Volume 5, Issue 8, August 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Mining Sequential Patterns with Time Intervals by Fuzzy Sets

Dr. V Purna Chandra Rao
Nizamuddin Abid Mohammed
Lords Inst of Eng &Technology
Himayat Sagar, Hyderabad, India

Abstract: Prearranged a sequence database and minimum sustain verge; the errand of sequential pattern mining is to determine the absolute combinations of sequential patterns in databases. Commencing the exposed sequential patterns, we can be acquainted with what items are recurrently bought in concert and in what sort they appear. Though, they cannot inform us the time gaps concerning consecutive items in patterns. Appropriately, Chen, Chiang and Ko have projected a generalization of sequential patterns, called time-interval sequential patterns, which discloses not only the order of items but also the time intervals between consecutive items (Chen et al. 2003). An instance of time-interval sequential pattern has a form like (X, t2, Y, t1, Z), significance that we buy X first, then after an interval of t2 we buy Y, and at last after an interval of t1 we buy Z, where t2 and t1 are prearranged time ranges. Even though this new category of pattern can improve the over disquiet, it reasons the pointed periphery problem. i.e, when a time interval is close to the margin of two prearranged time ranges, we more over disregard or exaggerate it. Consequently, this paper uses the concept of fuzzy sets to expand the original research so that fuzzy time-interval sequential patterns are exposed from databases. An efficient algorithm, the FTI-Apriori algorithm, is developed for mining fuzzy time-interval sequential patterns by modifying traditional Apriori algorithm. An experimental study is shown for the algorithm.

Keywords: data mining, sequential patterns, sequence data, time interval, fuzzy sets

I. INTRODUCTION

Data mining extorts inherent, formerly unidentified with potentially functional information in sequence from databases. The exposed acquaintance and information be effective for diverse relevance's, as well as business management, fraud detection, market analysis and decision support. Several move towards encompass been anticipated to winkle out information. According to (Han et al. 2000) the most of the important approaches comes from the data mining sequential patterns.

The predicament of mining sequential patterns is foremost commenced in the intermediate 1990s, which ascertains pattern with the intention of appears recurrently in a sequence database (Agrawal et al. 1995; Pei et al. 2000). A characteristic illustration of sequential pattern is an akin to that in which a client who, having bought a laptop, returns to buy a printer and a scanner. Even though an exposed sequential pattern able to disclose what items is normally bought simultaneously and in what coordinate they emerge, they cannot inform us the period interruptions concerning succeeding items. Regrettably, not perceptive the time intervals resources that, while we know what items will be bought subsequently, we encompass no proposal when the next acquisition will ensues; this composes it complex to take the accurate accomplishment at the accurate time. In an analysis of this predicament, Chen, Chiang and Ko (Chen et al. 2003) have projected a simplification of sequential patterns, called time-interval sequential patterns, which divulges not only the sort of items although more over the time intervals between consecutive items. The subsequent are various examples of the time-interval sequential pattern: (a) having bought a scanner, a customer returns to buy a CD burner in three months and then a laser printer in six months. (b) A client revisits browser X in a week. (c) Subsequent to a function A, a tolerant is extremely liable to be contaminated by virus B in two weeks.

At this time, we concisely paraphrase the method projected by Chen, Chiang and Ko (Chen et al. 2003). The participation of their predicament includes a sequence database *S*, a set of items *I* = {i1, *i2... im}* and a set of time intervals *T*= {t0, *t1, t2... tr*}, where *T* is a inclusive and non-overlap subset of the time domain. A sequence *Y*= *(y1,* &1, *y2,* &2... *yn-1,* &n-1, *yn*) is a time-interval sequence if *yi€I* for 1≤i≤n and & i€T for 1≤i≤n-1. The productivity is the entire time-interval sequences which appear regularly in database *S.* A paradigm of time-interval sequential pattern have a structure similar to (X, *t2,* Y, *t1,* Z), implication that we procure X first, after that subsequent to an interval of *t2* we procure Y, and conclusively subsequent to an interval of *t1* we procure Z, where *t2* and *t1* are prearranged time intervals.

Even though sequential patterns comprehensive with time-intervals are able to propose extra information than those exclusive of time-intervals, the method might reason the incisive boundary predicament. With the intention of be, while a time interval is close to the boundary of two adjoining ranges, we either disregard or exaggerate it. For paradigm, let the interval of *t2* be 10≤t<15 and that of *t3* be 15≤t<25, where *t* is the time gap between two consecutive items. Subsequently condition the time gap between items X and Y is close to 15, moreover a slight outsized or lesser 15, it is not reasonable to critic whether the time interval concerning X and Y is in *t2* or in *t3*. Conversely, according to the innovative definition of Chen, Chiang and Ko, it can only be 100% in *t2* or in *t3*. This impenetrability can be effectively accepted by with fuzzy techniques, for fuzzy set theory allocates this time gap to be fifty percentages in *t2* and at the same time 50% in *t3*. This uncomplicated paradigm denotes that the fuzzy perception is enhancing than the partition approach for the reason that fuzzy sets impart a smooth modification concerning member and non-member of a set.

Moreover the afore mentioned allowance, there are numerous other causes that sustain the use of fuzzy time interval in consign of crisp interval. First, the soul acquaintance can be epitomized further unsurprisingly and suitably by fuzzy logic. And how to subset and epitomize the time interval is an organizer of soul acquaintance. Second, it is commonly recognizable that various real world conditions are fundamentally fuzzy. And the partition of time interval is one of them. Third, fuzzy time interval is simple and easy for consumers. For example, if we use fuzzy sets to handle the time intervals, we can first delineate the linguistic provisos that are eloquent and explicable to users. Then, for each such expression we can desire appropriate fuzzy function to denote it.

A numeral of examines retain demoralized fuzzy techniques to excavation fuzzy association rules or sequential patterns from databases. These hard works can be approximately classified into the following: (1) fuzzy representation of item's quantity (Lee et al. 1997), (2) fuzzy representation of quantitative attribute (Hong et al. 1999; Zhang 1999), (3) fuzzy product taxonomies or generalization hierarchies (Chen et al. 2002), (4) fuzzy representation of item importance (Yue et al. 2000), (5)fuzzy representation of transactions (Lee 2000), (6) fuzzy support and confidence measure (Kuok et al. 1998), (7) using fuzzy techniques for determining linguistic terms or domain partition (Fu et al. 1998; Vazirgiannis 1998), and (8) using fuzzy techniques to determine rule's interestingness (Au et al. 1997; Au et al. 1998; Au et al. 1999; Au et al 2003).

To our acquaintance, no examine has perpetually functional fuzzy methods to compact with time intervals in time-interval sequential patterns. We, consequently, reach the original research of Chen, Chiang and Ko so that fuzzy time-interval sequential patterns can be exposed from databases. Some linguistic terms, such as Long, Middle, and Short, will be provided to represent time-intervals. And, a fuzzy time-interval sequential pattern may have a structure similar to: Having bought a laser printer, a customer returns to buy a scanner in a 'Short' period and then a CD burner in a 'Long' period.

The respite of this paper is prearranged as follows. Section 2 officially defines the problem and the fuzzy time-interval sequential pattern. Subsequently, Section 3 progresses an algorithm to find fuzzy time-interval sequential patterns, which is residential by modifying the traditional Apriori algorithm. Section 4 shows the accomplishment of the algorithm. Conclusions are finally drawn in Section 5.

II. PROBLEM DEFINITION

As done in the previous research of Chen, Chiang and Ko (Chen et al. 2003), we represent a sequence in the following way.

Definition 1.A sequence *s* is represented as ((a_1, t_1), (a_2, t_2), (a_3, t_3),…, (a_n, t_n)), where a_j is an item and t_j stands for the time at which a_j occurs, $1 \le j \le n$, and $t_{j-1} \le t_j$ for $2 \le j \le n$. In the sequence, if items occur at the same time, they are ordered alphabetically.

From the interval values as $ti_j = |t_{j+1} - t_j|$, where j=1, 2… $n-1$. For example, if we have a sequence *s* as ((*a*, 1), (*b*, 4), (*e*, 29)), then its time interval values are 3 and 25. Suppose we have the set $LT=\{lt_j \mid j=1, 2, \ldots, l\}$ of linguistic terms. Then we use $\mu_{lt_j}(ti)$ to denote the membership degree of time-interval value *ti* to linguistic term lt_j.

Two approaches have been used to determine linguistic terms and fuzzy membership functions (Medasani et al. 1998). The first approach relies on domain experts to specify the functions based on their background knowledge and requirements. The second approach assumes that the functions are obtained by a preprocessing phase that learns the functions from the data, such as learning by neural-network (Lin et al. 1991), by genetic algorithm (Karr et al. 1993), by clustering method (Fu et al. 1998), and by entropy measure (Ross 1995). Therefore, a complete process in fuzzy mining may contain two phases, where the first phase learns fuzzy functions from data and the second phase discovers patterns according to the fuzzy functions learned from the first phase. Interestingly but not surprisingly, almost all of the existing papers in fuzzy mining only deal with the second phase by assuming that the fuzzy functions are given, because this can simplify the presentation of the paper and enable us to focus on the design of mining algorithms. Due to these reasons, we adopt the same assumption that the fuzzy functions are given.

Example 1. Suppose we want to represent a time interval by using three linguistic terms:
Short(*S*), *Middle*(*M*), and *Long*(*L*). Their membership functions can be represented as follows.

$$\mu_{Short}(ti_j)=\begin{cases}1, & ti_j \le 2\\ \dfrac{15-ti_j}{13}, & 2 < ti_j < 15\\ 0, & ti_j \ge 15\end{cases} \qquad \mu_{Middle}(ti_j)=\begin{cases}0, & \text{either } 2 \le ti_j \text{ or } ti_j \ge 28\\ \dfrac{ti_j-2}{13}, & 2 < ti_j \le 15\\ \dfrac{28-ti_j}{13}, & 15 < ti_j < 28\end{cases}$$

$$\mu_{Long}(ti_j)=\begin{cases}0, & ti_j \le 15\\ \dfrac{ti_j-15}{13}, & 15 < ti_j < 28\\ 1, & ti_j \ge 28\end{cases}$$

Fig. 1. The fuzzy membership functions for time-interval concept.

By applying the fuzzy functions above, we find that the time-interval value 3 is 0.0/*Short+0.92/Short* + 0.08/*Middle* + 0.0/*Long* and the time-interval value 25 is 0.0/*Short* +0.23/*Middle* + 0.77/*Long*. According to the linguistic terms and the membership functions, we can define the fuzzy time-interval sequence as follows.

Definition 2. Let $I = \{i_1, i_2 \ldots i_m\}$ be the set of all items and $LT = \{lt_j \mid j=1, 2 \ldots l\}$ be the set of all linguistic terms. A sequence $\alpha = (b_1, lg_1, b_2, lg_2 \ldots b_{r-1}, lg_{r-1}, b_r)$ is a fuzzy time-interval sequence if $b_i \in I$ for $1 \le i \le r$ and $lg_i \in LT$ for $1 \le i \le r-1$.

Definition 3. Let $s = ((a_1, t_1), (a_2, t_2), (a_3, t_3) \ldots (a_n, t_n))$ be a sequence and $\alpha = (b_1, lg_1, b_2, lg_2, \ldots, b_{r-1}, lg_{r-1}, b_r)$ be a fuzzy time-interval sequence, where $r \ge 2$. Let $\mu_{lg_i}(t)$ denote the membership degree of time-interval value t to linguistic term lg_i.Suppose there are K lists of indexes in s, denoted as $1 \le w_{k,1} < w_{k,2} < \ldots < w_{k,r} \le n$ for k=1 to K, each of which satisfied the condition of $b_1 = a_{w\,k,1}$, $b_2 = a_{w\,k,2}$,…, $b_r = a_{wk,r}$. Then we call that α is contained in S with degree γ or that α is fuzzy time- interval sub sequence of S with degree γ if the following conditions hold.

(1) $ti_{wk,i} = |t_{wk,i+1} - t_{wk,i}|$ for i=1,2,...,r-1 and k= 1,2,…k.

(2) $\gamma = \max_{1 \le k \le K} \min_{1 \le i \le r-1} \{ \mu_{lgi}(ti_{wk,i}) \}$

Although Definition 3 seems to be a right definition, it does not consider the situation of *r*=1, where the fuzzy time-interval sequence degenerates into a crisp sequence containing a single item. To make the definition complete, we do the following amendment.

Definition 4. When a fuzzy time-interval sequence only contains a single item, it can be represented as $\alpha = (b_1)$, where $b_1 \in I$. In such a case, we call that α is *contained* in *s* with degree 1 if there exists an integer *j*, where $1 \le j \le n$, such that $b_1 = a_j$. The total number of items in a fuzzy time-interval sequence α is referred to as the *length* of the sequence. A fuzzy time-interval sequence whose length is *k* is referred to as a fuzzy *k*-time-interval sequence.

Example 2. Suppose we are given a sequence *s*= ((*a*, 4), (*d*, 5), (*d*, 10), (*e*, 28)) and a fuzzy time-interval sequence α= (*a*, *Short*, *d*, *Middle*, *e*). There are two ways that we can match α: one is ((*a*, 4), (*d*, 5), (*e*, 28)) and the other is ((*a*, 4), (*d*, 10), (*e*, 28)). For the first case, we have the degree as min $\{\mu_{Short}(1), \mu_{Middle}(23)\}$ = min {1, 5/13} =0.385. The second case has the degree as min $\{\mu_{Short}(6), \mu_{Middle}(18)\}$ = min {9/13, 10/13} = 0.692. Consequently, α is contained in s with degree max {0.385, 0.692} =0.692.

For ease of reference, let γ (α, *s*) represent the degree that a fuzzy time-interval sequence α is contained in sequence *s*, which is determined according to Definitions 2, 3 and 4. A transaction is represented by <*sid*, *s*>, where *sid* is the identifier of this transaction and *s* is a sequence. A sequence database *S* is formed by a set of transactions. For a given fuzzy time-interval sequence α, its support in database *S* is defined as follows.

Definition 5. $Support_S(\alpha) = \sum_{(Sid,\, s) \text{ in } S} \gamma(\alpha, s) / |S|$.

A fuzzy time-interval sequence α is called a fuzzy time-interval sequential pattern or a frequent fuzzy time-interval sequence if its support in *S* is greater than or equal to the user-specified minimum support (called min_*sup*). A fuzzy time-interval sequential pattern with length *k* is referred to as a fuzzy *k*-time-interval sequential pattern.

Given a sequence database and *min_sup*, the goal of fuzzy time-interval sequential pattern mining is to determine in the sequence database all the fuzzy time-interval subsequences whose supports are more than or equal to *min_sup*.

Sid	Sequence
10	((*a*, 1), (*b*, 4), (*e*, 29))
20	((*d*, 1), (*a*, 2), (*d*, 24))
30	((*b*, 1), (*a*, 11), (*e*, 28))
40	((*f*, 1), (*b*, 5), (*c*, 19))
50	((*a*, 4), (*b*, 5), (*d*, 10), (*e*, 28))
60	((*a*, 0), (*b*, 5), (*e*, 30))
70	((*j*, 2), (*a*, 17), (*h*, 17))
80	((*c*, 3), (*i*, 10), (*f*, 18))
90	((*h*, 4), (*a*, 10), (*b*, 21))
100	((*g*, 0), (*a*, 0), (*b*, 3), (*e*, 30))

Fig. 2. A sequence database

Example 3. Consider the sequence database shown in Fig. 2 with the linguistic terms defined in Example 1. If *min_sup*=0.3, then we can find fuzzy time-interval sequential pattern (*a*, *Short*, *b*, *Long*, *e*) with support 0.308 in the database. Four transactions (*Sid*=10, 50, 60 and 100) contribute to this pattern, whose degrees are respectively 0.77, 0.62, 0.77 and 0.92.

According to Definition 5, the support of this pattern is (0.77+0.62+0.77+0.92)/10=0.308.

III. ALGORITHMS FOR MINING FUZZY TIME - INTERVAL SEQUENTIAL PATTERNS

Theaspirationofthissegmentistoprogressanalgorithmforminingfuzzytime-interval sequential patterns from databases. Modifying the well-known Apriori algorithm develops the algorithm. We commence them in the following.

3.1. The FTI-Apriori algorithm

Modifying the well-known Apriori algorithm develops the Fuzzy Time Interval (FTI) -Apriori Algorithm. Fundamentally, two phases are constantly accomplished to spawn the patterns. The first phase generates *candidate sequences* of length r, denoted by C_r, from the *frequent sequences* of length r-1,designated by K_{r-1}.So, every candidate sequence produced in the existing succession will have one further it emend one extra linguistic term than the normal sequences in the precede cycle. After conclusion the set of candidate sequences, the second phase inspects the database to ascertain the sustain of each candidate precedent, and the consequential set embraces all frequent sequences of length r. In the following, we discuss how to execute the first phase for different values of k:

(1) For r=1: The set of candidate patterns of length1, *C1, will* be created by inventory all divergent items in databases.

(2)For r=2: Conventional, C_2 was acquired by exactly fusion L_1 with L_1. Conversely, as the first item and the second item in C_2, say p and q, might have numerous fuzzy time-interval relations, put together for all probable fuzzy time-interval relations must be engender. Let us contemplate an illustrative example. Suppose that (p) and (q) belong to L_1 and LT={l_1,l_2,l_3,l_4, l_5}.Therefore there are completely 20 candidate fuzzy time-interval sequences in C_2. Some of them are(p,l_1,p),(p,l_3,p),(p,l_2,p),(q, l_2,p)and(q,l_2,q). In a word, C_2 can be generated as $L_1 \bowtie TI \bowtie L_1$, where X denotes “join”.

(3)When r>2:Let(p_1,lg_1,p_2,lg_2,…,lg_{r-1},p_r) be a fuzzy r-time-interval sequence in K_r.

Then, the fuzzy(r-1)-time-interval sequences(p_1,lg_1,p_2,lg_2,…,lg_{r-2},p_{r-1})and(p_2 , lg_2,…,p_{r-1},lg_{r-1},p_r) should be too frequent ,for the reason that the sustain of(p_1,lg_1,p_2,lg_2,…,lg_{r-1},p_r) obligation be no superior than the supports of the other two .If the time-interval sequences (p_1,lg_1,p_2,lg_2,…,lg_{r-2},p_{r-1}) and (p_2,lg_2,…,p_{r-1},lg_{r-1},p_r) survive in K_{r-1}, then (p_1,lg_1,p_2,lg_2,…,lg_{r-1},p_r) must be in C_r. The entire the time-interval sequences in C_r *can* be caused by joining the time-interval sequences in L_{r-1}this method.

Subsequently, we desire converse how to implement the second phase, which is to ascertain the supports of all patterns in C_r.To this end, a tree structure, called fuzzy candidate tree, is used as a basis. The fore most divergence lies in that the traditional approach connects each tree branch with an item name, where as in the new approach two components are attached an item name and a linguistic term.

Consider we are given a candidate set C_r. Primarily; we have an empty tree with a single root node. Then we inter leave every fuzzy time-interval pattern in C_r into the tree, just as how we assemble a prefix tree. After all the patterns in C_r have been included, the tree is built. Next, we will negotiate the tree for every transaction. For a given transaction, subsequent to concluding the traversal we can ascertain the degrees that the patterns in the tree are enclosed in that transaction. At last, after the tree has been traversed by all transactions they maintain value of every pattern is held in reserve in the consequent leaf node in the tree. So, we can determine what patterns are frequent and what are not.

In the subsequent, the fore most steps of the FTI-Apriori algorithm are scheduled. For simplicity, we exclude the detailed functions and steps.

The FTI- Apriori algorithm.

Input: Sequence Database S, Minimum Support *min_sup*, Linguistic Terms LT;
Output: The complete set of fuzzy time-interval patterns **Variable:** *x.cnt* is the support of time-interval sequence x
Process:

C_1 = *find_all_items*(S);
K_1= {xЄC_1|*x.cnt* ≥ *min_sup*}
For each j_1ЄL_1 {
For each j_2ЄL_1 {
For each *ltd* Є LT;
x=j_1**ltd**j_2;
Add x to C_2;
}
}
L_2= {xЄC_2|*x.cnt*≥*min_sup*}
For (r>2; L_{r-1}≠Ø; r++) do begin
{
C_r=*fuzzy_ apriori_ gen* (L_{r-1});

Build the fuzzy candidate tree from C_r; For each sequence s Є S
{Traverse the fuzzy candidate tree and accumulate the supports; } L_k = {xЄC_r| ($x.cnt$/ |S|) ≥ *min_sup*} }

Return ∪ K_r;

Example 4. Consider the sequence data base shown in Fig.2 and assume that we set *min_sup* as 0.3. C_1 will be generated as follows:
(*i*) 8,(*ii*) 7, (*iii*)2,(*iv*)2,(*v*)5,(*vi*)2,(*vii*) 1,(*viii*) 2, (*ix*)1,(*x*)1,
After that, we have K_1 = {*i*,*ii*,*v*} since their supports are bigger than *min_sup*. Subsequent to, C_2 can because d by joining K_1 with *LT* = {*Sh*, *Mid*, *Ln*} (assume *Sh* as short, mid as middle and ln as long), where their relationship functions are referred to Fig. 1.The '*i then ii*' pattern can be generated as the following:

For *i* then *ii* with different linguistic terms:
Sh = (0.92+1.0+0.77+0.31+0.92)/10=3.92/10=0.392
Mid = (0.08+0.0+0.23+0.69+0.08)=1.08/10=0.108
Ln = (0.0+0.0+0.0+0.0+0.0) = 0.0

Amongst the above three, only the pattern of '*i* then *ii* in *Sh*' can be produced in L_2 because its sustain is better than *min_sup*. Moreover this pattern, other patterns in L_2 consists: '*i* then *ii* in *Long* with sustain 0.361', and '*ii* then *v* in *L*ong maintain 0.308.'Subsequent to the generation of L_2,the algorithm starts to generate C_r and L_r for *r*>2. While the patterns in L_2 are (*i*,*Sh*,*ii*),(*i*,*Ln*,*v*)and(*ii*,*Ln*,*v*),the candidate pattern that we can produce for C_3 is (*i*,*Sh*,*ii*,*Ln*,*v*). The subsequent working out signifys that maintain of this pattern go over *min_sup*, and thus we have L_3 as{(*i*, *Sh*, *ii*, *Ln*, *v*)}.

Sid=10:min{(*i*, 0.92/*Sh*, *ii*), (*ii*, 0.77/*Ln*, *v*)} = 0.77
Sid=50:min{(*i*, 1.0/*Sh*, *ii*), (*ii*, 0.62/*Ln*, *v*)}=0.62
Sid=60:min{(*i*, 0.77/*Sh*, *ii*), (*ii*, 0.77/*Ln*, *v*)} = 0.77
Sid=100:min{(*i*, 1.0/*Sh*, *ii*), (*ii*, 0.92/*Ln*, *v*)} = 0.92
Support= (0.77+0.62+0.77+0.92)/10=0.308

IV. EXPERIMENTAL RESULTS

In this segment, we present a simulation examine of the algorithm, FTI-Apriori. It is implemented by Sun Java language (J2SDK1.4.1_02) and test done a PC with two Intel Pentium III 933 processors and 1GB main memory under the Windows 2000 operating system. Neither the multi threading technology north parallel computing skill is used in our implemented programs.
Concerning the eminent synthetic data generation algorithm in Agrawal et al produces artificial datasets. (Agrawaletal. 1995). Fundamentally, each operation is a sequence of itemsets. Though, we extend the transaction data so that the items in dissimilar itemsets have dissimilar time values and that those in the same itemset have the same time values. A value *z* is drawn from a Poisson distribution with mean T_I for each customer. The drawn value z corresponds to them ean time interval betweenconsecutive itemsets in the sequence of this particular customer. Subsequent to that, we verify the intervals among consecutive itemsets of this customer by frequently drawing values from a Poisson distribution with average *z*.
Table I lists the parameters used in the simulation; the first eight parameters are the classic alones used in previous research but the last parameter T_I is a new parameter created for the problem considered here. In the simulation, some parameters are fixed: *M*=10000, M_S=5000,M_I=25000,T_I=15 and |*D*|=250000.

Table I Parameters

$\|D\|$	Number of customers
$\|C\|$	Average number of transactions per customer
$\|T\|$	Average number of items per transaction
$\|S\|$	Average length of maximal potentially large sequences
$\|I\|$	Average size of itemsets in maximal potentially large sequences
N_S	Number of maximal potentially large sequences
N_I	Number of maximal potentially large itemsets
N	Number of items
T_I	Average length of time intervals

Table IIParameters

Name	$\|C\|$	$\|T\|$	$\|S\|$	$\|I\|$
C10-T2.5-S4-I1.25	10	2.5	4	1.25
C10-T5-S4-I1.25	10	5	4	1.25
C10-T5-S4-I2.5	10	5	4	2.5
C20-T2.5-S4-I1.25	20	2.5	4	1.25
C20-T2.5-S4-I2.5	20	2.5	4	2.5
C20-T2.5-S8-I1.25	20	2.5	8	1.25

The first association would associate the run times of these seven algorithms for different minimum supports. The comparison is carried out on the basis of the six data sets shown in Table II, where the minimum support threshold is varied from 3.0% to 1.5%. Fig. 4 summarizes the results.

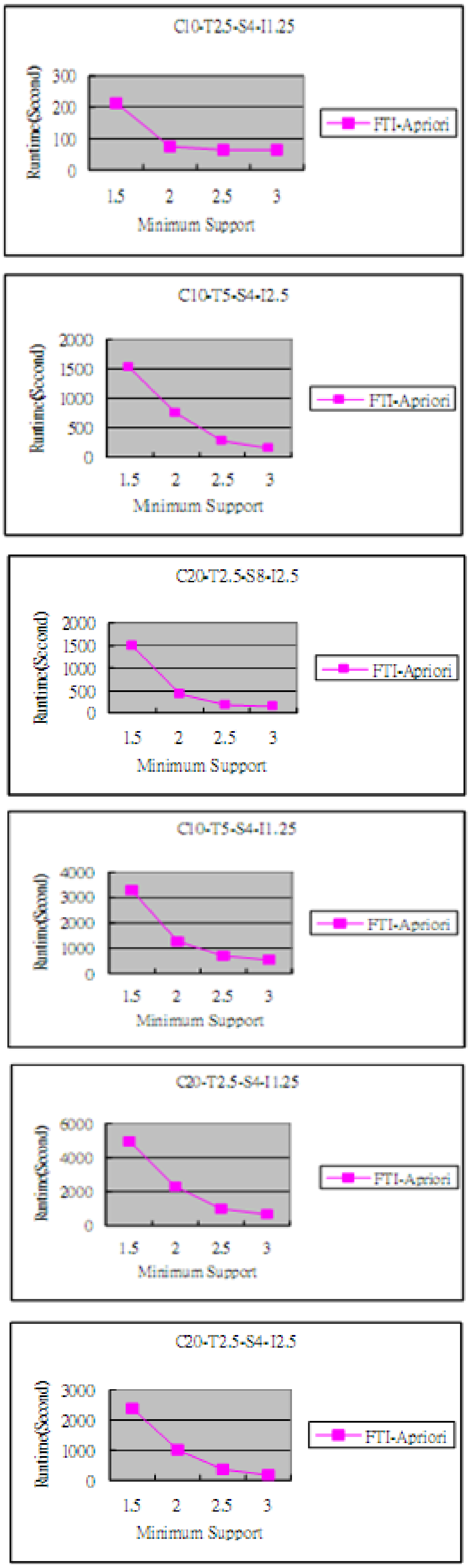

Fig.4. Run times for the six data sets.

V. CONCLUSION

Sequential-pattern mining is effective in ascertaining customer asset patterns along time from transactional databases. Since the approach was first proposed by Agrawal et al.(Agrawaletal.1995) in 1995, it has become an recognized and active research area. The presented approaches, though, do not determine the time intervals concerning succeeding items in the pattern. In inspection of this problem, Chen, Chiang and Ko proposed an ovel method to determine the time-interval information between consecutive items in the pattern. With this extra information, we can know when the next purchase will happen after the previous purchase was made.

Even though time-interval sequential patterns can present additional information than those without time-intervals, the approach may cause the sharp boundary problem. That is, when a time interval is close to the boundary of two adjoining assortments, we either disregard or exaggerate it. Consequently, this paper uses the perception of fuzzy set sto continue the imaginative research of Chen, Chiang and Ko so that fuzzy time-interval sequential pattern can be exposed from databases. Some linguistic terms, such as *Long*, *Middle* and *Short*, are provided to represent the linguistic terms for time-intervals.

Fuzzy time-interval sequential pattern mining characterizes a modern and capable research area in data mining. The Consequences of this paper can be comprehensive by taking into consideration time constraints, spatial constraints, fuzzy time-hierarchy and further kinds of time-related knowledge. Moreover, it is essential to survey how distinctive fuzzy membership functions may manipulate the result of mining.

REFERENCES

[1] Agrawal, R., and Srikant, R. “Fast Algorithms for Mining Association Rules,” in *Proceedings of 1994 International Conference Very Large Data Bases*,1994,pp.487-499.

[2] Agrawal, R., and Srikant, R. “Mining SequentialPatterns,” in *Proceedings of 1995*

[3] *International Conference Data Engineering*, 1995, pp. 3-14.

[4] Au, W.H.,andChan,K.C.C.“Mining fuzzy association rules, ”in *Proc.6thInt.Conference Information Knowledge Management*, Las Vegas, NV, 1997, pp.209-215.

[5] Au, W.H., and Chan, K.C.C.“An effective algorithm for discovering fuzzy rules in relational databases,”in*Proceedings IEEE International Conference Fuzzy Systems*, vol. II, 1998, pp. 1314-1319.

[6] Au,W.H., and Chan,K.C.C.“FARM: A data mining system for discovering fuzzy association rules,”in *Proceedings FUZZ-IEEE’99*, vol. 3, 1999, pp. 22-25.

[7] Au, W.H., and Chan, K.C.C.“Mining fuzzy association rules in a bank-account database,” *IEEE Transaction on Fuzzy Systems* (11),2003,pp.238-248.

[8] Chen,G.,andWei,Q.“Fuzzy association rules and the extended mining algorithms,” *Information Science*s (147), 2002, pp. 201-228.

[9] Chen,Y. L., Chiang, M. C., and Ko, M. T. “Discovering Time-interval Sequential Patterns in Sequence Databases, ”*Expert Systems with Applications*(25:3),2003, pp.343-354.

[10] Fu, A.W.C., Wong, M.H., Sze, S.C., Wong, W.C., Wong, W.L., and Yu, W.K. “Finding fuzzy sets for the mining of fuzzy association rules for numerical attributes, ”in *Proceedings International Symposium Intelligent Data Engineering Learning (IDEAL’98)*, Hong Kong,1998, pp. 263-268.

[11] Hong,T.P.,Kuo,C.S.,andChi,S.C.“Mining association rules from quantitative data,”*Intelligent Data Analysis*(3), 1999, pp. 363-376.

[12] Han,J.,Pei,J.,Mortazavi-Asl,B., Chen,Q., Dayal,U. , and Hsu,M.C .“FreeSpan: Frequent Pattern-projected Sequential Pattern Mining,” in *Proceedings of 2000 International Conference on Knowledge Discovery and Data Mining*, 2000, pp. 355-359.

[13] Karr, C.L., and Gentry, E.J.“Fuzzy control of p Husing genetic algorithms,”*IEEE Transactionon Fuzzy Systems* (1:1),1993,pp.46-53.

[14] Kuok, C.M.,Fu,A., and Wong,M.H.“Mining fuzzy association rules in data bases, ”*SIGMOD Record* (27:1), 1998, pp. 41-46.

[15] Lin,C.T.,andLee,C.S.G.“Neural network based fuzzy logic control and decision systems,” *IEEE Transaction on Computers* (40:12), 1991, pp. 1320-1336.

[16] Lee,J.H.,andKwang,H.L.“An extension of association rules using fuzzy sets, ”presented at the *IFSA’97*, Prague, Czech Republic, 1997.

[17] Lee,J.W.T.“An ordinal framework for data mining of fuzzy rules, ”in *FUZZI EEE2000*, San Antonio, TX, 2000, pp. 399-404.

[18] Liu,G. ,Lu,H., Xu,Y., and Yu,J.X. “Ascending frequency ordered prefix-tree: efficient mining of frequent patterns, ”in *Proceedings of the Eighth International Conference on Database Systems for Advanced Applications*, 2003, pp. 65-72.

