Journal of Computer Science, 9 (2): 225-234, 2013
ISSN 1549-3636

doi:10.3844/jcssp.2013.225.234 Published Online 9 (2) 2013 (http://www.thescipub.com/jcs.toc)

Efficient Approach to Discover Interval-Based Sequential Patterns

Sadasivam, R. and K. Duraiswamy

Department of Information Technology,
K.S. Rangasamy College of Technology, Tiruchengode-637 215, Namakkal District, India

Received 2012-07-13, Revised 2012-08-30; Accepted 2013-04-11

ABSTRACT

In most of the sequential pattern mining methodology they have concentrated only on time point base event data. But some research efforts have detailed the mining patterns from time interval based event data. In many application most of the events are occurred at time interval based event not a point based interval for example patient affected by the certain time period. Our goal is to mine the frequently occurred sequential patterns in the database. In this study we have introduced a new algorithm namely KPrefixspan by modifying the TPrefixspan algorithm to overcome the demerits of that algorithm. Here new approach called refined database can reduce the scanning time extremely since the unsupported events are removed at each projection also result of the sequential pattern is extremely precise. Experiments constructed for synthetic datasets. From the experimental results we reduced the running time almost 60% and also reduce the memory usage almost 25% when compared to the existing TPrefixspan algorithm.

Keywords: Datamining, TPrefixspan, KPrefixspan, Sequential Disease, Refined Database, Projected Database

1. INTRODUCTION

Generally, data mining tasks can be classified into two main types: Descriptive mining and Predictive mining. Descriptive data mining refers to the depiction of a dataset in a brief and summarized manner and discloses the significant properties of the data. Generalization is the basis of descriptive data mining approaches, which can be used to shorten the data by applying attribute-oriented induction with the aid of characteristic rules and generalized relations (Han and Kamber, 2001). Some of the descriptive mining techniques are Clustering (Liu and Yu, 2005), Association Rule Mining and Sequential Pattern Mining. On the other hand, predictive mining is the process of deriving patterns from data to make predictions. Classification, Regression and Deviation detection are some of the most important processes concerned in predictive mining techniques. Concisely, descriptive data mining aims to summarize the data and also highlights their interesting properties, while predictive data mining aspires to build models to forecast future behaviors (Han and Kamber, 2001).

Sequential pattern mining is one of the imperative subjects of data mining, which is an additional endorsement of association rule mining (Masseglia *et al*., 2003). The sequential pattern mining algorithm deals with the problem of finding the existing frequent sequences in a given database. Sequential pattern mining is strongly related to association rule mining, excepting that the events are associated by time (Sobh, 2007). Sequential patterns signify the association among transactions while association rules describe the intra transaction relationships. In association rule mining, the mined output is about the items that are bought together frequently in a single transaction. Whereas, the output of sequential pattern mining represents the items which are bought in a particular order by the same customer in diverse transactions (Zhao and Bhowmick, 2003).

Corresponding Author: Sadasivam, R., Department of Informaion Technology, K.S. Rangasamy College of Technology, Tiruchengode-637 215, Namakkal District, India

Most database related applications are temporal in nature, for example, financial applications such as portfolio management, accounting and banking; most of the applications depend on temporal databases that record time-referenced data (Jensen, 2000). Although much successful research has been made in the field of 'static' data mining, still there's much scope for further research regarding its extension to temporal data mining, wherein the temporal dimension is represented and reasoned about explicitly (Moskovitch and Shahar, 2005).

Time series prediction, sequence classification, sequence clustering, search and retrieval of sequences and pattern discovery are the five most important processes carried out for achieving temporal data mining tasks (Laxman and Sastry, 2006). Among them, Pattern Discovery has drawn a great deal of attention owing to its substantial use in stock trend prediction and application that using the history of symptoms to diagnose certain kind of diseases. There are two prominent frameworks for frequent pattern discovery: sequential patterns and episodes (Laxman and Sastry, 2006). In temporal data mining, mining of large sequential datasets is carried out, where the data is ordered with respect to some index (Antunes and Oliveira, 2001).

Mining of sequential pattern in time series data is often carried out in various fields in order to make a prediction and an opposite model should be proposed before the prediction can be done, therefore, the way how to discover time series pattern from time series database becomes extremely significant (Zhu *et al*., 2009). The sequence of events corresponds to a sequence of instants when these events happen. But, there are various situations where events have certain duration and so, the underlying time is computed in terms of intervals instead of points. Our work is motivated by several prior researches which are related to mining of temporal sequences from the time interval data (Guyet and Quiniou, 2011; Chen *et al*., 2010; Wu and Chen, 2007; Patel *et al*., 2008).

In this study, we proposed an algorithm called KPrefixspan to extract the frequently occurred sequential patterns. Our ultimate goal is to mine the time interval based sequential patterns efficiently, so that the frequently occurred sequential data's are computed by using the KPrefixspan algorithm. The proposed approach comprises three major steps: (i) creating refined database, (ii) constructing patterns based on time interval and (iii) mining sequential patterns based on projection database. In the projection stage, sequences having different length are selected from each projection such as, one length pattern, two length pattern. In each projection, the unsupported events are removed for reducing the scanning time in order to obtain accurate results.

1.1. Problem Statement

1.1.1. Database

Database DB consist a set of patient $P = \{P_1,..., P_i\}$, $1 \leq i \leq k$ where k illustrate the total number of patient where each patient p_i having the list of disease $D = \{d_1,...,d_j\}$ $1 \leq j \leq 1$, where 1 is the total number of disease for each patient. The each disease d_j having the time intervals which are starting time corresponds to t_s and ending time corresponds to t_e where the starting time of the disease always less than ending time of the disease $t_s < t_e$.

1.2. Generation of Refined Database (RDB)

The refined database is constructed from the original database which consists of less number of diseases when compared with the original database DB. We construct the RDB by removing the diseases having the value that will be less than threshold value T_h. The threshold value must be less than number of patient k. Calculate the number, from the DB for how much patients are affected by each disease N (d_j). The diseases are removed from the DB when the value of N (d_j) becomes less than the value of threshold N (d_j) < T_h. The remaining diseases are placed in the Refined Database (RDB).

1.3. Building the Sequences of Diseases P_i [S (d_j)]

From the refined database we can build the disease sequence, the sequences of diseases will form by sorting the diseases of ascending order depends on the time interval. This disease sequences P_i [S (d_j)] are input for the KPrefixspan algorithm.

1.4. Proposed Mining Technique to Mine the Sequential Disease of the Patient from the Hospital Database

In the proposed algorithm we have developed the efficient algorithm to mine the sequential disease of the patient by overcome the challenges of TPrefixspan algorithm that is used by Wu and Chen (2007). The major complexities occurred in the TPrefixspan algorithm to mine the sequential diseases from the database are 1. Running time is high 2. Need large memory space. Bearing in mind the above challenges, we have proposed an efficient mining technique called as KPrefixspan algorithm for mining the sequential disease based on the time interval of each disease. In the proposed algorithm, the steps involved in mining of

interval based sequence of patient diseases are achieved with three major steps. They are:

- Making the refined database from the original database
- Constructing the disease sequences with starting time and ending time
- Mining the disease sequences using the projection method

1.5. Algorithm Procedure

Input : Database DB
Output : Sequential diseases

1.6. Parameters

DB = Database
RDB = Refined database
K = Maximum number of patients
L = Maximum number of diseases
T_h = Threshold value
$N(d_j)$ = Count value of the disease
$P_i[S(d_j)]$ = Sequences of disease

1.7. Pseudo Code

```
Begin
    Call Pi[S(dj)]
      for all patient pi
        project disease for all dj in each pi
          calculate number of diseases N(dj)
            if N(dj)<Th
               remove that disease dj from pi
           else
            go to next projection
end;
subroutine: Sequential disease Pi[S(dj)]
    call Refined Database RDB
      for all patient pi
        sort diseases dj based on time interval
    end
subroutine: Refined database RDB
    call database DB
      get the value of Th
        for all patient pi
          calculate number of disease N(dj)
            if N(dj)<Th
               remove disease dj
          else
        add disease in RDB for the corresponding patient
 end
Subroutine: database DB
        P = {p1,…,pi} 1≤i≤k
            pi→{di,…,dj}, 1≤j≤1
              dj →(ts and te)
 end
```

1.8. Making of the Refined Database from the Original Database

D denotes temporal database with three attributes person ID, event type and time period. for instance some clinical records contains the attributes like patient Dim patient's disease and the time period of each disease, an instances of D is shown in **Table 1** the time period for each diseases are recorded using t_s and t_e which are the beginning time and ending time of the disease respectively.

1.9. Refined Database

The refined database is generated from the original database based on the threshold value. The count is calculated for each disease in the original database after that some of the diseases are removed from the original database which diseases having the count value below the threshold value. The below **Table 2** represents the refined database of the original database. Assume here the threshold values is 3.

1.10. Construction of Disease Sequences with Starting Time and Ending Time for Each Patient

Each disease having the starting time and the ending time, when one disease comes simultaneously another disease will also come. By adapting this we can conclude the next disease of the patient and prevent them from the new disease before affected the patient. The pictorial representation of the disease sequence based on the time is given in below **Fig. 1**.

From the figure, in first the patient101 affected by disease b when it finished get the disease d starts after some time disease a start and disease d finished before disease a finished. Their after the disease e comes before it finished a starts again after a stats e get finished. We plotted the sequences of diseases of each patient in the following **Table 3**; it consists of each patient id and their corresponding sequences of disease.

From the **Table 3** the patient id 101 consist of diseases a, b, d and e, the sequences of diseases are generated based on the time interval. For patient id 101 the disease b starts after that disease b end then disease d starts before disease a start and disease a ends before disease d ends.

Table 1. Original database D

Patient ID	Disease	Duration	Patient ID	Disease	Duration	Patient ID	Disease	Duration	Patient ID	Disease	Duration
101	b	(5, 8)	101	a	18, 26	103	c	07, 14	103	e	20, 26
101	c	12, 18	102	e	10, 15	103	b	11, 16	104	b	10, 17
101	d	10, 15	102	a	12, 16	103	a	15, 20	104	e	16, 22
101	a	09, 13	102	d	15, 19	103	f	23, 26	104	d	18, 26
101	e	16, 20	102	b	22, 26	103	h	22, 27	104	a	15, 20
101	g	19, 23	102	d	25, 29	103	g	25, 32			
101	f	22, 27	102	h	13, 19	103	e	18, 24			

Table 2. Represent the refined database of the original database

Patient ID	Disease	Duration	Patient ID	Disease	Duration	Patient ID	Disease	Duration
101	b	5, 80	102	a	12, 16	103	e	18, 24
101	d	10, 15	102	d	15, 19	103	e	20, 26
101	a	9, 13	102	b	22, 26	104	b	10, 17
101	e	16, 20	102	d	25, 29	104	e	16, 22
101	a	18, 26	103	b	11, 16	104	d	18, 26
102	e	10, 15	103	a	15, 20	104	a	15, 20

Table 3. Sequences of disease for each patient

Patient ID	Sequences of disease
101	$b_{sl} \rightarrow b_{el} \rightarrow a_{sl} \rightarrow d_{sl} \rightarrow a_{el} \rightarrow d_{el} \rightarrow e_{sl} \rightarrow a_{s2} \rightarrow e_{el} \rightarrow a_{e2}$
102	$e_{sl} \rightarrow a_{sl} \rightarrow e_{el} \leftrightarrow d_{sl} \rightarrow a_{el} \rightarrow d_{el} \rightarrow b_{sl} \rightarrow d_{s2} \rightarrow b_{el} \rightarrow d_{e2}$
103	$b_{sl} \rightarrow a_{sl} \rightarrow d_{el} \rightarrow e_{sl} \rightarrow a_{el} \leftrightarrow e_{s2} \rightarrow e_{el} \rightarrow e_{e2}$
104	$b_{sl} \rightarrow a_{sl} \rightarrow e_{sl} \rightarrow b_{el} \rightarrow d_{sl} \rightarrow a_{e2} \rightarrow e_{el} \rightarrow d_{el}$

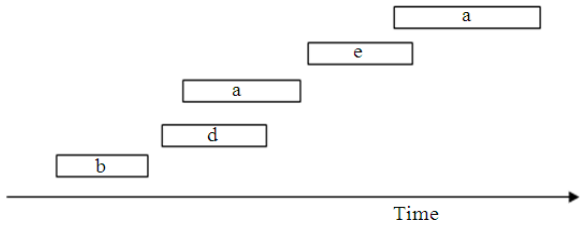

Fig. 1. Pictorial representation of the disease sequences of patient id 101

Then disease e starts before e ends disease a start again and a ends. By using this starting point and ending point of the diseases we plot the sequences of the diseases for the other patients based on the time interval.

Here after using the projection method we mine the sequences of diseases for whole database.

1.11. Mining the Sequences Based on the KPrefixspan Algorithm

Here, we present the TPrefixspan algorithm as KPrefixspan for mining the sequential patterns. The major advantage of this algorithm is to reduce the scanning time of the projected database and also the sequential pattern get accuracy. In this study we used the refined database for mine the sequential pattern since fewer amounts of data is used here. In this algorithm we first find the one length patterns from the refined database. The **Table 4** shows the one length pattern of KPrefixspan algorithm. From the refined database here we find the one length the patterns for a_{sl}, a_{e1}, b_{s1}, b_{e1}.

In order to find the sequential patterns we use the refined database, here we use the disease a_{sl} for sequential pattern. Mark the disease as_l in the refined database in all patients. Scan after the disease a_{sl} of each patient and place it into the projected database like $d_{s1} \rightarrow a_{e1} \rightarrow d_{e1} \rightarrow e_{s1} \rightarrow a_{s2} \rightarrow e_{e1} \rightarrow a_{e2}$, $e_{e1} \rightarrow d_{s1} \rightarrow a_{e1} \rightarrow d_{e1} \rightarrow b_{s1} \rightarrow d_{s2} \rightarrow b_{e1} \rightarrow d_{e2}$, $b_{e1} \rightarrow e_{s1} \rightarrow a_{e1} \leftrightarrow e_{s2} \rightarrow e_{e1} \rightarrow e_{e2}$ and $a_{s1} \rightarrow e_{s1} \rightarrow b_{e1} \rightarrow d_{e1} \rightarrow a_{e1} \rightarrow e_{e1} \rightarrow de_1$. Count the each disease of the projected database, the count value for each disease are $a_{e1} = 4$, $d_{e1} = 3$, $b_{s1} = 1$, $a_{s2} = 1$, $a_{e2} = 1$, $d_{e2} = 1$, $e_{s1} = 3$, $b_{e1} = 3$, $e_{e1} = 4$, $d_{s1} = 3$. Here the threshold value is 3; remove the diseases from the projected database which disease having the count value below threshold here the removed disease is b_{s1}, a_{s2}, a_{e2}, d_{e2} since those diseases are not supported for threshold value. The balanced diseases are selected for the two length sequences like a_{e1}, d_{e1}, e_{s1}, b_{e1}, e_{e1}, d_{s1}. Likewise we need to proceeds the same procedure for the other one length patterns. The example one length pattern is given in the following **Table 4**.

The two length sequences are generated from the above **Table 4**, by projecting the two length sequence we

can made the three length sequences in the **Table 5** by using the **Table 4**.

Table 4. Projected database, sequence of disease for the one length prefix

Prefix	Projected database	Counts	Sequential disease
a_{s1}	$\rightarrow d_{s1} \rightarrow a_{e1} \rightarrow d_{e1} \rightarrow e_{s1} \rightarrow a_{s2} \rightarrow e_{e1} \rightarrow a_{e2}$	$a_{e1} = 4$ $d_{e1} = 3$	$a_{s1} \rightarrow a_{e1}$, $a_{s1} \rightarrow d_{s1}$
	$\rightarrow e_{e1} \leftrightarrow d_{s1} \rightarrow a_{e1} \rightarrow d_{e1} \rightarrow b_{s1} \rightarrow d_{s2} \rightarrow b_{e1} \rightarrow d_{e2}$	$b_{s1} = 1 e_{s1} = 3$	$a_{s1} \rightarrow d_{e1}$, $a_{s1} \rightarrow e_{e1}$
	$\rightarrow b_{e1} \rightarrow e_{s1} \rightarrow a_{e1} \leftrightarrow e_{s2} \rightarrow e_{s2} \rightarrow e_{e1} \rightarrow e_{e2}$	$b_{e1} = 3 e_{e1} = 4$	$a_{s1} \rightarrow e_{s1}$, $a_{s1} \rightarrow b_{e1}$
	$e_{s1} \rightarrow b_{e1} \rightarrow d_{s1} \rightarrow a_{e1} \rightarrow e_{e1} \rightarrow d_{e1}$	$d_{s1} = 3$	
a_{e1}	$\rightarrow d_{e1} \rightarrow e_{s1} \rightarrow a_{s2} \rightarrow e_{e1} \rightarrow a_{e2}$	$d_{e1} = 3$	$a_{e1} \rightarrow d_{e1}$
	$\rightarrow d_{e1} \rightarrow b_{s1} \rightarrow d_{s2} \rightarrow b_{e1} \rightarrow d_{e2}$	$b_{s1} = 1$ $e_{s1} = 1$	$a_{e1} \rightarrow e_{e1}$
	$\leftrightarrow e_{s2} \rightarrow e_{e1} \rightarrow e_{e2}$	$b_{e1} = 1$ $e_{e1} = 3$	
	$e_{e1} \rightarrow b_{e1}$	$d_{s1} = 0$	
b_{s1}	$\rightarrow b_{e1} \rightarrow a_{s1} \rightarrow d_{s1} \rightarrow a_{e1} \rightarrow d_{e1} \rightarrow e_{s1} \rightarrow a_{s2} \rightarrow e_{e1} \rightarrow a_{e2}$	$a_{e1} = 3$ $d_{e1} = 2$	$b_{s1} \rightarrow b_{e1}$, $b_{s1} \rightarrow e_{e1}$
	$\rightarrow d_{s2} \rightarrow b_{e1} \rightarrow d_{e2}$	$a_{s1} = 3$ $e_{s1} = 3$	$b_{s1} \rightarrow a_{e1}$
	$\rightarrow a_{s1} \rightarrow b_{e1} \rightarrow e_{s1} \rightarrow a_{e1} \leftrightarrow e_{s2} \rightarrow e_{e1} \rightarrow e_{e2}$	$b_{e1} = 4$ $e_{e1} = 3$	$b_{s1} \rightarrow a_{s1}$
	$\rightarrow a_{s1} \rightarrow e_{s1} \rightarrow b_{e1} \rightarrow d_{s1} \rightarrow a_{e1} \rightarrow e_{e1} \rightarrow d_{e1}$	$d_{s1} = 2$	$b_{s1} \rightarrow e_{s1}$
b_{e1}	$\rightarrow a_{s1} \rightarrow d_{s1} \rightarrow a_{e1} \rightarrow d_{e1} \rightarrow e_{s1} \rightarrow a_{s2} \rightarrow e_{e1} \rightarrow a_{e2}$	$a_{e1} = 3$ $d_{e1} = 2$	$b_{e1} \rightarrow e_{s1}$
	$\rightarrow d_{e2}$	$a_{s1} = 1$ $e_{s1} = 2$	$b_{e1} \rightarrow a_{e1}$
	$\rightarrow d_{e2} \rightarrow a_{e1} \leftrightarrow e_{s2} \rightarrow e_{e1} \rightarrow e_{e2}$	$e_{e1} = 3$	
	$\rightarrow d_{s1} \rightarrow a_{e1} \rightarrow e_{e1} \rightarrow d_{e1}$	$d_{s1} = 2$	

Table 5. Projected database, sequence of disease for the two length prefix

Prefix	Projected database	Counts	Sequential disease
$a_{s1} \rightarrow a_{e1}$	$\rightarrow d_{e1} \rightarrow e_{s1} \rightarrow e_{e1}$	$d_{e1} = 3$	$a_{s1} \rightarrow a_{e1} \rightarrow d_{e1}$
	$\rightarrow d_{e1} \rightarrow b_{e1}$	$e_{e1} = 3$	$a_{s1} \rightarrow a_{e1} \rightarrow e_{e1}$
	$\rightarrow e_{e1}$		
	$\rightarrow e_{e1} \rightarrow d_{e1}$		
$a_{s1} \rightarrow d_{s1}$	$\rightarrow a_{e1} \rightarrow d_{e1} \rightarrow e_{s1} \rightarrow e_{e1}$	$d_{e1} = 3$	$a_{s1} \rightarrow d_{s1} \rightarrow d_{e1}$
	$\rightarrow a_{e1} \rightarrow d_{e1} \rightarrow b_{e1}$	$a_{e1} = 3$	$a_{s1} \rightarrow d_{s1} \rightarrow a_{e1}$

	$\rightarrow a_{e1} \rightarrow e_{e1} \rightarrow d_{e1}$		
$a_{s1} \rightarrow d_{e1}$	$\rightarrow e_{s1} \rightarrow e_{e1}$	----	----
	$\rightarrow b_{e1}$		

For the prefix a_{s1} the possible two length sequences are $a_{s1} \rightarrow a_{e1}$, $a_{s1} \rightarrow d_{s1}$, $a_{s1} \rightarrow d_{e1}$, $a_{s1} \rightarrow e_{e1}$, $a_{s1} \rightarrow e_{s1} \rightarrow a_{e1} \rightarrow be_1$. For example $a_{s1} \rightarrow a_{e1}$ is a two length sequence, the scanning is start after a_{e1} in the first projected database, here the scanning will be reduced massively since more amount of data's are removed from the first projected database due to not supporting of the threshold value. the available two length patterns from the one length patterns are given below $a_{s1} \rightarrow a_{e1}$, $a_{s1} \rightarrow d_{s1}$, $a_{s1} \rightarrow d_{e1}$, $a_{s1} \rightarrow e_{e1}$, $a_{s1} \rightarrow e_{s1}$, $a_{s1} \rightarrow b_{e1}$, $a_{e1} \rightarrow d_{e1}$, $a_{e1} \rightarrow e_{e1}$, $b_{s1} \rightarrow b_{e1}$, $b_{s1} \rightarrow e_{e1}$, $b_{s1} \rightarrow a_{e1}$, $b_{s1} \rightarrow a_{s1}$, $b_{s1} \rightarrow e_{s1}$, $b_{e1} \rightarrow e_{s1}$, $b_{e1} \rightarrow a_{e1}$, from this available two length patterns, the following **Table 5** describes the finding of three length patterns for $a_{s1} \rightarrow a_{e1}$, $a_{s1} \rightarrow d_{s1} \rightarrow a_{s1}$, $a_{s1} \rightarrow d_{e1}$. From the **Table 5**, $a_{s1} \rightarrow a_{e1}$ is one of the two length pattern that having two threshold support patterns like d_{e1}, e_{e1}. Here, also the unsupported threshold values are removed from the projected database for computing the three length patterns.

The following **Table 6** shows projected database and sequential diseases for the three length pattern. The available three length patterns from the **Table 5** are $a_{s1} \rightarrow a_{e1} \rightarrow d_{e1}$, $a_{s1} \rightarrow a_{e1} \rightarrow e_{e1}$, $a_{s1} \rightarrow d_{s1} \rightarrow d_{e1}$, $a_{s1} \rightarrow d_{s1} \rightarrow a_{e1}$. By using the above three length patterns find the possible patterns in the following.

From the **Table 6** we can find the four length pattern. The above **Table** the four length patterns are find for the following three length patterns $a_{s1} \rightarrow a_{e1} \rightarrow d_{e1}$, $a_{s1} \rightarrow a_{e1} \rightarrow e_{e1}$, $a_{s1} \rightarrow d_{s1} \rightarrow d_{e1}$, $a_{s1} \rightarrow d_{s1} \rightarrow a_{e1}$ from these patterns the four length pattern is derived from the only one three length pattern that is $a_{s1} \rightarrow d_{s1} \rightarrow a_{e1}$ since only this three length pattern having the support for threshold value. The four length pattern is given below. Furthermore patterns are not available by seeing this four length patterns. $a_{s1} \rightarrow d_{s1} \rightarrow a_{e1} \rightarrow de_1$ this four length patterns are describes disease d starts after the disease a start and the disease a finished before the

disease finished. By using these patterns we can conclude

Table 6. Projected database, sequence of disease for the three length prefix

Prefix	Projected database	Counts	Sequential disease
$a_{s1} \rightarrow a_{e1} \rightarrow d_{e1}$	$\rightarrow e_{s1} \rightarrow e_{e1}$	---	---
	$\rightarrow b_{e1}$		

$a_{s1} \rightarrow a_{e1} \rightarrow e_{e1}$	---	---	---

	---		
	d_{e1}		
$a_{s1} \rightarrow d_{s1} \rightarrow d_{e1}$	---	---	----

$a_{s1} \rightarrow d_{s1} \rightarrow a_{e1}$	d_{e1}		$a_{s1} \rightarrow d_{s1} \rightarrow a_{e1} \rightarrow d_{e1}$
	d_{e1}	$d_{e1} = 3$	
	---		
	d_{e1}		

1.12. Experimental Results

The experimental results of the proposed approach devising an efficient approach to discover time interval-based sequential patterns from temporal database. The experimental and the comparative analysis is done with the aid of the existing literature work, which was proposed by Wu and Chen (2007).

1.13. Experimental Design

The proposed techniques for effectual mining of walking path sequences are programmed using Java (jdk 1.6). The experimentation has been carried out on a 2.9 GHz, i5 PC machine with 4 GB main memory running a 64-bit version of Windows 7. The performance of the proposed techniques has been evaluated using the synthetic datasets. The major application of this approach is to find the sequential diseases of the particular area.

1.14. Performance Evaluation and Comparative Analysis

The performance of the proposed sequential pattern mining algorithm from the sequential dataset is evaluated by means of four evaluation measures. They are: (1) Generated number of patterns-the maximum number of random patterns generated based upon the given threshold value and number of input data records, (2) Running time-the time taken to execute the computer program and it typically grows with the input size and the threshold value, (3) Memory usage- the memory utilized by the current jobs present in the particular system it based on the number of input data records and sequential of disease for the patient. value of the threshold, (4) Length of the sequential patterns-the length of the sequential pattern is depends on the threshold value and number of input data records. We have analyzed and compared our proposed approach with the well known existing work (Wu and Chen, 2007) by applying the using the KPrefixspan mining algorithm.

1.15. Evaluation of Measures Based on the Number of Input Value

We have done the analysis and plotted as a graph by computing the generated number of sequences, execution time and the memory usage length of sequential pattern with different minimum support threshold based on the number of input data and here the threshold value is constant value as 10

The **Fig. 2** describes the running time of KPrefixspan algorithm and TPrefixspan algorithm. The running time of the KPrefixspan algorithm is randomly changed at different level of the input data but in the TPrefixspan algorithm is increased based on the input level. The highest running time of KPrefixspan algorithm is 32047 ms for 400 numbers of input values at the same time the lowest running time is 307 ms for 100 numbers of input values.

The **Fig. 3** describes, our KPrefixspan algorithm is better than the TPrefixspan algorithm when we consider the memory usage while executing. Our KPrefixspan algorithm removes the unsupported events at every time since the storage get decrease but the TPrefixspan is keep the unwanted events since it need more memory space.

The **Fig. 4** describes how much number of patterns is got by our KPrefixspan algorithm and TPrefixspan algorithm. While comparing the two algorithms, our algorithm has less number of patterns because the some unwanted events are removed from the database at every projection based on the user defined threshold value.. From this we can conclude one thing, by use of KPrefixspan algorithm when the number of input data is increase the accuracy of the output also get increase.

While we seeing the **Fig. 5** the number of patterns of the TPrefixspan algorithm is decreased gradually when the threshold value is increased while comparing the result of KPrefixspan with TPrefixspan, the length of pattern of KPrefixspan get less.

1.16. Evaluation of Measures Based on the Threshold Value

We have done the analysis and plotted as a graph by computing the generated number of sequences, execution time and the memory usage length of sequential pattern with different minimum support

threshold based on the threshold value and here the number of input values are constant as 5000.

The **Fig. 6** illustrates the running time of our KPrefixspan algorithm is less when compared with the TPrefixspan algorithm. When the threshold value increase automatically the execution time get reduce since the unwanted patterns are increase when the threshold value increase then more number of unwanted patterns are removed by each projection consequently the results are achieved in a short period.

From the **Fig. 7** the memory usage of our proposed KPrefixspan algorithm is less than the TPrefixspan algorithm because of every time the some of the events are removed from the database since the need of storing the events become reduces when the number of threshold value is increased.

The **Fig. 8** illustrates our algorithm gets less number of patterns at the same time both of the algorithms are get more number of patterns for the threshold value 200. The numbers of patterns are decreased gradually for the KPrefixspan and TPrefixspan algorithm from the threshold value 300 to 500.

While we seeing the **Fig. 9**, the number of patterns of the KPrefixspan algorithm is decreased gradually when the threshold value is increased while comparing the result of KPrefixspan with TPrefixspan, the length of pattern of TPrefixspan get less.

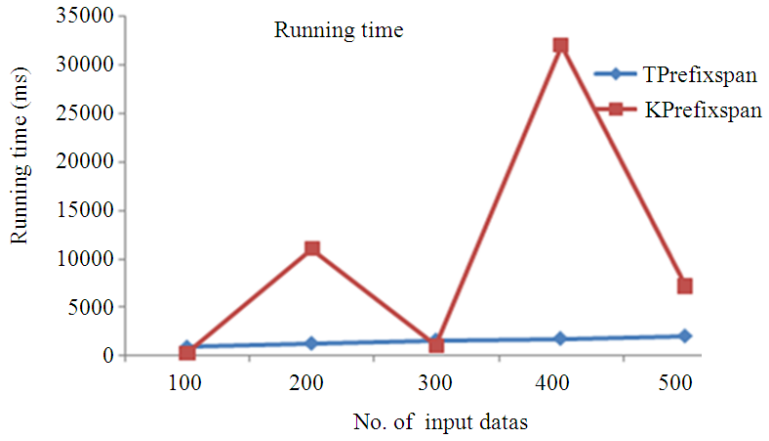

Fig. 2. Illustrate the running time based on the number of input data

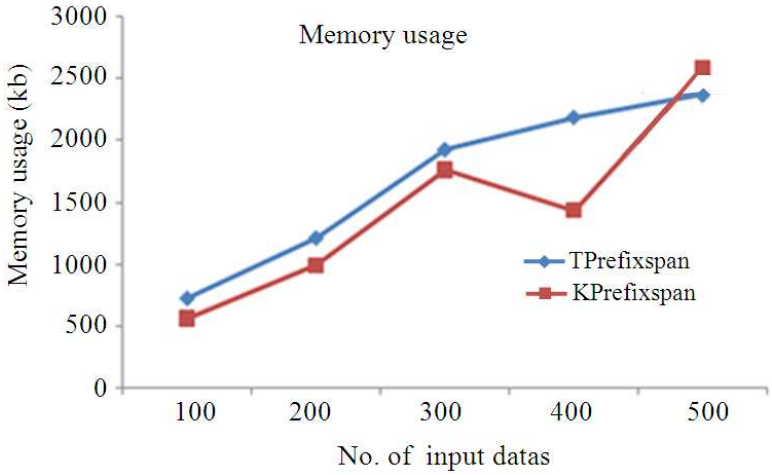

Fig. 3. Illustrate the memory usage based on the number of input data

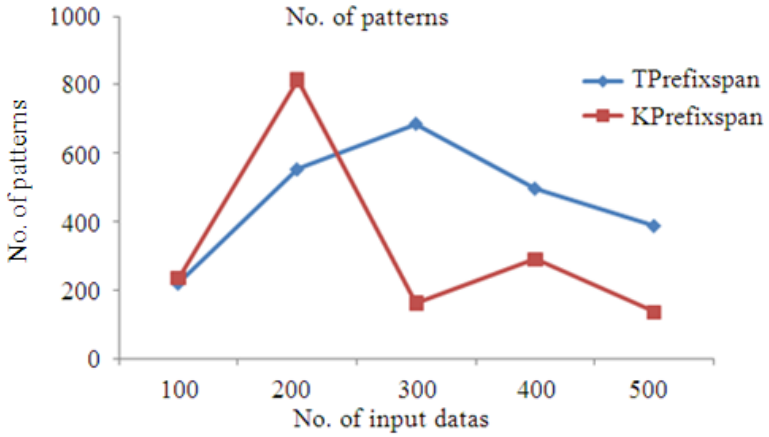

Fig. 4. Illustrate the number of patterns based on the number of input data

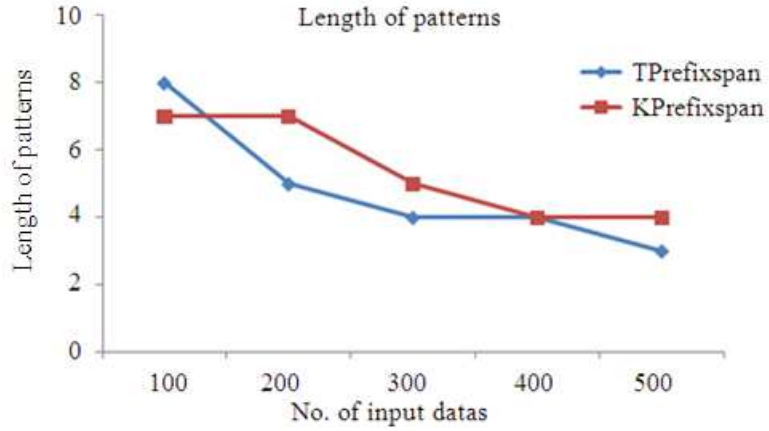

Fig. 5. Illustrate the length of sequence based on the number of input data

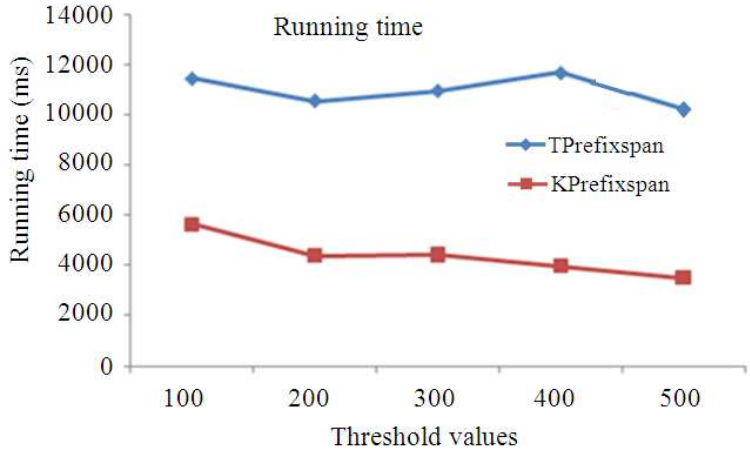

Fig. 6. Illustrate the running time based on the various threshold values

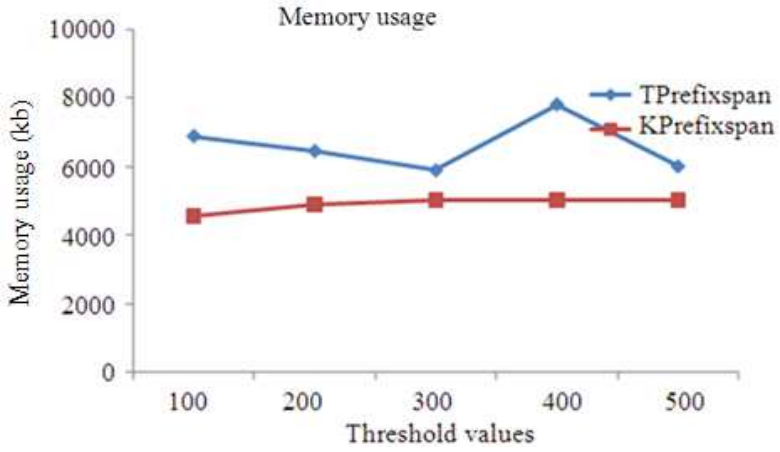

Fig. 7. Illustrate the memory usage based on the various threshold values

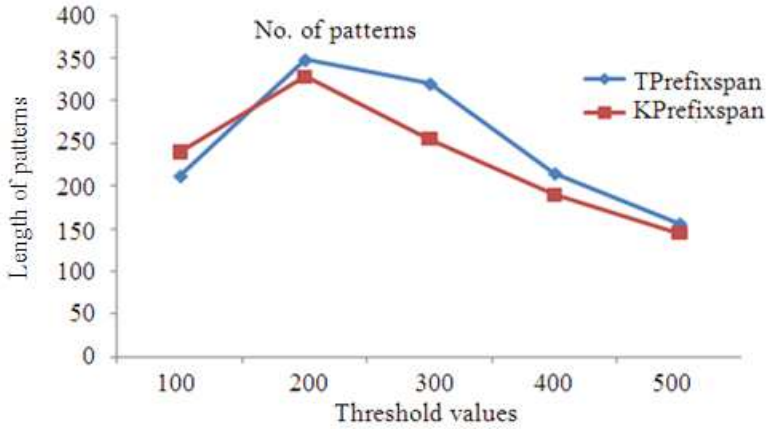

Fig. 8. Illustrate the number of patterns based on the various threshold values

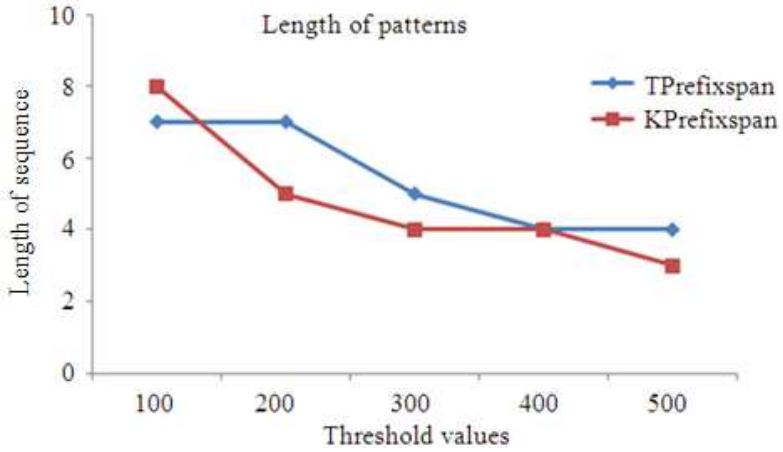

Fig. 9. Illustrate the length of sequences based on the various threshold values

2. CONCLUSION

In this study, we have presented the devising an efficient approach to discover time interval based sequential patterns from temporal database. In order to find the sequential patterns we have presented the efficient sequential pattern mining algorithm that is an improved version of the TPrefixspan algorithm. At first the databases are converted into the refined database by eliminating the unsupported threshold values from the original database. After that the input data's are converted into interval based format that will be the input of the proposed approach, the formatted patterns are sorted based on the time interval. Consequently the proposed approach is done and patterns are removed which patterns are having the values of below threshold value. At last we got the patterns in a sequential based on the time interval. Finally the experimentation has carried out on the synthetic datasets, from the experimental results we reduced the running time almost 60% and also reduce the memory usage almost 25% when compared to the existing TPrefixspan algorithm.

3. REFERENCES

Antunes, C.M. and A.L. Oliveira, 2001. Temporal data mining: An overview. Lecture Notes Comput. Sci.

Chen, Y.C., J.C. Jiang, W.C. Peng and S.Y. Lee, 2010. An efficient algorithm for mining time interval-based patterns in large database. Proceedings of the 19th ACM International Conference on Information and Knowledge Management, Oct. 26-30, ACM Press, Toronto, ON, Canada, pp: 49-58. DOI: 10.1145/1871437.1871448

Guyet, T. and R. Quiniou, 2011. Extracting temporal patterns from interval-based sequences. Proceedings of the 22nd International Joint Conference on Artificial Intelligence, (CAI' 11), ACM Press, pp: 1306-1311. DOI: 10.5591/978-1-57735-516-8/IJCAI11-221

Han, J. and M. Kamber, 2001. Data Mining: Concepts and Techniques. 7th Edn., Morgan Kaufmann, San Francisco, ISBN-10: 1558604898, pp: 550.

Jensen, R., 2000. Agricultural volatility and investments in children. Am. Econ. Rev., 90: 399-404.

Laxman, S. and P.S. Sastry, 2006. A survey of temporal data mining. Sadhana, 31: 173-198. DOI: 10.1007/BF02719780

Liu, H. and L. Yu, 2005. Toward integrating feature selection algorithms for classification and clustering. IEEE Trans. Knowl. Data Eng., 17: 491-502. DOI: 10.1109/TKDE.2005.66

Masseglia, F., P. Poncelet and M. Teisseire, 2003. Incremental mining of sequential patterns in large databases. Data Knowl. Eng., 46: 97-121. DOI: 10.1016/S0169-023X(02)00209-4

Patel, D., W. Hsu and M.L. Lee, 2008. Mining relationships among interval-based events for classification. Proceedings of the ACM SIGMOD International Conference on Management of Data, Jun. 9-12, ACM Press, Vancouver, Canada, pp: 393-404. DOI: 10.1145/1376616.1376658

Sobh, T., 2007. Innovations and Advanced Techniques in Computer and Information Sciences and Engineering. 1st Edn., Springer, Dordrecht, ISBN-10: 1402062680, pp: 576.

Zhao, Q. and S.S. Bhowmick, 2003. Association rule mining: A survey. Nanyang Technological University, Singapore.

Moskovitch, R. and Y. Shahar, 2005. Temporal data mining based on temporal abstractions. Proceedings of International Conference on Data Mining, Workshop on Temporal Data Mining: Algorithms, Theory and Application, (ATA' 05), Houston, Texas, USA., pp: 113-115.

Wu, S.Y. and Y.L. Chen, 2007. Mining nonambiguous temporal patterns for interval-based events. IEEE Trans. Knowl. Data Eng., 19: 742-758. DOI: 10.1109/TKDE.2007.190613

Zhu, C., X. Zhang, J. Sun and B. Huang, 2009. Algorithm for mining sequential pattern in time series data. Proceedings of the WRI International Conference on Communications and Mobile Computing, Jan. 6-8, IEEE Xplore Press, Yunnan, pp: 258-262. DOI: 10.1109/CMC.2009.208