IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 9, Issue 6 (Mar. - Apr. 2013), PP 47-52
www.iosrjournals.org

Relevant Tiws Pattern Mining With Reduced Search Space

J. Mercy Geraldine[1], G. Shanthi Krishna[2]
[1] HOD, Department of CSE, Srinivasan Engineering college, Perambalur, India.
[2] M.E Student, Department of CSE, Srinivasan Engineering college, Perambalur, India.

Abstract: *The sequential pattern mining finds all sequential patterns which occurs frequently for a given sequence database whose frequency is no less than the threshold. The weighted sequential pattern mining aims to find more interesting sequential patterns by considering the different significance of each data element in a sequence database. In the conventional weighted sequential pattern mining, pre-assigned weights of data elements are used to get the priorities which are derived from their quantitative information. Generally in sequential pattern mining, the generation order of data is considered to find sequential patterns. However, generation time and time-intervals between the data are also important in real world application domains. Therefore, time-intervals information of data elements can be helpful in finding more interesting sequential patterns. The proposed system presents a framework for finding time-interval weighted sequential (TiWS) patterns in a sequence database. In addition, the CloSpan algorithm is used for mining TiWS patterns in a sequence database which reduces the multiple database scans and the processing time. This is useful in application fields such as web access pattern analysis, customer purchase pattern analysis, DNA sequence analysis, etc.*

Keywords – *CloSpan, Sequential database, Sequential pattern mining, Time-interval weight, TiWS support, TiWS pattern, weighted sequential pattern.*

I. INTRODUCTION

Data mining is the process of analyzing data from different perspectives and summarizing it into useful information. This useful information can be used to increase revenue, cuts costs, or both. Data mining is sometimes called as data or knowledge discovery. Companies have used powerful computers to sift through volumes of supermarket data and analyze market research reports for years. However, continuous innovations in computer processing power, disk storage, and statistical software are dramatically increasing the accuracy of analysis while driving down the cost. Data mining software is one of the analytical tools for analyzing data. It allows users to analyze data from many different angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among some fields in large relational databases.

Sequential pattern mining [1] is a topic of data mining concerned with evaluating statistically relevant patterns between data where the values are delivered in a sequence. It is usually presumed that the values are discrete. And thus time series mining is closely related, but usually considered a different activity. Sequence mining is a special case of structured data mining. A sequence database is a large collection of computerized customer shopping sequence, nucleic acid sequences, protein sequences, or other sequences stored on a computer. A database can include sequences from only one branch, or it can include sequences from multiple branches.

In databases, a huge number of possible sequential patterns are hidden. For satisfying the minimum support threshold, a mining algorithm should find the complete set of patterns. The mining algorithm should be able to incorporate various kinds of user-specific constraints, highly scalable, efficient, involving only a small few database scans. In many of the previous researches on sequential pattern mining, sequential patterns and items in a sequential pattern had been considered uniformly. However, they have a different importance in real world applications are considered in the sequential pattern mining. Based on this observation, prioritized time-interval based sequential pattern mining has been proposed.

General sequential pattern mining is based on simple support counting; if weight of the information is used, it finds interesting sequential patterns. For a sequence or a sequential pattern, not only the generation order of data elements but also their time-intervals and generation times are important to get more valuable sequential patterns. In a sequence database, if the importance of sequences is differentiated based on the time-intervals in the sequences, more interesting sequential patterns can be found.

II. Related Work

Many studies have contributed to efficient mining of sequential patterns, such as GSP, MFS, SPADE, SPAM, FreeSpan, and PrefixSpan algorithms. GSP [1] algorithm makes multiple database passes. In the first pass, all single items are counted. From the frequent items, a set of candidate 2-sequences are formed, and another pass is made to identify their frequency, and this process is repeated until no more frequent sequences are found. MFS [2] algorithm first mines the database samples to obtain the rough estimation of the frequent sequences and then refines the solution.

SPADE [3] decomposes the original problem into smaller sub-problems. All frequent sequences can be enumerated via simple temporal joins and intersections on id-lists. ID-list has a sequence id and the event id. SPAM [4] algorithm uses a depth-first search strategy to generate candidate sequences. The transactional data is stored using a vertical bitmap representation, which allows for efficient support counting as well as significant bitmap compression. FreeSpan [5] is a divide and conquer approach. Frequent items are found from databases. List of frequent items in support descending order is called f_list**.** FreeSpan has to keep the whole sequence in the original database without length reduction of a sequence. Moreover, the growth of a subsequence is explored at any split point in a candidate sequence. It is costly.

PrefixSpan[6] partitions the problem recursively. That is, each subset of sequential patterns can be further divided into next level when necessary. PrefixSpan constructs the corresponding projected databases to mine the subsets of sequential patterns. In these above researches on sequential pattern mining, sequential patterns and items in the sequential patterns are considered uniformly. To improve the usefulness of mining results in real world applications, weighted pattern mining has been studied.

Most of the weighted pattern mining [7][8] algorithms usually require pre-assigned weights, and the weights are generally derived from the quantitative information and the importance of items in a real world application. It does not consider the generation time of each sequence and the items within it. ConSGapMiner[9][10][11] algorithm which uses time-interval and gap information as a constraint. 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

Generally in sequential pattern mining, only data element order is considered. For mining sequential patterns with time-interval, the sequences of a sequence database have its corresponding time stamp values. Let $I = \{i_1, i_{2,}\ldots,i_n\}$ be a set of all items. A sequence $s = <s_1,s_2,\ldots s_l>$ is an ordered list of itemsets, where s_j is one of an itemset, and its time stamp list $TS(s) = <t_1,t_2,\ldots,t_l>$ which stands for the occurrence time. The number of items in a sequence is called as the length of the sequence. For a sequence database, a tuple (sid,S) is said to contain a sequence “a”, if ” a” is a subsequence of s. The count of a sequence A in a sequence database is the number of tuples in the database containing a. And the support of A is the ratio of the count over the size of the sequence database.

The proposed system considers the weight of a sequence, and uses it to find the count of a sequential pattern and the size of a sequence database. For a given sequence database and a support threshold, the problem of time-interval weighted sequential pattern mining is to find the complete set of all time-interval weighted sequential patterns whose weighted supports are no less than the threshold.

IV. Weight Of A Sequence Based On Time-Interval

In mining sequential patterns, a sequence with small time-intervals among its elements can be considered more important than that with large time-intervals. Based on this observation, an interesting mining result of time-interval weighted sequential patterns is found for a sequence database by considering the weight of each sequence obtained from the time-intervals in the sequence. This section presents a technique to get the time-interval weight of a sequence for a sequence database as well as a framework for finding time-interval weighted sequential patterns.

4.1. A time-interval between a pair of itemsets

A sequence in the sequence database consists of itemsets and their corresponding time stamps. For a sequence $S = <s_1, s_{2,} \ldots, s_l>$ and its time stamp list $TS(S) = <t_1, t_{2,} \ldots, t_l>$, the time-interval between two itemsets s_i and s_j in the sequence, that is TI_{ij}, is defined in the equation 1.

$$TI_{ij} = tj - ti \tag{1}$$

If a sequence consists of n itemsets there exist $\frac{n(n-1)}{2}$ pairs of itemsets in the sequence. Consider the sequence S = <a,(abc),d> and its time stamp list TS(S) = <2, 3, 4>. Time interval for each possible pair in sequence is calculated.

Table 1: Possible pairs of itemsets

1^{st} itemset	2^{nd} itemset	Time-interval
a	**abc**	**1**
a	**d**	**2**
abc	**d**	**1**

4.2. A time-interval weight of a pair of itemsets

Normalization is needed to fairly enumerate the time-intervals of different pairs of itemsets in a sequence database. For this purpose, the time-interval weight of the pair is found for each pair of itemsets. Let u (u > 0) be the size of unit time and d (0 < d < 1) be a base number to determine the amount of weight reduction per unit time u, for a sequence S = $<s_1, s_{2,} ..., s_l>$ and its time stamp list TS(S) = $<t_1, t_{2,} ..., t_l>$, the weight of a time-interval TI_{ij} between two itemsets s_i and s_j are defined in equation 2.

$$W(TI_{ij}) = \delta \text{ power } \left(\frac{tj-ti}{u}\right) \qquad (2)$$

4.3. A time-interval weight of a sequence

The time-interval weight of a sequence is computed from the strength and weight of a pair of itemsets as represented in equation 3. Among the itemsets in a sequence, a large-sized itemset may contribute more to the sequence than a small-sized one. For this purpose strength of pair of itemsets are calculated.

$$W(S) = \frac{1}{N}\sum_{i=1}^{l-1} \sum_{j=i+1}^{l}\{w(TIij) \times STij\} \qquad (3)$$

Where $N = \sum_{i=1}^{l-1}\sum_{j=i+1}^{l} STij \qquad (4)$

The strength of a pair of two itemsets s_i and s_j in the sequence ST_{ij} is defined by the equation 4.

ST_{ij} = length(s_i) × length(s_j) (5)

Where length(s_i) denotes the number of items in s_i .

4.4. Time-interval weighted support

Usually, sequential pattern evaluation by support is based on simple counting in the classical sequential pattern mining. But in this section the term TiW-support is calculated by equation 6 which is used to an evaluation process of time-interval weighted sequential patterns in a sequence database.

$$TiW - supp(x) = \frac{\sum_{S:(x \sqsubseteq S)\wedge(S \in SDB)} W(S)}{\sum_{S:S \in SDB} W(S)} \qquad (6)$$

For a given support threshold minSupport (0 < minSupport ≤1), a sequence X is a time-interval weighted sequential pattern if TiW-Supp(X) is no less than the threshold, that is, TiW-Supp(X) ≥ minSupport.

V. MINING TIWS PATTERNS USING CLOSPAN ALGORITHM

In a mining process time-interval weight of each sequence in a sequence database is first obtained from the time-intervals of elements in the sequence. Subsequently, the TiW-support of each sequential pattern is found based on the weight, and a set of TiWS patterns is found considering the TiW-support. Then the clospan algorithm is applied to mine the tiws patterns. Clospan has two major steps. It first generates the lattice sequence set which is a super set of closed frequent sequences and then store it in a prefix sequence lattice. Next step is the post-pruning to eliminate non-closed sequences.

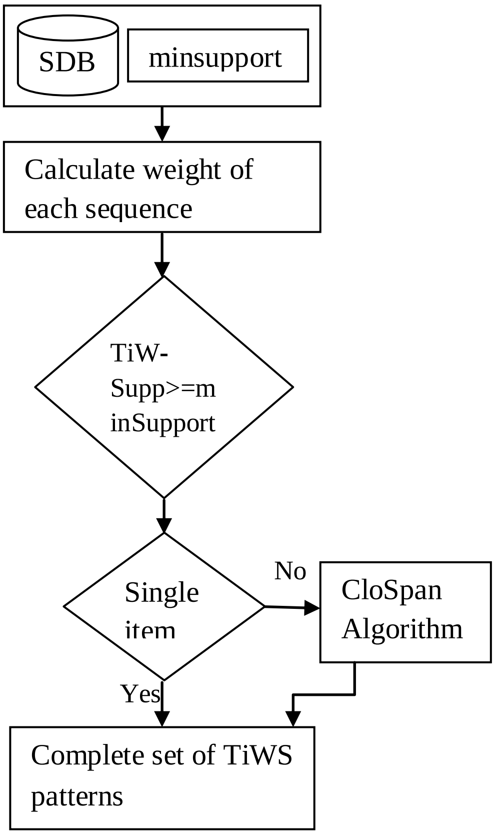

Figure 1: A method of mining TiWS patterns

It first shorts every item and removes infrequent items. Then it calls the clospan recursively by doing depth first search on the prefix search tree and builds the corresponding prefix sequence lattice. The second step is the post-pruning to eliminate non-closed sequences. Clospan outperforms than prefixspan by using the search space pruning techniques. Clospan first checks weather a discovered sequence is exists. The remaining task is to eliminate the non-closed sequences from the prefix sequence lattice. Clospan finds all sequences that have the same support of sequence s. Then it checks whether there is a super sequence containing s.

The sequence database, min-support and the weight function are given as inputs to this process. For each sequence in database, weight and weighted support are calculated from the time-intervals. Then the TiWS-support is compared with the given minsupport. If it is greater than minsupport then it is taken as time-interval weighted pattern. The Clospan algorithm is applied on those patterns. The time-interval weight of each sequence in a sequence database is obtained in the first scan of the sequence database, and it is used to find TiWS patterns. In TiWS pattern mining for a sequence database, sequential patterns with large time-intervals may be excluded from its mining result, so that the number of scans for the sequence database or its subset can be reduced. As a result, the processing time to get the mining result can also be reduced.

VI. Performance And Evaluation

The weighted time-interval information is combined with the CloSpan algorithm to produces more interesting sequential patterns by reducing the size of the database. This reduces the number of scans. The performance of CloSpan algorithm related to PrefixSpan algorithm is as follows. CloSpan prunes the search space more deeply compared to the PrefixSpan algorithm by using the pruning technique.

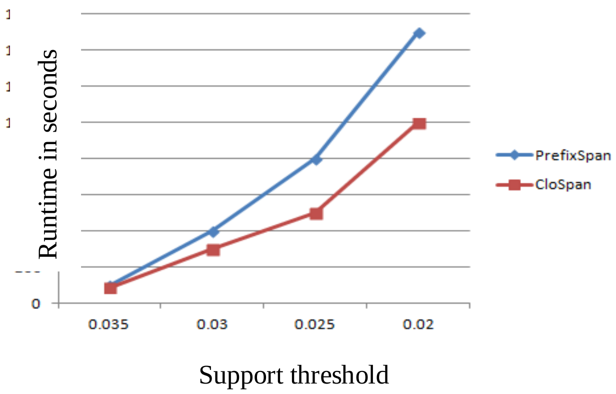

Figure 2: Comparison among CloSpan and PrefixSpan

A frequent closed sequence mining algorithm like CloSpan can definitely outperform frequent sequence mining algorithm like PrefixSpan when the support threshold is low. Considering the time-interval information provides most important sequences. The combination of TiWS and CloSpan algorithm provides most interesting sequences which is closer to the user's expectation.

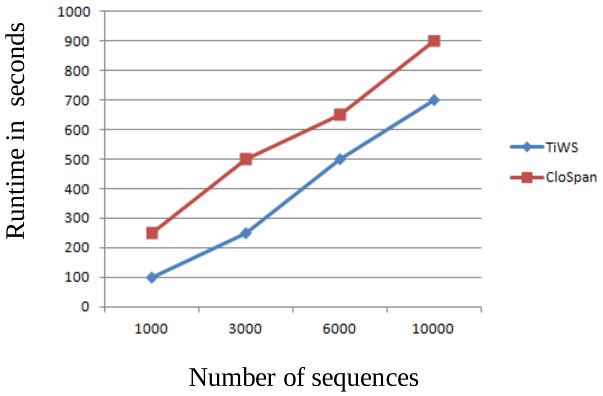

Figure 3: Performance comparison.

VII. Conclusion And Future Work

The survey of the traditional sequence pattern mining algorithms has revealed that those algorithms do not consider the weight of the time-intervals. However, considering the weight of a time-interval between the items in each sequence is important. Therefore, an efficient algorithm called TiWS making use of CloSpan algorithm to mine TiWS patterns is proposed. This algorithm mines the sequential patterns based on the weight of the time-intervals, requiring the patterns to abide by the threshold value specified by the user. This provides more interesting sequential patterns. It reduces the number of scans and the processing time. An interesting direction for future research could be in defining other approaches to get the time-interval of a sequence and other weighting functions.

References

[1] [13] Minghua. Zhang and Ben. Kao. A GSP- based Efficient Algorithm for Mining Frequent Sequences. In HKU CSIS Tech Report TR-2002-05.

[2] [8] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In Proc.5th Int. Conf. Extending Database Technology(EDBT'96), pages 3–17, Avignon, France, Mar. 1996.

[3] [12] Mohammed J. Zaki, SPADE: An Efficient Algorithm for Mining Frequent Sequences, Computer Science Department, Rensselaer Polytechnic Institute, Troy NY 12180-3590

[4] J. Ayres, J. Gehrke, T. Yiu, and J. Flannick, Sequential PAttern Mining using a Bitmap Representation. In SIGKDD'02, Edmonton, Canada, July 2002.

[5] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, M.-C. Hsu, FreeSpan: frequent pattern-projected sequential pattern mining, in: Proceedings of the 2000 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD '00), 2000, pp. 355–359.

[6] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, M.-C. Hsu, Mining sequential patterns by pattern-growth: the PrefixSpan approach, IEEE Transactions on Knowledge and Data Engineering 16 (11) (2004) 1424–1440.

[7] U. Yun, An efficient mining of weighted frequent patterns with length decreasing support constraints, Knowledge-Based Systems 21

(8) (2008) 741–752.

[8] Joong Hyuk. Chang, Mining weighted sequential patterns in a sequence database with a time-interval weight, In the Knowledge-Based Systems 24 (2011) 1–9.

[9] Yi-Cheng Chen and Ji-Chiang Jiang. An Efficient Algorithm for Mining Time Interval-based Patterns in Large Databases. In CIKM'10, October 26–30, 2010.

[10] X. Ji, J. Bailey, G. Dong, Mining minimal distinguishing subsequence patterns with gap constraints, Knowledge and Information Systems 11 (3) (2007) 259–296.

[11] J. Wang, J. Han, C. Li, Frequent closed sequence mining without candidate maintenance, IEEE Transactions on Knowledge and Data Engineering 19 (8) (2007) 1042–1056.

[12] X. Yan, J. Han, R. Afshar, CloSpan: mining closed sequential patterns in large datasets, in: Proceedings of the 2003 SIAM International Conference on Data Mining (SDM '03), 2003, pp. 166–177.