sciendo

BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **18**, No 2
Sofia • 2018 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2018-0024

Mining Fuzzy Sequential Patterns with Fuzzy Time-Intervals in Quantitative Sequence Databases

Truong Duc Phuong[1], *Do Van Thanh*[2], *Nguyen Duc Dung*[3]

[1]*Hanoi Metropolitan University, Vietnam*
[2]*Faculty of Information Technology, Nguyen Tat Thanh University, Vietnam*
[3]*Institute of Information Technology, Vietnamese Academy of Science and Technology, Vietnam*
E-mails: tdphuong@daihocthudo.edu.vn dvthanh@ntt.edu.vn nddung@ioit.ac.vn

Abstract**:** *The main objective of this paper is to introduce fuzzy sequential patterns with fuzzy time-intervals in quantitative sequence databases. In the fuzzy sequential pattern with fuzzy time-intervals, both quantitative attributes and time distances are represented by linguistic terms. A new algorithm based on the Apriori algorithm is proposed to find the patterns. The mined patterns can be applied to market basket analysis, stock market analysis, and so on.*

Keywords**:** *Data mining, fuzzy sequential pattern, fuzzy time-interval, sequence database.*

1. Introduction

Mining sequential patterns are one of the most important domains in data mining. Mining sequential patterns from transaction databases (events present or not) is the first introduced in 1995 [1] and there are many related works [8-10, 17, 22, 23]. The sequential pattern presented a relationship between events in a chronological sequence with the same object. For instance, "**If** a customer buys a Laptop and later a Modem, **then** he will buy a Printer". Mining fuzzy sequential patterns in sequence databases, in which values of attributes are numeric or categorical, is also introduced in [2, 5, 12, 14, 15, 18]. In these works, the values are transformed into fuzzy sets. A relationship between events in the fuzzy sequential patterns is like "**If** a customer buys a Large number of Laptops and later an Average number of Modems, **then** he will buy a Small number of Printers".

In many cases, the values of time distances among the events in a sequence are preferred. The time-interval sequential patterns in transaction databases were investigated by authors such as Chen and Huang [6], Chen, Chiang and Ko [7], Chang, Chueh and Lin [3], Chang, Chueh and Luo [4]. In [7], time-interval sequential patterns were presented such as ⟨Laptop, I_1, Modem, I_2, Printer⟩, which meant "**If** a customer buys a Laptop and later a Modem an interval of I_1, **then** he will buy a Printer an interval of I_2”, where I_1 and I_2 were predetermined time ranges. For instance, I_1 was ranged from 3 to 5 days, I_2 from 10 to 12 days. To solve sharp boundary problems when a time interval is near the boundary of two time ranges in [7], the fuzzy theory was applied to time intervals [6]. The fuzzy time-interval sequential patterns in [6] showed the relationship among events such as ⟨Laptop, Short, Modem, Long, Printer⟩, which meant “**If** a customer buys a Laptop and later a Modem an interval of Short, **then** he will buy a Printer an interval of Long”, where Short and Long were linguistic terms for time intervals. In [6], time intervals among events were transformed into fuzzy sets and the two algorithms were proposed. They were the FTI-Apriori algorithm based on the idea of an Apriori algorithm and the FTI-PrefixSpan algorithm based on the PrefixSpan algorithm. Papers [3, 4] also revealed fuzzy time-interval sequential patterns in sequence databases such as ⟨Laptop, $\mu_{\text{Laptop\_Modem}}$, Modem, $\mu_{\text{Modem\_Printer}}$, Printer⟩ that meant “**If** a customer buys a Laptop and later a Modem an interval of $\mu_{\text{Laptop\_Modem}}$, **then** he will buy a Printer an interval of $\mu_{\text{Modem\_Printer}}$”, where $\mu_{\text{Laptop\_Modem}}$ and $\mu_{\text{Modem\_Printer}}$ were trapezoidal fuzzy numbers which present time intervals between events of pair (Laptop, Modem) and (Modem, Printer). The fuzzy numbers were computed by frequency of time intervals of events in a sequence. For example, $\mu_{\text{Laptop\_Modem}}$ = (6, 6, 12, 12) and $\mu_{\text{Modem\_Printer}}$ = (2, 2, 7, 15). The main idea of the SPFTI algorithm [3] and the ISPFTI algorithm [4] were taking the advantage of the idea of Apriori algorithm, but the trapezoidal fuzzy numbers were received from databases unlike predetermined fuzzy sets in [6].

In [21], we considered to mine fuzzy association rules with fuzzy time-intervals from quantitative databases. The results showed that the patterns such as ⟨Laptop$_{\text{_Large}}$, Short, Modem$_{\text{_Average}}$, Long, Printer$_{\text{_Small}}$⟩ that meant “**If** a Large number of Laptops are sold and later an Average number of Modems an interval of Short, **then** a Small number of Printers will be sold an interval of Long”. In the pattern, Laptop$_{\text{_Large}}$, Modem$_{\text{_Average}}$, and Printer$_{\text{_Small}}$ were the fuzzy sets of attributes; Short and Long were the pre-defined fuzzy sets for time intervals.

Works [3, 4, 6, 7] have indicated only the fuzzy time-interval sequential patterns in transaction sequence databases, not yet in quantitative sequence databases. While paper [21] has been applied to only quantitative databases, not yet quantitative sequence databases.

So, the main objective of the paper is to introduce an algorithm for mining fuzzy sequential patterns with fuzzy time-intervals in quantitative sequence databases. They are like ⟨⟨Laptop$_{\text{_Large}}$, Short, Modem$_{\text{_Average}}$, Long, Printer$_{\text{_Small}}$⟩⟩ that means “**If** a customer buys a Large number of Laptops and later an Average number of Modems an interval of Short, **then** he will buy a Small number of Printers an interval of Long”, where Short and Long are predetermined linguistic terms for time intervals. These patterns are different from the ones in [3, 4, 6, 7] by quantitative attributes and from the ones in [21] by interesting in the objects. The main idea of the algorithm is to convert quantitative attributes and time intervals to linguistic terms in the same way as in [21] and then improving the Apriori algorithm [1] to find fuzzy sequential patterns with fuzzy time-intervals. The comparison of datasets and patterns of the selected algorithms is described in Table 1.

Table 1. Comparison of datasets and patterns of the selected algorithms

Algorithm	Dataset	Pattern	Description
AprioriAll [1]	Transactional sequence databases	⟨Laptop, Modem, Printer⟩	**If** a customer buys a Laptop and later a Modem, **then** he will buy a Printer
FGBSPMA [5]	Quantitative sequence databases	⟨Laptop$_{_Large}$,Modem$_{_Average}$, Printer$_{_Small}$⟩	**If** a customer buys a Large number of Laptops and later an Average number of Modems, **then** he will buy a Small number of Printers
I-Apriori algorithm, I-PrefixSpan [7]	Transactional sequence databases	⟨Laptop, I_1, Modem, I_2, Printer⟩	**If** a customer buys a Laptop and later a Modem an interval of I_1, **then** he will buy a Printer an interval of I_2
FTI-Apriori, FTI-PrefixSpan [6]	Transactional sequence databases	⟨Laptop, Short, Modem, Long, Printer⟩	**If** a customer buys a Laptop and later a Modem an interval of Short, **then** he will buy a Printer an interval of Long
SPFTI [3], ISPFTI [4]	Transactional sequence databases	⟨Laptop,μ_{Laptop_Modem},Modem, $\mu_{Modem\_Printer}$, Printer⟩	**If** a customer buys a Laptop and later a Modem an interval of μ_{Laptop_Modem} **then** he will buy a Printer an interval of $\mu_{Modem\_Printer}$
FTQ [21]	Quantitative (not sequence) databases	⟨Laptop$_{_Large}$,Short,Modem$_{_Average}$, Long, Printer$_{_Small}$⟩	**If** a Large number of Laptops are sold and later an Average number of Modems an interval of Short, **then** a Small number of Printers will be sold an interval of Long
Proposed	**Quantitative sequence databases**	**⟨⟨Laptop$_{_Large}$, Short, Modem$_{_Average}$, Long, Printer$_{_Small}$⟩⟩**	**If** a customer buys a Large number of Laptops and later an Average number of Modems an interval of Short, **then** he will buy a Small number of Printers an interval of Long

The rest of the paper is organized as follows: Section 2 defines problems. Section 3 develops the FSPFTIM algorithm to find out the fuzzy sequential patterns with fuzzy time-intervals and gives an example. Section 4 presents experimental results. Conclusions are drawn in Section 5.

2. Problem definition

This section presents some definitions related to the proposed problem.

Definition 1. Let E={e_1, e_2, …, e_u} be a set of attributes, s = ⟨(a_1, q_1, t_1), (a_2, q_2, t_2), (a_3, q_3, t_3),…, (a_n, q_n, t_n)⟩ be a quantitative sequence, where $a_k \in E$ is an attribute ($1 \leq k \leq n$), t_k is time of a_k, $t_{k-1} \leq t_k$ with $2 \leq k \leq n$ and $a_k(t_k) = q_k$, q_k is numeric or categorical. A transaction sequence is ⟨Sid, s⟩ where s is a quantitative sequence and Sid is the identifier of the sequence. A quantitative sequence database is a set of all transaction sequences.

Example 1. A quantitative sequence database is shown in Table 2.

In Table 2, E = {a, b, c, d, e, f, g, h, i} is set of attributes, time is started at 0. The first transaction sequence is ⟨1, (a, 2, 1), (b, 2, 4), (e, 5, 29)⟩. It means that this sequence includes Sid with value 1 and three transactions: The a with value 2 (quantitative of *a* is 2) at time 1, the *b* with value 2 at time 4 and the *e* with value 5 at time 29.

Table 2. An example of a quantitative sequence database

Sid	Quantitative sequence
1	⟨(*a*, 2, 1), (*b*, 2, 4), (*e*, 5, 29)⟩
2	⟨(*d*, 2, 1), (*a*, 5, 2), (*d*, 4, 24)⟩
3	⟨(*b*, 1, 1), (*d*, 2, 11), (*e*, 5, 28)⟩
4	⟨(*f*, 6, 1), (*b*, 6, 5), (*c*, 1, 19), (*c*, 2, 25)⟩
5	⟨(*a*, 1, 4), (*b*, 1, 5), (*d*, 2, 10), (*e*, 5, 28)⟩
6	⟨(*a*, 2, 0), (*b*, 1, 5), (*e*, 1, 30)⟩
7	⟨(*i*, 5, 2), (*a*, 3, 17), (*h*, 2, 17)⟩
8	⟨(*c*, 6, 3), (*i*, 5,10), (*f*, 3, 18)⟩
9	⟨(*h*, 3, 4), (*a*, 1, 10), (*b*, 6, 21)⟩
10	⟨(*a*, 2, 0), (*g*, 5, 0), (*b*, 2, 3), (*e*, 1, 30)⟩

Definition 2. Let FE = $\left\{F^{e_1}, F^{e_2}, \dots, F^{e_u}\right\}$ be a set of fuzzy sets of attributes of *E*, $F^{e_k} = \left\{ f^{e_k}_{h_k,1}, f^{e_k}_{h_k,2}, \dots, f^{e_k}_{h_k,h_k} \right\}$ be a set of fuzzy sets of e_k attribute (k=1, 2, ..., u), where $f^{e_k}_{h_k,j}$ is the *j*-th fuzzy set ($1\le j \le h_k$), h_k is the number of fuzzy sets of e_k; $f^{e_k}_{h_k,j}$ is called a fuzzy attribute. Each fuzzy set has its membership function μ: $X\to[0, 1]$. Sequence fs=⟨(fa_1, fq_1, t_1), (fa_2, fq_2, t_2),…, (fa_n, fq_n, t_n)⟩ is called a fuzzy one, where $\mathrm{fa}_i\in F^{a_i}$ ($1\le i\le n$) is a fuzzy set, fq_i is the value of the membership function μ_{fa_i} of fa_i at q_i ($\mathrm{fq}_i = \mu_{\mathrm{fa}_i}(q_i)$), fa_i is called a fuzzy attribute. ⟨Sid, fs⟩ is called a *fuzzy transaction sequence*. A fuzzy sequence database is a set of all fuzzy transaction sequences.

Example 2. Given fuzzy sets are in [13], $A^{x_m}_{K,i_m}$ is i_m-th linguistic value ($1\le i_m\le K$) of the x_m attribute ($x_m\in E$), *K* is the number of partitions of x_m, $\mu^{x_m}_{K,i_m}$ is the membership function of $A^{x_m}_{K,i_m}$ that is determined as follows:

(1) $$\mu^{x_m}_{K,i_m}(v) = \max\left\{1 - \left|v - a^K_{i_m}\right| / b^K, 0\right\},$$

where

$$a^K_{i_m} = \mathrm{mi} + (\mathrm{ma} - \mathrm{mi})(i_m - 1) / (K - 1),$$

$$b^k = (\mathrm{ma} - \mathrm{mi})/(K - 1),$$

where mi is the minimum value of x_m attribute domain, and ma is the maximum value. With K=3 for all attributes in the database of the Example 1, we get membership functions as in the Fig. 1 and the fuzzy transaction sequence database D' is described in Table 3.

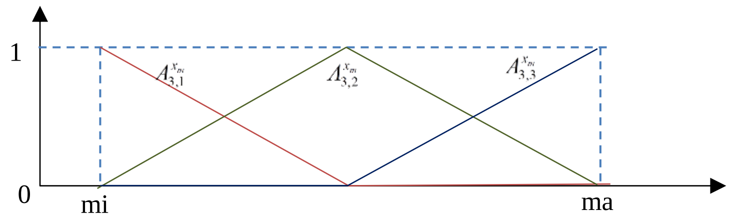

Fig. 1. The membership functions of x_m with K=3

Table 3. The fuzzy transaction sequence database D'

Sid	Fuzzy sequence
1	$\langle (f_{3,1}^{a}, 0.5, 1), (f_{3,2}^{a}, 0.5, 1), (f_{3,1}^{b}, 0.6, 4), (f_{3,2}^{b}, 0.4, 4), (f_{3,3}^{e}, 1, 29) \rangle$
2	$\langle (f_{3,1}^{d}, 1, 1), (f_{3,3}^{a}, 1, 2), (f_{3,1}^{d}, 1, 24) \rangle$
3	$\langle (f_{3,1}^{b}, 1, 1), (f_{3,1}^{d}, 1, 11), (f_{3,3}^{e}, 1, 28) \rangle$
4	$\langle (f_{3,3}^{f}, 1, 1), (f_{3,3}^{b}, 1, 5), (f_{3,1}^{c}, 1, 19), (f_{3,1}^{c}, 0.6, 25), (f_{3,2}^{c}, 0.4, 25) \rangle$
5	$\langle (f_{3,1}^{a}, 1, 4), (f_{3,1}^{b}, 1, 5), (f_{3,1}^{d}, 1, 10), (f_{3,3}^{e}, 1, 28) \rangle$
6	$\langle (f_{3,1}^{a}, 0.5, 0), (f_{3,2}^{a}, 0.5, 0), (f_{3,1}^{b}, 1, 5), (f_{3,1}^{e}, 1, 30) \rangle$
7	$\langle (f_{3,1}^{i}, 1, 2), (f_{3,2}^{a}, 1, 17), (f_{3,1}^{h}, 1, 17) \rangle$
8	$\langle (f_{3,3}^{c}, 1, 3), (f_{3,1}^{i}, 1, 10), (f_{3,1}^{f}, 1, 18) \rangle$
9	$\langle (f_{3,3}^{h}, 1, 4), (f_{3,1}^{a}, 1, 10), (f_{3,3}^{b}, 1, 21) \rangle$
10	$\langle (f_{3,1}^{a}, 0.5, 0), (f_{3,2}^{a}, 0.5, 0), (f_{3,1}^{g}, 1, 0), (f_{3,1}^{b}, 0.6, 3), (f_{3,2}^{b}, 0.4, 3), (f_{3,1}^{e}, 1, 30) \rangle$

Each fuzzy tuple $\left(f_{K,i_m}^{x}, \text{fq}, t \right)$ in the D' above means that the fuzzy attribute f_{K,i_m}^{x} is the i_m-th fuzzy set of x attribute, fq$\in$[0, 1] is the value of the membership function of f_{K,i_m}^{x} at q and t is the time of the event. For example, with $(f_{3,2}^{a}, 0.5, 1)$, the value of membership function of the fuzzy set $f_{3,2}^{a}$ at a=2 is 0.5, the transaction time is 1.

Definition 3. Let LT={lt_j| j=1, 2,..., p} be fuzzy sets of time-intervals, $\mu_{\text{lt}_j}: X \to [0,1]$ be the membership function of the fuzzy set lt_j [13]. Then, $\alpha = \langle\langle b_1, \text{lt}b_1, b_2, \text{lt}b_2, \ldots, b_{r-1}, \text{lt}b_{r-1}, b_r \rangle\rangle$ is called a *fuzzy sequence with fuzzy time-intervals* if b_i, $1 \le i \le r$ is a fuzzy set and $\text{lt}b_i \in$LT, $1 \le i \le r-1$. A fuzzy sequence with fuzzy time-intervals whose length is r referred to r-fuzzy sequential pattern with fuzzy time-intervals.

Example 3. Let LT={Short, Medium, Long} be fuzzy sets of time intervals. Its membership functions as in the Fig. 2 [6] and the fuzzy sequence database D' in Table 3, then $\alpha = \langle\langle f_{3,1}^{a}$, Short, $f_{3,1}^{b}$, Medium, $f_{3,1}^{e} \rangle\rangle$ is a fuzzy sequence with fuzzy time-intervals whose length is 3 or 3-fuzzy sequence with fuzzy time-intervals.

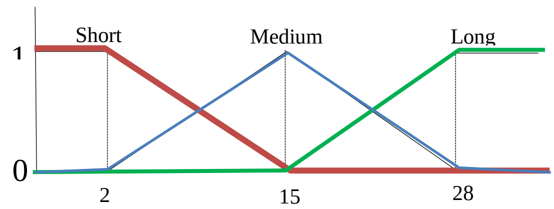

Fig. 2. The membership functions of the fuzzy sets in LT

Definition 4. Given a fuzzy sequence B = ⟨(b_1, bq_1, bt_1), (b_2, bq_2, bt_2), …, (b_r, bq_r, bt_r)⟩ and a fuzzy sequence with fuzzy time-intervals α = ⟨⟨b_1, ltb_1, b_2, ltb_2,…, b_{r-1}, ltb_{r-1}, b_r⟩⟩, we define:

- The support of B for α, denoted by $\gamma_B(\alpha)$, is

(2) $$\gamma_B(\alpha)=\begin{cases}\mathrm{bq}_1 & \text{if } r=1,\\ \left(\prod_{i=1}^{r}\mathrm{bq}_i\right)\times\min_{1\le j\le r-1}\left\{\mu_{\mathrm{ltb}_j}(\mathrm{bt}_{j+1}-\mathrm{bt}_j)\right\} & \text{if } r>1.\end{cases}$$

- A fuzzy sequence B belongs to a fuzzy transaction sequence S =⟨Sid, fs⟩ where fs = ⟨(fa_1, fq_1, t_1), (fa_2, fq_2, t_2),.., (fa_n, fq_n, t_n)⟩ if there exists a integer r such that $b_k = fa_{i_k} \wedge bq_k = fq_{i_k} \wedge bt_k = t_{i_k}$, $1\le k\le r$, and $i_1 < i_2 <\ldots< i_r$. The support of the fuzzy transaction sequence S for α, denoted by $\mathrm{Supp}_S(\alpha)$, is the maximum of the supports of B which belong to S,

(3) $$\mathrm{Supp}_S(\alpha) = \max_{B\in S}\left(\gamma_B(\alpha)\right).$$

- The support of a fuzzy sequence with fuzzy time-intervals α, denoted Supp(α), is the average of supports of all fuzzy transaction sequences in D' for α.

(4) $$\mathrm{Supp}(\alpha)=\frac{\sum_{i=1}^{\mathrm{NS}}\mathrm{Supp}_{S_i}(\alpha)}{\mathrm{NS}},$$

where S_i is the i-th fuzzy transaction sequence in D', NS is the number of fuzzy transaction sequences in D'.

- A fuzzy sequential pattern with fuzzy time-intervals is the fuzzy sequence with fuzzy time-intervals whose support is not less than a user-defined threshold.

Example 4. Given the fuzzy transaction sequence database D', the membership functions of fuzzy sets of LT mentioned in the Example 3, we compute the support of two fuzzy sequences with fuzzy time-intervals ⟨⟨ $f^a_{3,1}$, Short, $f^b_{3,1}$⟩⟩ and ⟨⟨ $f^f_{3,3}$, Short, $f^b_{3,2}$, Medium, $f^c_{3,1}$⟩⟩.

Case 1. α=⟨⟨ $f^a_{3,1}$, Short, $f^b_{3,1}$⟩⟩.

$\mathrm{Supp}_{S_1}(\alpha)$ = 0.5×0.667×μ_{Short}(4 – 1)= 0.5×0.667×0.923=0.308, because the first fuzzy transaction sequence has only the fuzzy sequence ⟨($f^a_{3,1}$, 0.5, 1), ($f^b_{3,1}$, 0.667, 4)⟩ fitting α. Similarly, the supports of the 5th fuzzy transaction sequence and the 6th fuzzy transaction sequence are $\mathrm{Supp}_{S_5}(\alpha)$ = 1×1×μ_{Short}(5 – 4) = 1×1×1=1 and $\mathrm{Supp}_{S_6}(\alpha)$ = 0.5×1×μ_{Short}(5 – 0)= 0.5×0.769=0.385, respectively. Not all the rest fuzzy transaction sequences support for α. So that, the support of the fuzzy sequence with fuzzy time-intervals α is computed as follows:

$$\text{Supp}(\alpha)=(0.308+0+0+0+1+0.385+0+0+0+0)/10 = 0.1693.$$

Case 2. $\beta = \langle\langle f^f_{3,3}$, Short, $f^b_{3,2}$, Medium, $f^c_{3,1}\rangle\rangle$.

In D', there is only the 4th fuzzy transaction sequence, S_4, containing fuzzy sequences fitting β,

$S_4 = \langle 4, (f^f_{3,3}, 1, 1), (f^b_{3,2}, 0.333, 5), (f^b_{3,3}, 0.667, 5), (f^c_{3,1}, 1, 19), (f^c_{3,1}, 0.6, 25), (f^c_{3,2}, 0.4, 25)\rangle$ has two fuzzy sequences, fitting β that are B_1, B_2:

$B_1 = \langle(f^f_{3,3}, 1, 1), (f^b_{3,2}, 0.333, 5), (f^c_{3,3}, 1, 19)\rangle$ has the support for β,

$\gamma_{B_1}(\beta) = 1\times0.333\times1\times\min\{\mu_{\text{Short}}(4), \mu_{\text{Medium}}(14)\} = 0.333\times\min\{0.846, 0.923\} = 0.333\times0.846 = 0.285$,

$B_2 = \langle(f^f_{3,3}, 1, 1), (f^b_{3,2}, 0.333, 5), (f^c_{3,1}, 0.6, 25)\rangle$ has the support for β,

$\gamma_{B_2}(\beta) = 1\times0.333\times0.6\times\min\{\mu_{\text{Short}}(4), \mu_{\text{Medium}}(20)\} = 0.2\times\min\{0.846, 0.615\} = 0.2\times0.615 = 0.123$.

Hence, the support of S_4 for β is $\text{Supp}_{S_4}(\beta) = \max\{\gamma_{B_1}(\beta), \gamma_{B_2}(\beta)\} = 0.285$.

Consequently, $\text{Supp}(\beta)=(0+0+0+0.285+0+0+0+0+0+0)/10 = 0.0285$.

If the user-defined threshold is of value 0.1 then the fuzzy sequence with fuzzy time-intervals $\langle\langle f^a_{3,1}$, Short, $f^b_{3,1}\rangle\rangle$ is a fuzzy sequential pattern with fuzzy time-intervals because its support 0.1693 is greater than min_sup, whereas the fuzzy sequence with fuzzy time-intervals $\langle\langle f^f_{3,3}$, Short, $f^b_{3,2}$, Medium, $f^c_{3,1}\rangle\rangle$, having the support of $0.0285 < 0.1$, is not a sequential pattern.

3. Algorithm of Mining Fuzzy Sequential Patterns with Fuzzy Time-Intervals – FSPFTIM Algorithm

3.1. The problem

Input: D is a quantitative sequence database, min_sup is a user-defined threshold, FE is a set of fuzzy sets with its membership functions of attributes in D, LT is a set of fuzzy sets with its membership functions of time intervals.

Output: k-fuzzy sequential patterns with fuzzy time intervals, $k > 1$

3.2. Proposed approach

First, all the quantitative attributes in D are partitioned into fuzzy sets, then the transaction sequences in D are transformed into fuzzy transaction sequences based on these fuzzy sets and their corresponding membership functions. And we get a fuzzy sequence database, called D'. Next, the FSPFTIM Algorithm is applied to find out fuzzy sequential patterns with fuzzy time-intervals. This algorithm is improved from the Apriori algorithm.

The algorithm is an iterative process consisting of several steps. At the k-th step, the algorithm generates a set denoted C_k of candidate k-fuzzy sequences with fuzzy time intervals and then calculates the support of these candidate sequences. The sequences having the support greater than min_sup will be added into L_k as a set of k-fuzzy sequential patterns with fuzzy time-intervals. This process will be ended when it is unable to generate any new set of fuzzy sequential patterns with fuzzy time-intervals.

More specifically, the process of generating the sets C_k is as follows:

For *k*=1. All fuzzy attributes of D' are added into the set C_1 of candidate 1-fuzzy sequences with fuzzy time-intervals. By calculating the support of sequences in C_1, the set L_1 of 1-fuzzy sequential patterns with fuzzy time-intervals is created.

For *k*=2. The set C_2 of 2-fuzzy sequences with fuzzy time-intervals are generated by joining two sequences in L_1 together and with LT in the form of $L_1 \times \mathrm{LT} \times L_1$. For example, if $L_1=\{f_a, f_b\}$, LT=$\{\mathrm{lt}_1, \mathrm{lt}_2, \mathrm{lt}_3\}$, then there are nine candidate 2-fuzzy sequences with fuzzy time-intervals such as $\langle\langle f_a, \mathrm{lt}_1, f_a\rangle\rangle$, $\langle\langle f_a, \mathrm{lt}_2, f_a\rangle\rangle$, $\langle\langle f_a, \mathrm{lt}_3, f_a\rangle\rangle$, $\langle\langle f_a, \mathrm{lt}_1, f_b\rangle\rangle$, $\langle\langle f_a, \mathrm{lt}_2, f_b\rangle\rangle$, $\langle\langle f_a, \mathrm{lt}_3, f_b\rangle\rangle$, $\langle\langle f_b, \mathrm{lt}_1, f_b\rangle\rangle$, $\langle\langle f_b, \mathrm{lt}_2, f_b\rangle\rangle$, $\langle\langle f_b, \mathrm{lt}_3, f_b\rangle\rangle$.

For *k*>2. Let $\langle\langle b_1, \mathrm{lt}_1, b_2, \mathrm{lt}_2,\ldots, \mathrm{lt}_{k-2}, b_{k-1}\rangle\rangle$ and $\langle\langle b_2, \mathrm{lt}_2, b_3, \mathrm{lt}_3,\ldots, \mathrm{lt}_{k-1}, b_k\rangle\rangle$ be 2 $(k-1)$-fuzzy sequential patterns with fuzzy time-intervals in L_{k-1}, then $\alpha = \langle\langle b_1, \mathrm{lt}_1, b_2, \mathrm{lt}_2, b_3, \mathrm{lt}_3,\ldots,b_{k-1}, \mathrm{lt}_{k-1}, b_k\rangle\rangle$ is a candidate k-fuzzy sequence with fuzzy time-intervals [6]. In the same way, all candidate k-fuzzy sequences with fuzzy time-intervals are generated and set C_k is created. The support of each candidate k-fuzzy sequence with fuzzy time-intervals in C_k is calculated by the formula (4) and the set L_k is created.

The result of the algorithm is a set of all k-fuzzy sequential patterns with fuzzy time-intervals L_k for k>1.

3.3. FSPFTIM Algorithm

Pseudo code of the algorithm is shown in Fig. 3.

In the algorithm, the operator "*" joins the values into a fuzzy sequence with fuzzy time-intervals. For example, fe_1*Short*fe_2 is a presentation of the fuzzy sequence with fuzzy time-intervals $\langle\langle \mathrm{fe}_1, \mathrm{Short}, \mathrm{fe}_2\rangle\rangle$ where Short $\in$LT and fe_1, fe_2 are fuzzy sets of quantitative attributes.

At line 2, each fuzzy tuple, considered as a 1-fuzzy sequence with fuzzy time-intervals, is added into C_1. If the support of a sequence is greater than or equal to min_sup then the sequence will be added into L_1 at line 3. The set of candidates 2-fuzzy sequences with fuzzy time-intervals, $C_2=L_1\times\mathrm{LT}\times L_1$, is generated by the lines 4-12. The support of candidate 2-fuzzy sequences with fuzzy time-intervals in C_2 are calculated (lines 13-15) and 2-fuzzy sequences with fuzzy time-intervals having the support greater than or equal to min_sup, are added into L_2 (line 16). In the lines 17-24, candidate k-fuzzy sequences with fuzzy time-intervals are added into C_k (with k >2) and by calculating their support, k-fuzzy patterns with fuzzy time-intervals are included in L_k. The loop for generating candidate sequences is ended when L_k is empty. The result of the algorithm is the set of all k-fuzzy sequential patterns with fuzzy time-intervals (line 24) with k>1.

```
Procedure FSPMFTI (D, min_sup)
1.   Creating a fuzzy sequence database D′ from the sequence database D
2.   C1←{fe| fe is an attribute of D′}
3.   L1←{α∈C1|Supp(α)≥min_sup}
4.   C2←∅;
5.   for each fe1∈L1
6.      for each fe2∈L1
7.            for each ltd∈LT
8.                 α←fe1*ltd*fe2;
9.                 add α to C2;
10.           end for
11.        end for
12.  end for
13.  for each α∈C2
14.     Computing the support of α (Supp(α));
15.  end for
16.  L2←{α∈C2|Supp(α) ≥min_sup}
17.  for (k>2;Lk–1≠∅;k++)
18.        Ck←fuzzy_apriori_gen(Lk–1);
19.        for each c∈Ck
20.           Computing the Supp(α)
21.        end for
22.        Lk←{α∈Ck|Supp(α) ≥min_sup}
23.  end for
24.  return ∪Lk //k>1
Procedure fuzzy_apriori_gen(Lk–1) //generating the candidates of Ck
25.  Ck←∅;
26.  for each a∈Lk–1
27.        for each b∈Lk–1
28.           α←∅;
29.           for (i=2; i<=k–2; i++)
30.                if (ai≠bi–1 or alti≠blti–1)
31.                        break;
32.                end if
33.                α←c*ai*alti;
34.           end for
35.           if (i=k–1 and ak–1=bk–2)
36.                α←a1*alt1 * c*ak–1*bltk–2*bk–1;
37.                add α to Ck;
38.           end if
39.        end for
40.  end for
41.  return Ck
```

Fig. 3. The FSPFTIM Algorithm

The function generating candidate k-fuzzy sequences with fuzzy time-intervals in C_k from $L_{k\text{-}1}$ is presented in the lines 25-41. The check of the conditions for combining two (k – 1)-fuzzy sequential patterns with fuzzy time-intervals in $L_{k–1}$ to creates a candidate k-fuzzy sequence with fuzzy time-intervals in lines 29-35. A candidate k-fuzzy sequence with fuzzy time-intervals α is created and added into C_k in lines 36-37. At line 41, the set of candidate k-fuzzy sequences with fuzzy time-intervals, C_k, is created.

The complexity of the FSPFTIM Algorithm

The parameters used to evaluate the computational complexity of the proposed algorithm are as follows:

N is the number of transaction sequences in *D*;
M is the total number of attributes in *D*;
l is the average length of transaction sequences in *D*;
h is the number average of of fuzzy sets associated with each attribute in *D*;
|LT| is the number of fuzzy sets of time-intervals LT;
min_supp is a user-defined threshold.

Based on the similar way as in [20], the computational complexity of the FSPFTIM algorithm is

$$O(N.l.h + M.N.l.h^2 + |L_1|^2.\binom{2}{l.h}.N.|LT|) + \sum_{k=3}^{l.h}|L_{k-1}|^2.\binom{k}{l.h}.N\ |LT|)\ \blacksquare$$

Details are given in the Appendix.

3.4. An example of executing the FSPFTIM Algorithm

Input:

- min_sup = 0.11;
- Quantitative sequence database *D* in Table 2;
- Fuzzy sets of attributes and their membership functions in the Example 2;
- Set of fuzzy sets of time-intervals LT={Short, Medium, Long} and their membership functions in the Example 3.

Output: *k*-fuzzy sequential patterns with fuzzy time-intervals, *k* >1.

Execution:

- *D′* is created as the Table 3;
- $C_1=\left\{\begin{matrix} f^a_{3,1}, f^a_{3,2}, f^a_{3,3}, f^b_{3,1}, f^b_{3,2}, f^b_{3,3}, f^c_{3,1}, f^c_{3,2}, f^c_{3,3}, \\ f^d_{3,1}, f^e_{3,1}, f^e_{3,3}, f^f_{3,1}, f^f_{3,3}, f^g_{3,1}, f^h_{3,1}, f^h_{3,3}, f^i_{3,1} \end{matrix}\right\}$;
- $L_1=\left\{f^a_{3,1}, f^a_{3,2}, f^b_{3,1}, f^b_{3,3}, f^d_{3,1}, f^e_{3,1}, f^e_{3,3}, f^i_{3,1}\right\}$;
- C_2 (included 147 elements) =

$$=\left\{\begin{matrix} \langle\langle f^a_{3,1}, \text{Short}, f^a_{3,1}\rangle\rangle, \langle\langle f^a_{3,1}, \text{Medium}, f^a_{3,1}\rangle\rangle, \langle\langle f^a_{3,1}, \text{Long}, f^a_{3,1}\rangle\rangle, \\ \langle\langle f^a_{3,1}, \text{Short}, f^a_{3,2}\rangle\rangle, \langle\langle f^a_{3,1}, \text{Medium}, f^a_{3,2}\rangle\rangle, \langle\langle f^a_{3,1}, \text{Long}, f^a_{3,2}\rangle\rangle, \\ ..., \\ \langle\langle f^i_{3,1}, \text{Short}, f^i_{3,1}\rangle\rangle, \langle\langle f^i_{3,1}, \text{Medium}, f^i_{3,1}\rangle\rangle, \langle\langle f^i_{3,1}, \text{Long}, f^i_{3,1}\rangle\rangle \end{matrix}\right\};$$

- $L_2=\left\{\begin{array}{l}\langle\langle f_{3,1}^{a}, \text{Short}, f_{3,1}^{b}\rangle\rangle, \langle\langle f_{3,1}^{a}, \text{Long}, f_{3,3}^{e}\rangle\rangle, \langle\langle f_{3,1}^{b}, \text{Long}, f_{3,3}^{e}\rangle\rangle, \\ \langle\langle f_{3,1}^{b}, \text{Short}, f_{3,1}^{d}\rangle\rangle, \langle\langle f_{3,1}^{b}, \text{Long}, f_{3,1}^{e}\rangle\rangle, \langle\langle f_{3,1}^{d}, \text{Medium}, f_{3,3}^{e}\rangle\rangle\end{array}\right\}.$
- $C_3=\left\{\begin{array}{l}\langle\langle f_{3,1}^{a}, \text{Short}, f_{3,1}^{b}, \text{Long}, f_{3,1}^{e}\rangle\rangle, \langle\langle f_{3,1}^{a}, \text{Short}, f_{3,1}^{b}, \text{Short}, f_{3,1}^{d}\rangle\rangle, \\ \langle\langle f_{3,1}^{a}, \text{Short}, f_{3,1}^{b}, \text{Long}, f_{3,3}^{e}\rangle\rangle, \langle\langle f_{3,1}^{a}, \text{Short}, f_{3,1}^{d}, \text{Medium}, f_{3,3}^{e}\rangle\rangle\end{array}\right\};$
- $L_3=\left\{\langle\langle f_{3,1}^{a}, \text{Short}, f_{3,1}^{d}, \text{Medium}, f_{3,3}^{e}\rangle\rangle\right\};$
- $C_4=\varnothing;$
- $L_4=\varnothing.$
- **Output**= $\left\{\begin{array}{l}\langle\langle f_{3,1}^{a}, \text{Short}, f_{3,1}^{b}\rangle\rangle, \langle\langle f_{3,1}^{a}, \text{Long}, f_{3,3}^{e}\rangle\rangle, \\ \langle\langle f_{3,1}^{b}, \text{Long}, f_{3,3}^{e}\rangle\rangle, \langle\langle f_{3,1}^{b}, \text{Short}, f_{3,1}^{d}\rangle\rangle, \\ \langle\langle f_{3,1}^{b}, \text{Long}, f_{3,1}^{e}\rangle\rangle, \langle\langle f_{3,1}^{d}, \text{Medium}, f_{3,3}^{e}\rangle\rangle, \\ \langle\langle f_{3,1}^{a}, \text{Short}, f_{3,1}^{d}, \text{Medium}, f_{3,3}^{e}\rangle\rangle\end{array}\right\}.$

4. Experimental results

The algorithm is implemented in the C# programming language and run on Chip Intel Core i5 2.5 GHz, RAM 4 GB, Windows 7 OS.

4.1. Datasets

Two datasets used for experiment of the proposed algorithm are S100I1000T3D341K and Online Retail_France. These datasets are shown in Table 4. The dataset S100I1000T3D341K is generated according to the theory of *R*. Agrawal and *R*. Srikant [19]. Input parameters for generating data include the number of transactions *D*, the average of transactions *T*, the number of attributes *I*, and the number of transaction sequences *S*. The length of the transactions is generated by the Poisson distribution in which the parameter is the average of transactions (*T*). The quantitative values and the time are random integers from 1 up to 1000. The Online Retail_France dataset is gathered from Online Retail [24] with the Country="France" from 01/12/2010 to 09/12/2011. The dataset includes information as follows:

- Customer ID;
- Invoice Date;
- Stock Code;
- Quantity.

Table 4. Datasets

Dataset	Number of attributes (I)	Number of transactions (D)	Number of transaction sequences (S)	Average of transactions (T)	Average of transaction sequences
S100I1000T3D341K	1000	341	100	29.3	3.41
Online Retail_France	1523	365	87	22.8	4.20

The test dataset is transformed into a quantitative sequence database fitted for the input of the algorithm as follows:

- CusId: Customer ID;
- Time: is an integer number starting at 1, which is the number of days of Invoice Date differ from the first day (01/12/2010);
- StockCode: Descriptions of stock codes bought;
- Quantity: Quantitative value of each StockCode.

Each StockCode is partitioned into fuzzy sets and their membership functions are defined according to the Formulae (1) in Example 2 with K=3 and $A_{3,1}^{x_m} = x_{m\_\text{Small}}$, $A_{3,2}^{x_m} = x_{m\_\text{Average}}$, $A_{3,3}^{x_m} = x_{m\_\text{Large}}$. The ma, mi are the maximum, minimum of the quantitative values of x_m StockCode. Quantitative values of the StockCode are used to calculate fuzzy values.

Fuzzy sets of the time-intervals and their membership functions are defined as in formula (1) in the Example 2, too. The ma, mi are the first time and the last time in order and $A_{3,1}^{t} = \text{Short}$, $A_{3,2}^{t} = \text{Medium}$, $A_{3,3}^{t} = \text{Long}$ where t is time-interval. The time distances are used to calculate fuzzy time-intervals.

4.2. Results

4.2.1. Relationships between the number of fuzzy sequential patterns with fuzzy time-intervals and min_sup, and between the execution time and min_sup according to the different numbers of partitions of quantitative attributes.

In this case, the number of fuzzy sets of time intervals is fixed. Namely, the time intervals are partitioned into 3 fuzzy sets (K_t=3). The relationships between the number of fuzzy sequential patterns with fuzzy time-intervals and min_sup, and between the execution time and min_sup according to different numbers of partitions of quantitative attributes for the S100I1000T3D341K dataset are shown in the Figs 4a and b, respectively, and for the Online Retail_France datasets are also shown in the Figs 5a and b.

From these Figs, we can see that when the number of partitions of time intervals is fixed, the number of fuzzy sequential patterns with fuzzy time-intervals as well as the execution time will be changed in inverse direction with changes of min_sup. This is true for both the S100I1000T3D341K and Online Retail_France datasets, and does not depend on the number of partitions of quantitative attributes (K). Also, as K increases, the number of fuzzy sequential patterns with fuzzy time-intervals decreases, but with min_sup greater than a certain threshold, when K increases then the execution time increases. In addition, the difference between the numbers of sequence patterns, and between the times of execution (depending on the number of partitions of the quantitative attributes) will be narrowed very rapidly according to the increase of min_sup.

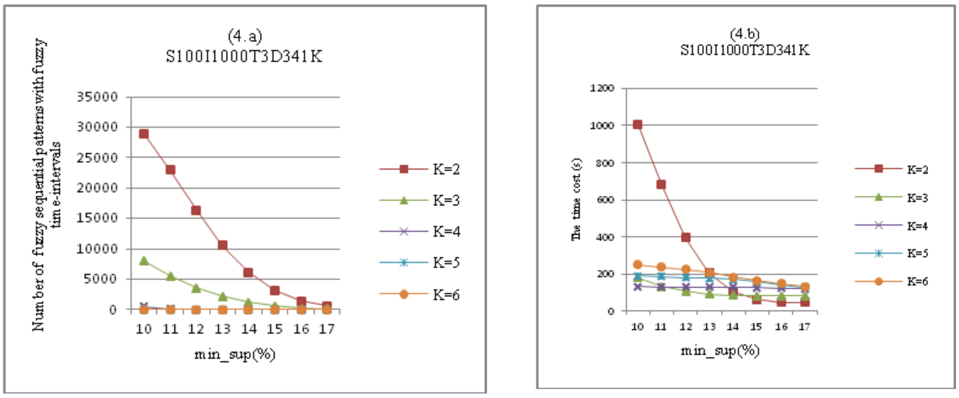

Fig. 4. Relationship between the number of fuzzy sequential patterns with fuzzy time-intervals and min_sup (a), Relationship between the execution time and min_sup (b) according to the different numbers of partitions of quantitative attributes (*K*) for the S100I1000T3D341K dataset

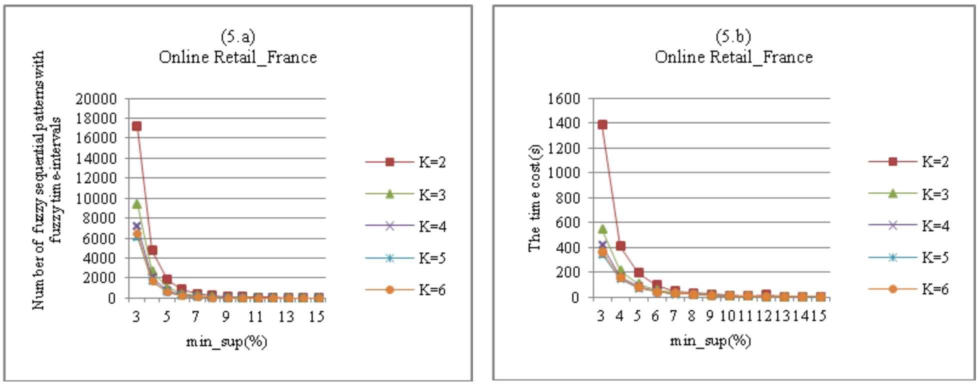

Fig. 5. Relationship between the number of fuzzy sequential patterns with fuzzy time-intervals and min_sup (a), Relationship between the execution time and min_sup (b) according to different numbers of partitions of quantitative attributes (*K*) for the Online Retail_France dataset

4.2.2. Relationships between the number of fuzzy sequential patterns with fuzzy time-intervals and min_sup, and between the execution time and min_sup according to the different numbers of partitions of time-intervals

In this case, the number of partitions of quantitative attributes is 3 (*K*=3). Figs 6a and 7a, show the relationship between the number of fuzzy sequential patterns with fuzzy time-intervals and min_sup according to the different numbers of partitions of the time-intervals (K_t) for the S100I1000T3D341K and Online Retail_France datasets, respectively. Similarly, Figs. 6b and 7b show the relationship between the time of execution and min_sup for the two test datasets mentioned above.

The relationship between the number of fuzzy sequential patterns with fuzzy time-intervals and min_sup, and between the time of execution and min_sup according to changes of the number of partitions of time-intervals in the context the number of partitions of the quantitative attributes is fixed they are very similar to the relationship mentioned in Section 4.2.1. For example, Figs 6b and 7b also show that when the number of partitions of quantitative attributes is fixed, if the number of partitions of time-intervals increases then the execution time also increases.

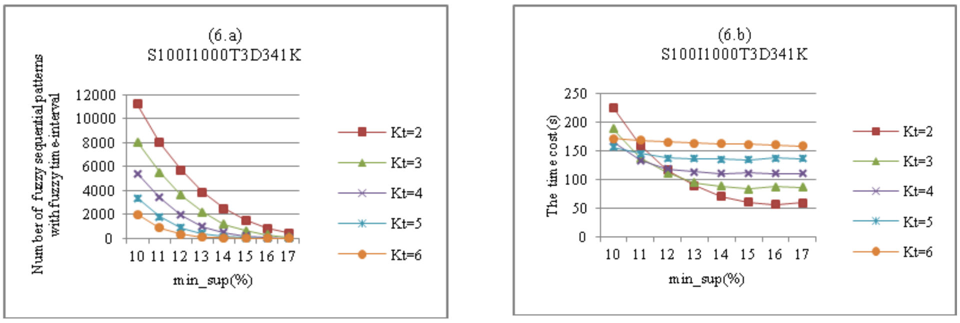

Fig. 6. Relationship between the number of fuzzy sequential patterns with fuzzy time-intervals and min_sup (a), between the execution time and min_sup (b) according to the different numbers of partitions of time-intervals (K_t) for the S100I1000T3D341K dataset

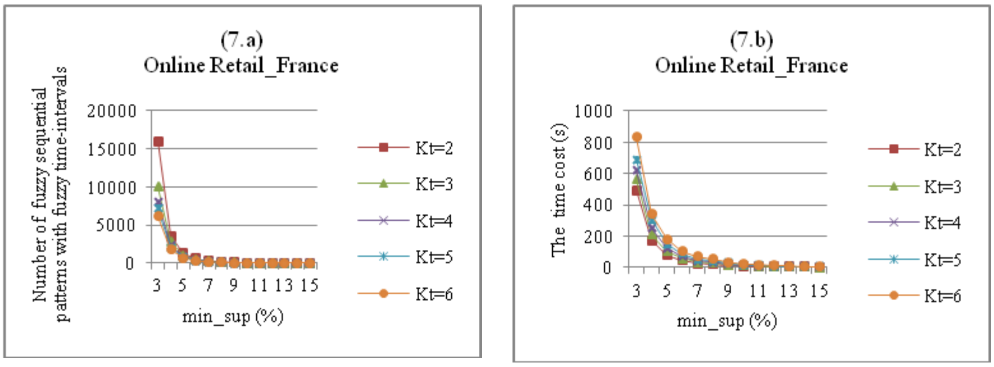

Fig. 7. Relationship between the number of fuzzy sequential patterns with fuzzy time-intervals and min_sup (a), between the execution time and min_sup (b) according to the different numbers of partitions of time-intervals (K_t) for the Online Retail_France dataset

4.3. Meaning of fuzzy sequential patterns with fuzzy time-intervals

Table 5 shows all fuzzy sequence patterns with fuzzy time-intervals that were found out using the FSPFTIM algorithm in the Online Retail_France dataset when the quantitative attributes are partitioned into three linguistic terms {Small, Average, Large}, the time interval into three linguistic terms {Short, Medium, Long}, the user – defined threshold min_sup is 0.14.

Some of these patterns can be explained as follows:

⟨⟨POSTAGE$_{_Small}$, **Short**, POSTAGE$_{_Small}$⟩⟩ means "**If** a customer buys a Small number of POSTAGE products, **then** he will buy a Small number of these products an interval of **Short**";

⟨⟨POSTAGE$_{_Small}$, **Medium**, RABBIT-NIGHT-LIGHT$_{_Small}$⟩⟩ means "**If** a customer buys a Small number of POSTAGE products, **then** he will buy a Small number of RABBIT-NIGHT-LIGHT an interval of **Medium**";

⟨⟨POSTAGE$_{_Small}$, **Medium**, POSTAGE$_{_Small}$, **Short**, POSTAGE$_{_Small}$⟩⟩ means "**If** a customer buys a Small number of POSTAGE products and later a Small number of POSTAGE products an interval of **Medium**, **then** he will buy a Small number of POSTAGE products an interval of **Short**".

Table 5.The fuzzy sequential patterns with fuzzy time-intervals in Online Retail_France dataset

Fuzzy sequential patterns with fuzzy time-intervals	Support
⟨⟨POSTAGE$_{_Small}$, **Short**, POSTAGE$_{_Small}$⟩⟩	0.236
⟨⟨POSTAGE$_{_Small}$, **Medium**, POSTAGE$_{_Small}$⟩⟩	0.257
⟨⟨POSTAGE$_{_Small}$, **Short**, RABBIT-NIGHT-LIGHT$_{_Small}$⟩⟩	0.151
⟨⟨POSTAGE$_{_Small}$, **Medium**, RABBIT-NIGHT-LIGHT$_{_Small}$⟩⟩	0.161
⟨⟨POSTAGE$_{_Average}$, **Short**, POSTAGE$_{_Small}$⟩⟩	0.141
⟨⟨POSTAGE$_{_Small}$, **Short**, POSTAGE$_{_Small}$, **Short**, POSTAGE$_{_Small}$⟩⟩	0.146
⟨⟨POSTAGE$_{_Small}$, **Medium**, POSTAGE$_{_Small}$, **Short**, POSTAGE$_{_Small}$⟩⟩	0.144

5. Conclusions and future work

The paper proposes the FSPFTIM algorithm to mine fuzzy sequential patterns with fuzzy time-intervals in quantitative sequence databases. The proposed algorithm is presented in both idea and pseudo code. An example of application of the algorithm is also illustrated in this paper. The experimental results show the influence of some input parameters to the number of fuzzy sequential patterns with fuzzy time-intervals and the execution time. In this paper, the FSPFTIM algorithm is developed based on the Apriori algorithm with the approach of search in breadth first. In the future, we will continue to work on developing more efficient algorithms to mine fuzzy sequence patterns with fuzzy time – intervals. These algorithms will be developed based on the approach of search in depth first, such as based on the projected database organization [10] or the FPTree algorithm [11].

References

1. Agrawal, R., R. Srikant. Mining Sequential Patterns. – In: Proc. of 11th International Conference on Data Engineering, 1995.
2. Cao, H., et al. A Fuzzy Sequential Pattern Mining Algorithm Based on Independent Pruning Strategy for Parameters Optimization of Ball Mill Pulverizing System. – Information Technology and Control, Vol. **43**, 2014, No 3, pp. 303-314.
3. Chang, C.-I., H.-E. Chueh, N. P. Lin. Sequential Patterns Mining with Fuzzy Time-Intervals. – In: 6th International Conference on FSKD'09, Vol. **3**, 2009.
4. Chang, C.-I., H.-E. Chueh, d Y.-C. Luo. An Integrated Sequential Patterns Mining with Fuzzy Time-Intervals – In: International Conference on Systems and Informatics (ICSAI'12), 2012.

5. C h e n, R.-S. et al. Discovery of Fuzzy Sequential Patterns for Fuzzy Partitions in Quantitative Attributes. – In: ACS/IEEE International Conference on Computer Systems and Applications, 2001, pp. 144-150.
6. C h e n, Y.-L., T. C.-K. H u a n g. Discovering Fuzzy Time-Interval Sequential Patterns in Sequence Databases. – IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), Vol. **35**, 2005, No 5, pp. 959-972.
7. C h e n, Y.-L., M.-C. C h i a n g, M.-T. K o. Discovering Time-Interval Sequential Patterns in Sequence Databases. – Expert Systems with Applications, Vol. **25**, 2003, No 3, pp. 343-354.
8. F o w k e s, J., C. S u t t o n. A Subsequence Interleaving Model for Sequential Pattern Mining. – arXiv preprint arXiv:1602.05012, 2016.
9. G a r o f a l a k i s, M. N., R. R a s t o g i, K. S h i m. SPIRIT: Sequential Pattern Mining with Regular Expression Constraints. – VLDB, Vol. **99**, 1999.
10. H a n, J., et al. Prefixspan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. – In: Proc. of 17th International Conference on Data Engineering, 2001.
11. H a n, J., J. P e i, Y. Y i n. Mining Frequent Patterns without Candidate Generation. – In: ACM Sigmod Record, Vol. **29**, 2000.
12. H o n g, T.-P., C.-S. K u o, S.-C. C h i. Mining Fuzzy Sequential Patterns from Quantitative Data. – In: Proc. of Conference IEEE SMC'99, Vol. **3**, 1999.
13. H u, Y.-C., et al. A Fuzzy Data Mining Algorithm for Finding Sequential Patterns. – International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol. **11**, 2003, No 2, pp. 173-193.
14. H u, Y.-C., G.-H. T z e n g, C.-M. C h e n. Deriving Two-Stage Learning Sequences from Knowledge in Fuzzy Sequential Pattern Mining. – Information Sciences, Vol. **159**, 2004, No 1, pp. 69-86.
15. H u a n g, T., et al. Extracting Various Types of Informative Web Content via Fuzzy Sequential Pattern Mining. – Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint Conference on Web and Big Data. Springer, Cham, 2017, pp. 230-238.
16. H u a n g, T. C.-K. Discovery of Fuzzy Quantitative Sequential Patterns with Multiple Minimum Supports and Adjustable Membership Functions. – Information Sciences, Vol. **222**, 2013, pp. 126-146.
17. K i e u, T., et al. Mining Top-K Co-Occurrence Items With Sequential Pattern. – Expert Systems with Applications, Vol. **85**, 2017, pp. 123-133.
18. K u o, R. J., C. M. C h a o, C. Y. L i u. Integration of k-Means Algorithm and AprioriSome Algorithm for Fuzzy Sequential Pattern Mining. – Applied Soft Computing, Vol. **9**, 2009, No 1, pp. 85-93.
19. A g r a w a l, R., R. S r i k a n t. Fast Algorithms for Mining Association Rules. – In: J. Bocca, M. Jarke, C. Zaniolo, Eds. Proc. of 20th International Conference on Very Large Data Bases (VLDB'94). Santiago, Morgan Kaufmann, 1994, pp. 487-499.
20. T a n, P.-N., M. S t e i n b a c h, V. K u m a r. Association Analysis: Basic Concepts and Algorithms. – Introduction to Data Mining, 2005, pp. 327-414.
21. P h u o n g, T. D., D. V. T h a n h, N. D. D u n g. An Effective Algorithm for Association Rules Mining from Temporal Quantitative Databases. – Indian Journal of Science and Technology, Vol. **9**, 2016, No 17.
22. W r i g h t, A. P., et al. The Use of Sequential Pattern Mining to Predict Next Prescribed Medications. – Journal of Biomedical Informatics, Vol. **53**, 2015, pp. 73-80.
23. Y u, C.-C., Y.-L. C h e n. Mining Sequential Patterns from Multidimensional Sequence Data. – IEEE Transactions on Knowledge and Data Engineering, Vol. **17**, 2005, No 1, pp. 136-140.
24. UCI-Machine Learning Repository. June 2017.
http://archive.ics.uci.edu/ml/datasets.html

Appendix

The computational complexity of the FSPFTIM algorithm is estimated as follows:

•*Creating the fuzzy sequence database D′*: Each fuzzy transaction sequence in D' has l.h fuzzy attributes, so the computational complexity of converting the quantitative sequence database D to the fuzzy sequence database D' is $O(N.l.h)$.

•*Generating C_1 and calculating the support of sequences in C_1*: C_1 includes the $M.h$ candidate 1-fuzzy sequences with fuzzy time-intervals (in this case, they are the fuzzy attributes in D'). To calculate the support for each candidate sequence in C_1, it is necessary to look at all fuzzy transaction sequences in D' as well as all fuzzy attributes in these transaction sequences. So, the computational complexity is $O(M.h).\ O(N.l.h) = O(M.N.l.h^2)$.

•*Generating C_2 and calculating the support of sequences in C_2 (or creating L_2* for short):

•C_2 – the set of candidate 2-fuzzy sequences with fuzzy time-intervals is created by joining two sequences in L_1 together and with LT, namely, $C_2=L_1\times \text{LT}\times L_1$. The number of candidate sequences in C_2 can be $|L_1|^2.|\text{LT}|$. To calculate the support of each candidate sequence in C_2, we have to look at all fuzzy transaction sequences in D' as well as all 2-fuzzy sequences in these transaction sequences, i.e., the computational complexity of calculating the support of each candidate sequence in C_2 is $O(N.\binom{2}{l.h})$. Hence the computational complexity of creating L_2 is $O(|L_1|^2.|\text{LT}|)\ .$ $O(N.\binom{2}{l.h}) = O|L_1|^2.\ \binom{2}{l.h}.N.|\text{LT}|)$.

•*Generating C_k and calculating the support of sequences in C_k with k>2* (or *creating L_k* for short): C_k is generated by merging two (k–1)-fuzzy sequential patterns with fuzzy time-intervals in L_{k-1}, so the number of sequences in C_k can be $|L_{k-1}|^2.|\text{LT}|$. Check of merging conditions requires a maximum of $2k–5$ comparisons, including $k–2$ comparisons for fuzzy attributes and $k–3$ comparisons for fuzzy time-intervals. Hence, the cost of generating C_k is $O((2k–5).\ |L_{k-1}|^2.|\text{LT}|)$. And similar to the above, the computational complexity of each sequence in C_k is $O(N.\binom{k}{l.h})$. Therefore the computational complexity of creating L_k (k >2) is

$$O((2k–5).\ |L_{k-1}|^2.|\text{LT}|).\ O(N.\binom{k}{l.h}) = O((2k–5).\ |L_{k-1}|^2.\binom{k}{l.h}.N\ |\text{LT}|).$$

Finally, the computational complexity of the proposed algorithm is

$$O(N.l.h + M.N.l.h^2 + |L_1|^2.\ \binom{2}{l.h}.N.|\text{LT}|) + \sum_{k=3}^{l.h}(2k-5).|L_{k-1}|^2.\binom{k}{l.h}.N\ |\text{LT}|\ ) =$$

$$O(N.l.h + M.N.l.h^2 + |L_1|^2.\ \binom{2}{l.h}.N.|\text{LT}|) + \sum_{k=3}^{l.h}|L_{k-1}|^2.\binom{k}{l.h}.N\ |\text{LT}|) \blacksquare$$

Note: L_k (k≥1) depends so much on min_sup.

Received 20.08.2017; Second Version 20.05.2018; Accepted 28.05.2018