ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Comparative Study of Time Interval Weighted Sequential pattern Mining and Prefixspan Algorithm

A. Sirisha, Suresh Pabboju, G. Narsimha

Assistant Professor, Dept. of IT, Chaitanya Bharthi Institute of Technology, Hyderabad, India

Professor, Dept. of IT, Chaitanya Bharthi Institute of Technology, Hyderabad, India

Associate Professor, Dept. of CSE, JNTUH, College of Engineering, Jagtial, Karimnagar, India

ABSTRACT: Sequence pattern mining is one of the essential data mining tasks with broad applications. Many sequence mining algorithms have been developed to find a set of frequent sub-sequences in a sequence database given the support threshold. The main drawbacks with algorithms is they generate huge number of sequential patterns when the support threshold is low and all the sequence patterns are treated uniformly while real sequential patterns have different importance. To address these drawbacks the weighted sequential pattern mining was proposed, which aims to find more interesting sequential patterns, considering the different significance of each data element in a sequence database. In weighted sequential pattern mining, only the generation order of data elements is considered to find sequential patterns However, their generation times and time-intervals are also important in real world application domains. This paper presents a new approach for finding time-interval weighted sequential (TIWS) patterns in a sequence database. In the proposed approach, the weight of each sequence in a sequence database is first obtained from the time-intervals of elements in the sequence, and subsequently TIWS patterns mined considering the weight. Experimental results show that TIWS pattern mining is efficient and effective in generating more interesting patterns.

KEYWORDS: Sequential pattern mining; Time interval weight; Time interval weighted support; TIWS pattern.

I. INTRODUCTION

Sequential pattern mining is one of the most important data mining techniques, which has drawn a great number of researches to pursue it, because of its numerous applications including usage in customer purchase behavior analysis, webpage access pattern detection, disease treatment pattern analysis, DNA sequence analysis and stock sequence analysis. The sequential pattern mining problem was first introduced by Srikanth and Agrawal [1]. Sequential pattern mining finds all frequent sub sequences whose occurrence frequency is no less than the given threshold frequency from a sequence database. Many Sequential pattern mining algorithms have been extensively developed due to its huge applications but the basic approaches do not reflect the characteristic of real datasets i.e they consider the sequential patterns and items in a sequential pattern uniformly. However, they have different importance in real world applications. For example, when finding the traversal patterns in the World Wide Web, different pages can have different importance. In biomedical and DNA data analysis, some genes are more important than others in causing a particular disease and some genes are more effective than others in fighting diseases. With these observations weighted sequential pattern mining [9] has been proposed. In this approach weights are pre assigned to the items according to their priority or importance, A weight range is used to restrict the weight values of items and the values are normalized. During mining the mean weight is used to prune the weighted infrequent patterns. Although conventional sequential patterns can reveal the order of items, they do not include time intervals between successive items. In [4] and [5], several sequential pattern mining algorithms have been presented which consider a time-interval between two successive items in a sequential pattern. However, they simply consider a time-interval between two successive data elements as an item, and thus they are unable to get weighed sequential patterns considering different weights of sequences in a sequence database. If the importance of sequences in a sequence database is differentiated based on the time-intervals in the sequences, more interesting sequential patterns can be found.

For example, in a sequence database gathered from a computer store, let us say the following two sequences were

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

found:
Customer_A: Having bought a CD burner, returns to buy a laser printer after 1 month and then a scanner after 1 month.
Customer_B : Having bought a CD burner, returns to buy a printer after 6 months and then a scanner after 3 months.

The above sequences consist of the same items and the orders of items are the same in both customers, but the time-intervals between the items are different. Therefore, they may appear to be the same if only the order of items is considered but they are completely different sequences.In the above example, the sequence by a Customer_A can be considered more important than the sequence by a Customer_B, since the former has relatively smaller time-intervals than the latter. Accordingly, for a sequence in a sequence database, its importance, i.e. its weight, can be computed from the time-intervals in the sequence. This paper proposes a new approach for finding weighted sequential patterns with a time-interval weight. The effectiveness of this approach is analysed in terms of number of resulting sequential pattern and processing time. And the experiment results show that this approach is efficient and helpful in finding more interesting sequential patterns.

The rest of this paper is organized as follows: Section 2 gives a brief summary of related work, and a problem definition is given in Section 3. In Section 4, a new approach for finding time interval weighted sequential patterns is described. Section 5 summarizes a series of experimental results, and finally conclusions are described in Section 6.

II. RELATED WORK

Many algorithms have been proposed for mining sequential patterns. And most of the basic and earlier algorithms are based on the Apriori property that any super-pattern of a non frequent pattern cannot be frequent. The apriori-based methods have drawbacks of generating large sets of candidate sequences and performing multiple scans of a data set to get a resulting set of frequent sequences. To overcome these drawbacks a series of data projection based algorithms FreeSpan[2] and PrefixSpan[3] were proposed. PrefixSpan is one of the pattern-growth approaches that recursively projects a sequence database into a set of smaller projected sequence databases and grows sequential patterns in each projected database by exploring only the locally frequent fragments. However, the PrefixSpan method also partially requires multiple scans of data sets, i.e., projected data sets. Among the algorithms for sequential pattern mining, SPADE [6] and PrefixSpan are more efficient than others in terms of processing time. SPADE is one of the vertical-format based algorithms and uses equivalence classes in the mining process. All the sequential pattern mining algorithms [6, 7, 8, 9] suggested so far have given same importance to the sequences and the elements in a sequence. However, it is important to distinguish important sequences from a large number of sequence patterns. To solve this problem Wspan[10] algorithm is proposed but in this algorithm the weights of the items are set manually based on the domain knowledge. Time-interval and gap information between items in a sequence can been used to find interesting sequential patterns, and there have been several studies on mining sequential patterns considering them. Pei et al. [12] and Ji et al. [11] have proposed constrained sequential pattern mining methods, which use time-interval and gap information as a constraint. In these methods, they are used only to confine the mining result of sequential patterns, but they do not give a mining result of weighted sequential patterns. Chen et al. have proposed sequential pattern mining algorithms, considering time-interval information between two successive items in a sequence [4, 5]. However, the algorithms just consider time-interval information between two successive items as an item, so that they cannot support to get weighted sequential patterns based on different weights of sequences.

III. PROBLEM DEFINITION

The problem of sequential pattern mining is to find the complete set of sequential patterns in the sequence database with a support constraint. The most general form of the sequence mining problem can be stated as follows:

Let $I = \{i_1, i_2,..., i_m\}$ be a set of m-distinct items. An itemset s_j is a set of items, $s_j \subseteq I$.

A sequence S is an ordered list of itemsets denoted by $S=<s_1, s_2, \ldots, s_n>$ where s_j is an itemset, also called an element of the sequence. and its time stamp list $TS(S)=<t_1; t_2; \ldots ; t_n>$ is an ordered list of corresponding time stamps of the itemsets, which stand for the time when they occur, where t_j $(1 \le j \le n)$ is the time stamp of s_j and $t_{j-1} \le t_j$ $(2 \le j \le n)$. Each itemset in a sequence represents a set of events occurring at the same timestamp. We assume that the items in each itemset are sorted in certain order. An item can occur at most one time in an element of a sequence but it can occur multiple times in different elements of a sequence. The sequence with a total of k items is referred to as k-sequence.

A sequence $S = <s_1, s_2,\ldots, s_m>$ is a *subsequence* of another sequence $T = <t_1, t_2,\ldots, t_n>$, denoted by $S \subseteq T$ if and only

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

if there exists set of integers i_1, i_2,…, i_m, such that $1 \leq i_1 < i_2 < \ldots < i_m \leq n$ and $s_1 \subseteq t_{i1}$, $s_2 \subseteq t_{i2}, \ldots, s_m \subseteq t_{im}$. A sequence database is a set of tuples <sid, S> where sid is a sequence identifier and S is a sequence.

For a sequence database, a tuple <sid,S> is said to contain a sequence T if T is a subsequence of S. The support of a sequence T in a sequence database is the ratio of number of tuples in the database containing T to the size of the database. If the support of a sequence *S* satisfies a pre-specified *min_sup* threshold, *S* is a *frequent* sequential pattern.

TABLE 1
A SEQUENCE DATABASE

Sid	Sequence	Timestamp list
10	< (*A*) (*A B C D E F*) (*A E G*)>	<0 1 2>
20	< (*A D*)(*A B C D F*) (*B C E F*)>	<1 2 3>
30	<*A* (*A B C E F H*) (*A D*)>	<1 3 5>
40	<*A* (*A C*) (*ABD*) *B C*>	< 2 3 4 5>

Example 1. Table 1 shows a sequence database SDB with four sequences, and their sids are 10, 20, 30, and 40, respectively. Assume that *min_sup* is 0.5. The *SDB* has 8 unique items, and 4 input sequences. The *size* of SDB is 4 and the *length* of *the* sequence < *(A)* (*A B C D E F*) (*A E G*) > is 10. The Sequence < (*B C E F*) (A)> is a sub sequence of <(*A*) (*A B C D E F*) (*A E G*)> since (*B C E F*) ⊆(*A B C D E F*), (A) ⊆ (*A E G*).

The TIWS approach proposed in this paper considers the weight of a sequence to find the count of a sequence and uses it to find the support of a sequence. The weighted count of a sequence A in a sequence database SDB is the sum of weights of sequences in SDB containing A. Likewise, its weighted support is the ratio of its weighted count over the sum of the weights of all sequences in SDB. Given a support threshold min_Support ($0 < \text{min\_Support} \leq 1$) and a sequence database SDB, a sequence A is called a time-interval weighted sequential pattern if the weighted support of A is no less than the support threshold. Accordingly, time-interval weighted sequential pattern mining is the problem of finding the complete set of all time-interval weighted sequential patterns whose weighted supports are no less than the threshold support.

IV. MINING TIME-TERVAL WEIGHTED SEQUENTIAL PATTERNS

In this section, we propose a new approach to find time interval weighted sequential patterns. In our approach first we obtain the weight of each sequence from the time-intervals in the sequence, later we use these weight values to find the weighted support of sequences and then we define time interval weighted sequential pattern. Finally, an algorithm for detecting time interval weighted sequential patterns is presented.

A. Time Interval Weighted sequential pattern

A sequence in the sequence database consists of itemsets and their corresponding time stamps. This section describes how the time-intervals of a sequence are found from the time stamps of itemsets in the sequence which are used to get the time-interval weight of the sequence. For a sequence in a sequence database where the sequence consists of n itemsets, there exist n*(n-1)/2 pairs of itemsets in the sequence, and the time-interval between the itemsets of each pair is defined as in Definition 1.

Definition 1 *Time-interval between a pair of itemsets*
For a sequence $S=<s_1, s_2, \ldots, s_n>$) and its time stamp list $TS(S)=<t_1; t_2; \ldots; t_n>$ the time-interval between two itemsets s_i and s_j ($1 \leq i < j \leq n$) in the sequence, i.e., TI_{ij}, is defined as follows: $TI_{ij} = t_j \_ t_i$.

The time-interval between a pair of itemsets is a positive value with no limitation. Therefore, to fairly enumerate the time-intervals of different pairs of itemsets in a sequence database, they need to be normalized. For this purpose, the time-interval weight of the pair is found for each pair of itemsets in a sequence based on its time-interval, and defined as in Definition 2.

Definition 2 *Time-interval weight*

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Let u(u > 0) be the size of unit time and δ (0 < δ < 1) be a base number to determine the amount of weight reduction per unit time u, for a sequence S=<s_1, s_2, …,s_n>) and its time stamp list TS(S)=<t_1; t_2; . . . ; t_n> the time interval weight of a time-interval TI_{ij} between two itemsets s_i and s_j (1 ≤ i < j ≤ n), i.e., $w_g(TI_{ij})$ and $w_l(TI_{ij})$ are defined, respectively, as follows:

(i) General scale weighting: $w_g(TI_{ij}) = \delta^{TIij/u} = \delta^{tj-ti/u}$

(ii) Log scale weighting: $w_l(TI_{ij}) = \delta^{\log2[1+TIij/u]}$

The time-interval weight of a sequence is computed from the time-intervals of pairs of itemsets in the sequence. In this process, for two different pairs of itemsets in the same sequence, a contribution of each pair to the sequence may be different even though they have the same time-interval weight. This is because the diversity of the itemsets in terms of the length of an itemset. That is, among the itemsets in a sequence, a large-sized itemset may contribute more to the sequence than a small-sized one. Based on this observation, for a pair of itemsets in a sequence, its strength of contribution to the sequence is defined as in Definition 3.

Definition 3 *Strength of a pair of itemsets*
For a sequence S=<s_1, s_2, …,s_n>) , the strength of a pair of two itemsets s_i and s_j (1 ≤ i < j ≤ n) in the sequence, is defined as i.e., ST_{ij}= len(s_i) * len(s_j) where len(s_i) is the number of items in s_i.

Definition 4 *Time-interval weight of a sequence*
For a sequence S=<s_1, s_2, …,s_n>) and its time stamp list TS(S)=<t_1; t_2; . . . ; t_n> the time-interval weight of the sequence i.e., W(S), is defined as follows:
$W(S) = \frac{1}{N}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} w(TI_{ij}) * ST_{ij}$ where $N = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} ST_{ij}$

Using the time-interval weight, the TIW-support of a sequence in a sequence database is defined as in Definition 5.

Definition 5 *TIW-support of a sequence*
For a sequence database SDB, the TIW-support of a sequence S in the sequence database i.e TIW-Sup(S) is defined as follows:
TIW-Sup(S) = $\frac{\sum_{X:S\subseteq X,\ X\in SDB} W(X)}{\sum_{X:\ X\in SDB} W(X)}$

Definition 6 *Time-interval weighted sequential pattern (TIWS pattern)*
Given a support threshold min_Sup (0 < min_Sup ≤ 1), a sequence S is a time-interval weighted sequential pattern if TIW-Sup(S) is no less than the threshold, i.e., TIW-Sup(S) ≥ min_Sup.

Let A and B be sequences in a sequence database SDB and B is a super-sequence of A, i.e., A⊆(B., then if TIW-Supp(A) is less than a predefined support threshold, TIW-Supp(B) is also less than the threshold. This property of TIW-support can be used to prune the exponential search space to find TIWS patterns in a large sequence database.

B. Description of the proposed method for Mining TIWS Pattern

In our approach, the sequence database is recursively projected into a set of smaller projected databases and time-interval weighted sequential patterns are grown in each projected database. Given a sequential pattern α in a sequence database, α-projected database (S|α) is the collection of suffixes of sequences in S with the prefix α.

TIWS Algorithm: Time Interval Weighted Sequential patterns mining from large sequence databases.
Input: A sequence database SDB, minimum support threshold min_sup and a time interval weighing function
Output: The complete set of time interval weighted sequential patterns.
Begin
1. Scan SDB once, For each sequence S call *Compute_TIWeight(S*) to get the time interval weight of a sequence

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

W(s)
2. Find each time-interval weighted frequent item β, satisfying the condition: TIW-Sup(β) ≥ min_sup
3. For each time-interval weighted frequent item β, in SDB, output β and Call Pspan (<β>, 1, S| β)
End

Procedure ***Compute_TIWeight(S)***
Parameters:
(1) Sequence S=<s_1, s_2, …,s_n>),
(2) Time stamp list TS(S)=<t_1; t_2; . . . ; t_n>
(3) W(TI) is the weighing function for a time interval TI
Begin
1. For each pair of elements s_i and s_j in the Sequence S, find the time interval TI_{ijj} and the strength of the pair ST_{ij}
2. Compute the time-interval weight of the sequence i.e., W(S) as

$$W(S) = \frac{1}{N}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} w(TI_{ij}) * ST_{ij} \quad \text{where } N = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} ST_{ij}$$

End

Procedure ***Pspan (α, L, S|α)***
Parameters:
(1) α is a weighted sequential pattern,
(2) L is the length of α
(3) S|α is the α-projected database.
Begin
1. Scan S|α, once, and find each time interval weighted frequent item, β in sequences.
(a) β can be assembled to the last element of α to form a time interval weighted sequential pattern or
(b) <β> can be appended to α to form a time interval weighted sequential pattern.
2. For each weighted frequent item β, Add it to α to form a sequential pattern α ', and output α '.
3. For each α ', Construct α '-projected database S|α`;
Call *Pspan(α`, L+1, S|α`)*
End

V. PERFORMANCE EVALUATION

This section discusses our performance study based on the experiment conducted on a randomly generated data set of 50,000 transactions generated by dataset generator class. This class takes the input values for the total number of different items that can be present in the database, the maximum number of items that can be present in an itemset, the maximum number of itemsets in a sequence and the total numbers of transactions in the database. The main purpose of this experiment is to demonstrate how effectively the time-interval weighted sequential patterns can be generated by incorporating a time-interval weight measure along with a support measure. The experiment was performed on a machine operating at 1GHZ with 2GB of memory and the algorithm was implemented in java. The results are compared with PrefixSpan for performance evaluation.

Table 2 shows the results of the experiment conduced for various supports and the corresponding frequent patterns genetared for the algorithms prefixspan, TIWS-1 and TIWS-2. For TIWS algorithms with General scale weighting(TIWS-1) and Log scale weighting(TIWS-2) the same δ and u values (δ=0.9 and u=1) are used

Figs. 1 shows that our algorithm TIWS generates fewer sequential patterns than Prefixspan. Specifically, from the Table 2 it is observed that much fewer sequential patterns are generated in TIWS-1 and TIWS-2 for reasonable support.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

TABLE 2
Test Results for Different Supports (δ=0.9,u=1)

Support	0.06	0.07	0.08
Prefixspan	2550	2550	1430
TIWS-1 (δ=0.9,u=1)	2550	139	50
TIWS-2 (δ=0.9,u=1)	2550	2550	95

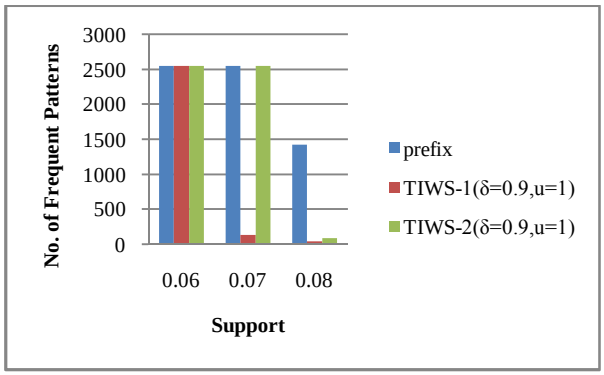

Fig. 1. Support v/s No. of Frequent Patterns

Table 3 shows the results of the experiment conduced for various supports and the corresponding time taken for execution in milliseconds for prefixspan, TIWS-1 and TIWS-2 algorithms with δ=0.9 and u=0.9 values. Table 3 shows that TIWS-1 is faster than Prefixspan.

Figs.2 show that our algorithm TIWS runs faster than Prefixspan. Specifically, from the Table 3 it can be observed that TIWS-1 is faster than Prefixspan for reasonable support.

TABLE 3
Test Results for Different Supports (δ=0.9,u=1)

Support	0.06	0.07	0.08
Prefixspan	303629	296224	190733
TIWS-1 (δ=0.9,u=1)	384121	48445	38829
TIWS-2 (δ=0.9,u=1)	348741	340000	42249

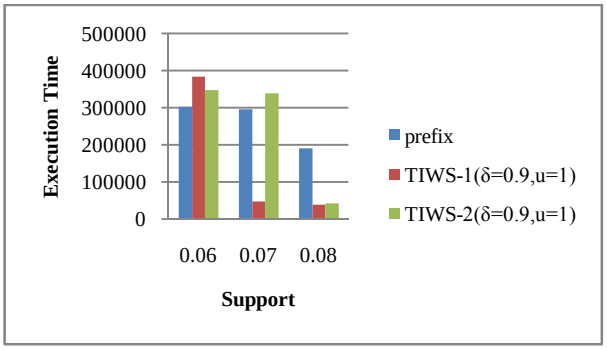

Fig. 2. Support v/s Execution Time

Table 4 shows the results of the experiment conduced for various supports and the corresponding frequent patterns generated for the Prefixspan, TIWS-1 and TIWS-2 algorithms with values (δ=0.9 and u=0.9).

TABLE 4
Test Results Showing the Frequent Patterns Generated for Different Supports (δ=0.9,u=0.9)

Support	0.06	0.07	0.08
Prefixspan	2550	2550	1430
TIWS-1 (δ=0.9,u=0.9)	2550	60	50
TIWS-2 (δ=0.9,u=0.9)	2550	2550	90

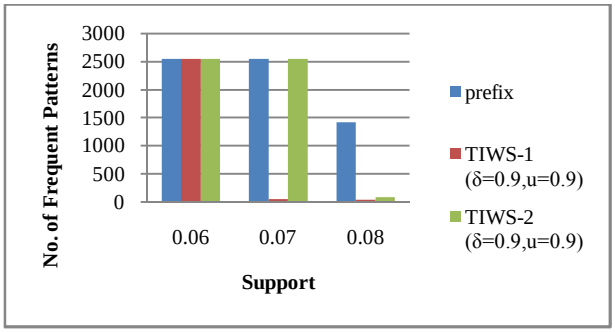

Fig. 3. Support v/s Number of Frequent Patterns

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Figs. 3 show that the algorithms TIWS-1, TIWS-2 generate fewer sequential patterns than Prefixspan algorithm for reasonable support threshold.

Table 5 shows the results of the experiment conduced for various supports and the corresponding execution time taken for the Prefixspan, TIWS-1 and TIWS-2 algorithms with values (δ=0.9 and u=0.9).

TABLE 5
Test Results Showing the Execution Time for Different Supports (δ=0.9,u=0.9)

Support	**0.06**	**0.07**	**0.08**
Prefixspan	303629	296224	190733
TIWS-1 (δ=0.9,u=0.9)	240120	40123	20345
TIWS-2 (δ=0.9,u=0.9)	240000	203204	25315

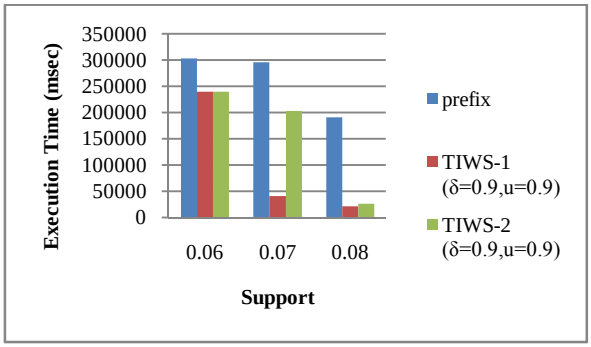

Fig. 4. Support v/s Execution Time

Figs. 4 show that the algorithms TIWS-1, TIWS-2 runs faster than Prefixspan for reasonable support threshold specifically TIWS-1 algorithm takes very less time to generate the sequential patterns.

VI. CONCLUSION

The main limitations of the traditional approach for mining sequential patterns is that all items are treated uniformly, while real items have different characteristics and they generate a very large number of patterns as the minimum support becomes lower. In this paper, we described an approach to mine more interesting sequential patterns by considering the time interval between the elements in the sequences. Our algorithm TIWS focuses on mining the time interval weighted frequent patterns based on the prefix projected sequential pattern growth approach. The experiment results have showed that it is more efficient than prefixspan in generating interesting pattern

REFERENCES

1. Agrawal R. and Srikant R., "Mining sequential patterns", in Proceedings of the 1995, International Conference on Data Engineering (ICDE '95), pp. 3–14, 1995.
2. 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 the 2000 ACM, SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '00), pp. 355–359, 2000.
3. Pei J., Han J., Mortazavi-Asl B., Wang, H. Pinto J., Chen Q., Dayal U. and Hsu M.-C., "Mining sequential patterns by pattern-growth: the PrefixSpan approach", IEEE Transactions on Knowledge and Data Engineering 16 (11), pp.1424–1440, 2004.
4. Chen Y.-L., Chiang M.-C. and Ko M.-T., "Discovering fuzzy time-interval sequential patterns in sequence databases", IEEE Transactions on Systems Man and Cybernetics – Part B: Cybernetics 35 (5), pp 959–972, 2005.
5. Chen Y.-L. and Huang T.C.-H., "Discovering time-interval sequential patterns in sequence databases", Expert Systems with Applications, 25 (1) pp.343–354, 2003.
6. Zaki M.J., "SPADE: an efficient algorithm for mining frequent sequences", Machine Learning, 42 (1/2), pp. 31–60, 2001.
7. Ayres J., Gehrke J., Yiu T. and Flannick J., "Sequential Pattern Mining using A Bitmap Representation", IGKDD'02, 2002.
8. Cheng H., Yan X., and Han J., "IncSpan: Incremental Mining of Sequential Patterns in Large Databases", SIGKDD'04, 2004.
9. Ester M., "A Top-Down Method for Mining Most Specific Frequent Patterns in Biological Sequence Data", SDM'04, 2004.
10. Yun U., "An efficient mining of weighted frequent patterns with length decreasing support constraints", Knowledge-Based Systems 21 (8), pp.741–752, 2008.
11. Ji X., Bailey J. and Dong G., "Mining minimal distinguishing subsequence patterns with gap constraints", Knowledge and Information Systems, 11 (3), pp. 259–296, 2007.
12. Pei J., Han J. and Wang W., "Mining sequential patterns with constraints in large databases", in: Proceedings of the 2002 ACM International Conference on Information and Knowledge Management (CIKM '02), pp. 18–25, 2002.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

BIOGRAPHY

Sirisha A is a Research Scholar in the Computer science Department, College of engineering, JNTU, Hyderabad. She is also working as an Assistant Professor in the department of Information Technology, CBIT, Hyderabad. She received Master of Technology (M. Tech) degree in 2008 from JNTU, Hyderabad, Telangana, India. Her research interests are Data Mining, Natural Language Processing, and Algorithms etc.