 177     | 137  | 61   | 18   | 1    |
| 0.25     | 32      | 17   | 4    | 0    | 0    |
| 0.30     | 9       | 5    | 0    | 0    | 0    |
| 0.33     | 1       | 1    | 0    | 0    | 0    |

h. Relationship among the number of rules and the min_sup, the min_conf

Figure 1. Relationship between the number of rules and the min_conf with different min_sup.

Figure 2. Relationship between the number of rules and the min_sup with different min_conf.

The results show us the number of the rules and time cost increase quickly when minimum support threshold decrease. It is reasonable cause of number of frequently fuzzy time-interval sequences grow very rapidly.
5. Conclusion

In this paper, FTQ algorithm is proposed to mining fuzzy time-interval association rules from temporal quantitative databases. Rules like a form “If a large number of computers are bought, small number of modem will be bought in a medium period”. The rules help to predict both quantity and occur time of an item from quantity of previous item. The FTQ algorithm is improved from Apriori algorithm to mining frequently fuzzy time-interval sequences. Finally, fuzzy time-interval association rules are determined from the frequently fuzzy time-interval sequences. Experimental results show the relationship among the number of rules and the min_sup, min_conf as well as the time cost of the algorithm.

6. References

1. Agrawal R, Imieliński T, Swami A. Mining association rules between sets of items in large databases. ACM SIGMOD Record. 1993; 22(2):207–16.
2. Qin L-X, Luo P, Shi Z-Z. Efficiently mining frequent item sets with compact FP-tree. Intelligent Information Processing II; Springer: US; 2004. p. 397–406.
3. Zaki MJ, Hsiao C-J. CHARM: An efficient algorithm for closed itemset mining. SDM. 2002; 2:457–73.
4. Qin LX, Shi Z-Z. Efficiently mining association rules from time series. International Journal of Information Technology. 2006; 12(4):30–8.
5. Rakesh A, Srikant R. Fast algorithms for mining association rules. Proceedings. 20th International Conference on Very Large Data Bases, VLDB; 1994. p. 487–99.
6. Saravanan, MS, Sree RJR. A simple process model generation using a new association rule mining algorithm and clustering approach. 2011 Third International Conference on Advanced Computing (ICoAC); 2011. p. 265–69.
7. Sumathi R, Kirubakaran E. Architectural perspective of parallelizing association rule mining. 2012 International Conference on Advances in Engineering, Science and Management (ICAESM); 2012. p. 437–42.
8. Zadeh AL. Fuzzy sets. Information and Control; 1965; 8(3):338–53.
9. Gyenesei A. A fuzzy approach for mining quantitative association rules. Acta Cybern. 2001; 15(2):305–20.
10. Fu A, Wong MH, Sze SC, Wong WL, Yu WK. Finding fuzzy sets for the mining of fuzzy association rules for numerical attributes. The First International Symposium on Intelligent Data Engineering and Learning (IDEAL); 1998. p. 263–68.
11. Chan CCK, Au W-H. Mining fuzzy association rules. Proceedings of the Sixth International Conference on Information and Knowledge Management, ACM; 1997. p. 209–15.
12. Yue JS, Tsang E, Yeung D, Shi D. Mining fuzzy association rules with weighted items. 2000 IEEE International Conference on Systems, Man, and Cybernetics; 2000. p. 1906–11.
13. Au W-H, Chan KCC. FARM: A data mining system for discovering fuzzy association rules. Fuzzy Systems Conference Proceedings, 1999. FUZZ-IEEE’99. 1999 IEEE International; 1999. p. 1217–22.
14. Subramaniam RBV, Goswami A. Mining fuzzy quantitative association rules. Expert Systems. 2006; 23(4):212–25.
15. Weng C-H, Chen Y-L. Mining fuzzy association rules from uncertain data. Knowledge and Information Systems. 2010; 23(2):129–52.
16. Chen C-H, Hong T-P, Tseng VS. Fuzzy data mining for time-series data. Applied Soft Computing. 2012; 12(1):536–42.
17. Chang JH. Mining weighted sequential patterns in a sequence database with a time-interval weight. Knowledge-Based Systems. 2011; 24(1):1–9.
18. Chang JH, Park NH. Finding interesting sequential patterns in sequence data streams via a time-interval weighting
An Effective Algorithm for Association Rules Mining from Temporal Quantitative Databases

approach. IEICE Transactions on Information and Systems 96.8; 2013. p. 1734–44.
19. Chang C-I, Chueh H-E, Lin NP. Sequential patterns mining with fuzzy time-intervals. Sixth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD’09; 2009. p. 165–69.
20. Chang C-I, Chueh H-E, Luo Y-C. An integrated sequential patterns mining with fuzzy time-intervals. International Conference on Systems and Informatics (ICSAI). 2012. p. 2294–98.
21. Moskovitch R, Walsh C, Hripcsak G, Tatonetti NP. Prediction of biomedical events via time intervals mining. ACM KDD Workshop on Connected Health in Big Data Era: USA; 2014.
22. Chen Y-L, Chiang M-C, Ko M-T. Discovering time-interval sequential patterns in sequence databases. Expert Systems with Applications. 2003; 25(3):343–54.
23. Chen Y-L, Huang T C-K. Discovering fuzzy time-interval sequential patterns in sequence databases. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics. 2005; 35(5):959–72.
24. Hu Y-C, Tzeng G-H, Chen C-M. Deriving two-stage learning sequences from knowledge in fuzzy sequential pattern mining. Information Sciences. 2004; 159(1):69–86.
25. UCI-Machine Learning Repository [Internet]. [Cited 2015 Oct 1]. Available from: http://archive.ics.uci.edu/ml/datasets.html.