Mining Temporal Patterns for Interval-Based and Point-Based Events

[1,] S.Kalaivani , [2,] M.Gomathi, [3,] R.Sethukkarasi
[1,2,3,]*Department of Computer Science and Engineering,R.M.K. Engineering College, Kavaraipettai, Tamil Nadu, India.*

Abstract

Previous research on mining sequential patterns mainly focused on discovering patterns from point-based event data and interval–based event data, where a pair of time values is associated with each event. Since many areas of research includes data on a snapshot of time points as well as time intervals, it is necessary to define a new temporal pattern. In this work, based on the existing thirteen temporal relationships, a new variant of temporal pattern is defined for interval-based as well as point–based event data. Then, a hybrid pattern mining technique is proposed. Experimental results show that the completeness and accuracy of the proposed hybrid technique are more efficient than the existing algorithm.

Keywords: Data mining, temporal pattern, sequential pattern, interval-based event, point-based event.

I. INTRODUCTION

Data mining (also called knowledge discovery) is useful in various domains such as market analysis, decision support, fraud detection, business management, and so on [1], [2], [3]. Many approaches have been proposed to extract information and sequential pattern mining is one of the most important methods.The sequential pattern mining problem was first proposed by Agrawal and Srikant [4]. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.Temporal Data Mining is a single step in the process of Knowledge Discovery in Temporal Databases that enumerates structures (temporal patterns or models) over the temporal data. Any algorithm that enumerates temporal patterns from, or fits models to, temporal data is a Temporal Data Mining Algorithm.

Temporal data mining tasks include:

- Temporal data characterization and comparison,
- Temporal clustering analysis,
- Temporal classification,
- Temporal association rules,
- Temporal pattern analysis,
- Temporal prediction and trend analysis.

Sequence Data Mining provides balanced coverage of the existing results on sequence data mining, as well as pattern types and associated pattern mining methods. While there are several research on data mining and sequence data analysis, currently there are no research that balance both of these topics. This paper fills in the gap, allowing readers to access state-of-the-art results in one place.

II. LITERATURE SURVEY

In [1] Shin-Yi Wu et al proposed a new kind of non ambiguous temporal pattern for interval-based event data. Then, the TPrefixSpan algorithm is developed to mine the new temporal patterns from interval-based events. W.J. Frawley et al. in [2] discussed the use of domain knowledge within Data Mining and defined three classes of domain knowledge such as Hierarchical Generalization Trees (HG-Trees), Attribute Relationship Rules (AR-rules) and Environment Based Constrains (EBC). Jiawei Han et al [3] proposed a novel Sequential Pattern Mining Method named FreeSpan(**Fre**qu**e**nt Pattern-Projected **S**equential **Pa**ttern mining).The general idea of this method is integration of mining frequent sequences with that of frequent

patterns and use projected sequence databases to confine the search and the growth of subsequent fragments. Jian Pei et al [4] proposed a novel sequential pattern mining method, called PrefixSpan (**Prefix**-projected **S**equential **pa**ttern mining), which explores prefix projection in sequential pattern mining. PrefixSpan mines the complete set of patterns but greatly reduces the efforts of candidate subsequence generation. R. Agrawal and R. Srikant in [5] proposed an algorithm MS-PISA, which stands for Multi Support Progressive mIning of Sequential pAttens, which discovers sequential patterns in, by considering different multiple minimum support threshold values for every possible combinations of item or item set.

C.-C. Yu and Y.-L. Chen in [6] discovered frequently occurred sequential patterns from databases and also two efficient algorithms are developed to mine frequent sequential patterns from multi-dimensional sequence data. J. Pei et al in [7] proposed a novel sequential pattern mining method called PrefixSpan, which explores prefix-projection in sequential pattern mining. Prefixspan mines the complete set of patterns but generally rduces the efforts of candidate subsequence generation. R. Srikant and R. Agrawal in [8] proposed the use of constraint approximations to guide the mining process, reduce the number of discovered patterns. J. Pei, J. Han, and W. Wang in [9] proposed a general model of sequential pattern mining with a progressive database while the data in the database may be static, inserted or deleted. In addition to a progressive algorithm PISA, standing for Progressive mIning of Sequential pAtterns, to progressively discover sequential patterns in defined time period of interest was presented. P.S. Kam and A.W.C. Fu in [10] considered an interval-based events where the duration of events is expressed in terms of endpoint values, and these are used to form temporal constraint in the discovery process. introduce the notion of temporal representation which is capable of expressing the relationships between interval-based events. We develop new methods for finding such interesting patterns.

III. EXISTING SYSTEM

3.1Point-based event is represented by one end point. It has three possible projection methods between two events. Prefix Span algorithm is used for this event. The advantages of this algorithm are novel, scalable and efficient.

Example : *(PrefixSpan)* Let the sequence database S given in Table1 an min-support =2. The set of items in the database is {a,b,c,d,e,f,g}.

Table 1: A Sequence Database.

Sequence_No	Sequence
01	*< a (abc) (ac) d (cf) >*
02	*< (ad) c (bc) (ae) >*
03	*< (ef) (ab) (df) cb >*
04	*< eg (af)* cbc >

Step 1: Find length-1 sequential pattern.

Scan S once to find all frequent items in sequences. Each of these frequent items is a length-1 sequential pattern. They are as follows:

< a >: 4, < b >: 4, < c >:4, *< d >: 3, < e >:3,* and *< f >:3.*

Step: 2 Divide search space.

The sequential patterns can be partitioned into the following six subsets according to the six prefixes. Table 2 represents the projected database.

Table 2: Projected database.

Prefix	Projected (Postfix) Database
< a >	*< (abc) (ac) d (cf >, < (_d) c (bc) (ae) >, <(_b) (df) cb >, < (_f) (cbc) >*
< b >	*< (_c) (ac) d (cf) >, < (_c) (ae) >, < (df) cb >, < c >*
< c >	*< (ac) d (cf) >, < (bc) (ae) >, < b >, < bc >*
< d >	*< (cf) >, < c (bc) (ae) >, < (_f) cb >*
< e >	*< (_f) (ab) (df) cb >, < (af) cbc >*
< f >	< (ab) (df) cb >, < cbc >

Step 3: Find subsets of sequential patterns.
The subsets of sequential patterns can be mined by constructing corresponding projected databases and mine each recursively. Table 3 represents the sequential database.

Table 3: Sequential patterns.

Prefix	Sequential patterns
< *a* >	< *a* >, < *aa* >, < *a (bc)* >, < *a (bc) a* >, < *aba*>, < *abc* >, < *(ab) c* >, < *(ab) d* >, < *(ab) f* >, < *(ab) dc* >, < *ac* >, < *aca* >, < *acb* >, <*acc* >, < *ad* >, <*adc* >,< *af* >
< *b* >	< *b* >, < *ba* >, < *bc* >,< *(bc)* >, < *(bc) a* >, < *bd* >, < *bdc* >, < *bf* >
< *c* >	< *c* >, < *ca* >, < *cb* >, < *cc* >
< *d* >	< *d* >, < *db* >, < *dc* >, < *dcb* >
< *e* >	< *e* >, < *ea* >, < *eab* >, <*eac* >, < *eacb* >, < *eb* >, < *ebc* >, < *ec* >, < *ecb* >, < *ef* >, < *efb* >, < *efc* >, < *efcb* >
< *f* >	< f >, < fb >, < fbc >, < fc >, < fcb >

3.2. Interval-based event is represented by two end points. It has thirteen possible projection methods between two events according to the classification scheme proposed by Allen [32]: "before", "after", "during", "contain", "meet", "met by", "overlap", "overlapped by", "start", "started by", "finish", "finished by", and "equal". Based on these relationships, the patterns that they find may look like "(*a* during *b*)," which means that "event *a* occurs during event b."

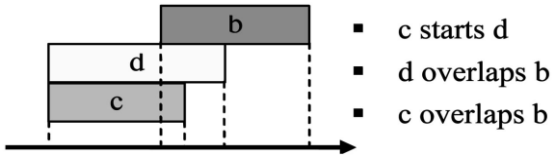

Fig.1. The temporal relationship of three event: b, c and d.

Definition 1: Event end points and order relations
An event e_i has two end points e_i^+ and e_i^-, called event end points, where e_i^+ is the starting point (esp) and e_i^- is the ending point (eep) of e_i. Let the time of an event end point, either esp or eep, be denoted as *time (u).* Then the order relation *Rel(u,v)* of two event end points *u* and *v* can be denoted as " < " if *time (u)* < *time (v)* and as " = " if *time (u) = time (v).*

Definition 2: Temporal sequence
The temporal sequence in Fig. 1 can be arranged in the following order: c^+, d^+, b^+, c^-, d^-, and b^-. A temporal sequence can be constructively defined as follows:
1. A temporal sequence of one event, called a 1-event temporal sequence, can be written as $(e_i^+ \oplus e_i^-)$, where $e_i \in \{1,\ldots t\}$, and $\oplus \in \{<\}$ is the order relation of e_i^+ and e_i^-.
2. Let $p = (p_1 \oplus_1 p_2 \oplus_2 \ldots \oplus_{i-1} p_i \oplus_i p_{i+1} \oplus_{i+1} \ldots \oplus_{j-1} p_j \oplus_j p_{j+1} \oplus_{j+1} \ldots \oplus_{2k-1} p_{2k})$
denote a temporal sequence of *k*-event (with *2k* event end points), called a *k*-event temporal sequence where p_i is an end point, and $\oplus_i \in \{<, =\}$ for all *i.* suppose p' is the temporal sequence obtained by inserting event e_a into *p*. Assume that e_a^+ must be placed between p_i and p_{i+1} and e_a^- must placed between p_j and p_{j+1}. Then we have,

$p' = (p_1 \oplus_1 p_2 \oplus_2 \ldots \oplus_{i-1} p_i \oplus_{i'} e_a^+ \oplus_a^+ p_{i+1} \oplus_{i+1} \ldots \oplus_{j-1} p_j \oplus_{j'} e_a^- \oplus_a^- p_{j+1} \oplus_{j+1} \ldots \oplus_{2k-1} p_{2k})$,
where,

$Rel(p_i, e_a^+) = \oplus_{i'}$, $Rel(e_a^+, p_{i+1}) = \oplus_a^+$, $Rel(p_j, e_a^-) = \oplus_{j'}$, *and* $Rel(e_a^-, p_{j+1}) = \oplus_a^-$.

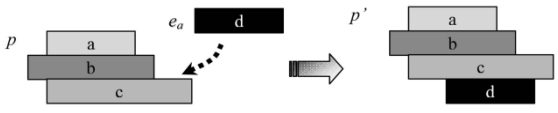

Fig.2. Construction of Temporal sequence.

3.3 TprefixSpan Algorithm:

TPrefixSpan is very similar to PrefixSpan. However, the input sequences and output patterns for these two methods are totally different. For example, the process to append a large 1-pattern to an original prefix in *TPrefixSpan* is much more complicated than that in *PrefixSpan*. The main reason for these differences is due to the fact that the input events and output patterns become interval-based instead of point-based, but in this paper the point-based events are also used as input. Therefore, "temporal prefix", "projection", and temporal Postfix" are used in the algorithm to design temporal patterns using *TPrefixSpan*.

3.4 Definition : (Temporal prefix)

Suppose we have two temporal sequences $\alpha = (p_1 \oplus_1 p_2 \oplus_2 \ldots \oplus_{n-1} p_n)$ and $\beta = (p'_1 \oplus'_1 p'_2 \oplus'_2 \ldots \oplus'_{m-1} p'_m)$, where m $\leq$ n. Let p'_x denote the last esp in β. Then, β is called a temporal prefix of α if and only if 1) β is contained in α, that is β is subset of α, 2) $p'_i = p_i$ (for $1 \leq i \leq x$), and 3) $\oplus'_i = \oplus_i$ (for $1 \leq i \leq x\text{-}1$).

3.5 Definition : (Projection)

Let $\alpha = (p_1 \oplus_1 p_2 \oplus_2 \ldots \oplus_{lastp-1} p_{lastp} \oplus_{lastp} p_{lastp+1} \oplus_{lastp+1} \ldots \oplus_{n-2} p_{n-1} \oplus_{n-1} p_n)$ and $\beta = (p'_1 \oplus'_1 p'_2 \oplus'_2 \ldots \oplus'_{x-1} p'_x \oplus'_x \ldots \oplus'_{m-1} p'_m)$ be two temporal sequence satisfying β is subset of α, where p'_x is the last esp in β, and p_{lastp} is the end point in α that matches p'_x. A projection of α with respect to temporal prefix β is temporal sequence α' is subset of α, which can be constructed by the following steps:

1) Set $\alpha' = (p'_1 \oplus'_1 p'_2 \oplus'_2 \ldots \oplus'_{x-1} p'_x \oplus_{lastp} p_{lastp+1} \oplus_{lastp+1} \ldots \oplus_{n-2} p_{n-1} \oplus_{n-1} p_n)$.

2) If p_i , where $(lastp+1) \leq i \leq n$, is an eep and its corresponding esp is not in α′ then delete p_i.

3) After deleting p_i set the order relation between p_{i-1} and p_{i+1} as small ($\oplus_{i-1}$, $\oplus$i).

3.6 Definition: (Temporal postfix)

Let $\alpha' = (p'_1 \oplus'_1 p'_2 \oplus'_2 \ldots \oplus'_{x-1} p'_x \oplus_{lastp} p_{lastp+1} \oplus_{lastp+1} \ldots \oplus_{n-2} p_{n-1} \oplus_{n-1} p_n)$.

be the projection of α with respect to β. Then temporal sequence $\gamma = (p'_x \oplus_{lastp} p_{lastp+1} \oplus_{lastp+1} \ldots \oplus_{n-2} p_{n-1} \oplus_{n-1} p_n)$ is called the temporal postfix of α with respect to temporal prefix β, denoted as $\gamma = \alpha / \beta$. If γ contains only one esp, that is p'_x, then we set γ = ϕ. The fig.3 illustrates the generation of projection and temporal postfix.

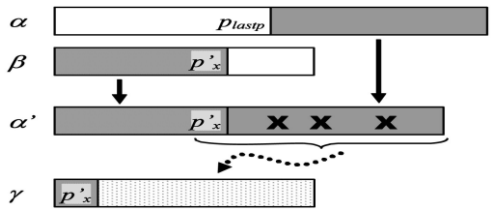

Fig.3.Illustrations for projection and postfix

IV. PROPOSED SYSTEM

4.1 HTPrefixSpan algorithm

Let us apply the HTPrefixSpan algorithm (which is given below) to the temporal sequence database S in Table 4. This table has both interval-based and point-based as an input.

Table 4: Temporal Sequence Database S

Patient	Temporal sequence
01	$(a^+ < e < b^+ < a^- < c^+ < f < b^- < g < c^-)$
02	$(a^+ < a^- < f < g < b^+ < c^+ < c^- < d^+ < h < d^- < b^-)$
03	$(a^+ < b^+ < e < f < b^- < c^+ < d^+ < c^- < g < d^- < a^-)$

The terms a+, b+, c+ d+ and a-, b-, c-, d- are the starting and ending events of the interval-based event respectively. Whereas, e, f, g, h are the point-based events. Initially we need to find,

$L_1 = \{ (a^+ < a^-), (b^+ < b^-), (c^+ < c^-), (d^+ < d^-), (e), (f), (g), (h) \}$

Consider $\alpha = (a^+ < b^+ < d^+ < e < g < c^+ < b^- < h < d^- < a^- < c-)$ and $\beta = (a^+ < b^+ < d^+ < e < g < c^+ < b^- < h)$.
Then the projection and temporal postfix for the above database S is given below respectively,

$\alpha = (a^+ < b^+ < d^+ < e < g < c^+ < b^- < h < a^- < c^-)$

$\gamma = (c^+ < b^- < h < a^- < c^-)$

After projecting S with respect to every frequent 1-pattern in L_1, we obtain eight projected databases, as shown in Table 5.

Table 5: Projected Database of L_1

Prefix	Projected Database
$(a^+<a^-)$	$(a^+ < e < b^+ < a^- < c^+ < f < b^- < g < c^-)$ $(a^+<a^- <f<g< b^+<c^+<c^- <d <h<d^- < b^-)$ $(a^+ < b^+ <e<f< b^- <c^+<d^+<c^- <g<d<a^-)$
$(b^+<b^-)$	$(b^+ < c^+ < f < b^- < g < c^-)$ $(b^+ < c^+ < c^- < d^+ <h< d^- < b^-)$ $(b^+ <e<f< b^- < c^+ < d^+ < c^- <g<d^-)$
$(c^+<c^-)$	$(c^+ < f < g < c^-)$ $(c^+ < c^- < d^+ <h< d^-)$ $(c^+ < d^+ < c^- <g<d^-)$
$(d^+<d^-)$	$(d^+ <h< d^-), (d^+ < c^- <g<d^-)$
(e)	$(e < b^+ < c^+ < f < b^- < g < c^-)$ $(e < f <c^+<d^+<c^- <g<d^-)$
(f)	$(f < g)$ $(f<g< b^+ < c^+ < c^- < d^+ <h< d^- < b^-)$ $(f <c^+<d^+<c^- <g<d^-)$
(g)	$(g), (g< b^+<c^+<c^- <d <h<d^- < b^-)$
(h)	(h)

Then the candidates can be generated by inserting the prefixes with one another.
For an example $\alpha = (a^+ < a^-)$ with L_1 { $(b^+ < b^-)$, $(c^+ < c^-)$, $(d^+ < d^-)$, (e), (f), (g), (h) }
Repeat the process till $\alpha = (g)$ with L_1 { h }
Consider $(a^+ < a^-)$ with $(b^+ < b^-)$ the possible insertion places are as follows:

$(a^+ < b^+ < a^- < b^-)$ ⟶ 1
$(a^+ < b^+ < b^- <a^-)$ ⟶ 2
$(a^+ < a^- < b^+ < b^-)$ ⟶ 3

From the above example 1, 2 and 3 are present in the database so they are consider as frequent temporal patterns. By repeating the process the following temporal patterns has been generated.

$(a^+ < c^+ < a^- < c^-)$
$(a^+ < a^- < c^+ < c^-)$
$(a^+ < d^+ < d^- < a^-)$
$(a^+ <a^- < d^+ < d^-)$
$(a^+ < e < a^-)$
$(a^+ < f < a^-)$
$(a^+ < a^- < e)$

$(a^+ < g < a^-)$
$(a^+ < a^- < g)$
$(a^+ < a^- < h)$
$(b^+ < c^+ < b^- < c^-)$
$(b^+ < c^+ < c^- < b^-)$
$(b^+ < d^+ < d^- < b^-)$
$(b^+ < b^- < d^+ < d^-)$
$(b^+ < e < b^-)$
$(b^+ < f < b^-)$
$(b^+ < b^- < g)$
$(b^+ < h < b^-)$
$(c^+ < d^+ < c^- < d^-)$
$(c^+ < c^- < d^+ < d^-)$
$(c^+ < f < c^-)$
$(c^+ < g < c^-)$
$(c^+ < c^- < g)$
$(c^+ < c^- < h)$
$(d^+ < g < d^-)$
$(d^+ < h < d^-)$
$(e < f)$
$(e < g)$
$(f < g)$
$(f < h)$
$(g < h)$

4.2 HTPrefixSpan Algorithm

Step1: Scan the database S to identify the interval-based and point-based inputs.
Step2: Find the projection and temporal postfix patterns using the above mentioned definition.
Step3: Using the prefix of length-1 generate projected database.
Step4: Generate the set of patterns using all the prefix combinations.
Step5: The generated patterns from step4 are compared with the database.
Step6: The coinciding patterns with the database are considered as the Temporal Patterns.

V. APPLICATIONS

This temporal pattern mining is applicable to various fields like financial services, census, stock fluctuations, library management, hospital management etc.,

VI. CONCLUSION AND FUTURE WORK

In this paper, a new algorithm Hybrid TPrefixSpan is proposed and implemented. This technique uses the approach of Tprefix span and extends the algorithm by including point based events also in the temporal sequence. Thus the proposed algorithm can be used for both interval-based and point-based events. This makes an efficient way of mining the exact data from the database. This work can be extended to find time span between frequent temporal patterns. The ambiguous problem of temporal pattern representation can also be considered in the future work.

REFERENCES

[1] Shin-Yi Wu and Yen-Liang Chen,"Mining Nonambiguous Temporal Patterns for Interval-Based Events", IEEE Transactions on Knowledge and Data Engineering,Vol.19,No.6,June 2007.
[2] W.J. Frawley, G. Piatetsky-Shapiro, and C.J. Matheus, Knowledge Discovery in Database: An Overview. AAAI/MIT Press, 1991.
[3] Jiawei Han,Jian Pei,Behzad Mortazavi-Asl,Qiming Chen Umeshwar Dayal,Mei-Chun Hsu,"FreeSpan: Frequent Pattern-Projected Sequential Pattern Mining",
[4] Jian Pei,Jiawei Han,Behzad Mortazavi-Asl,Helen Pinto, Qiming Chen Umeshwar Dayal Mei-Chun Hsu, "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth".
[5] R. Agrawal and R. Srikant, "Mining Sequential Patterns", Proc. 11th Int'l Conf. Data Eng. (ICDE '95), pp. 3-14, Mar. 1995.
[6] C.-C. Yu and Y.-L. Chen, "Mining Sequential Patterns from Multi-Dimensional Sequence Data", IEEE Trans. Knowledge and Data Eng., vol. 17, no. 1, pp. 136-140, Jan. 2005.
[7] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth", Proc. 17th Int'l Conf. Data Eng.(ICDE '01), pp. 215-224, 2001.
[8] R. Srikant and R. Agrawal, "Mining Sequential Patterns: Generalizations and Performance Improvements," Proc. Fifth Int'l Conf.Extending Database Technology (EDBT '96), pp. 3-17, 1996.
[9] J. Pei, J. Han, and W. Wang,"Mining Sequential Patterns with Constraints in Large Databases," Proc. 11th Int'l Conf. Information and Knowledge Management (CIKM '02), 2002.
[10] P.S. Kam and A.W.C. Fu, "Discovering Temporal Patterns for Interval-Based Events," Proc. Second Int'l Conf. Data Warehousing and Knowledge Discovery (DaWaK '00), 2000.