National Conference on Emerging Trends in Computer, Electrical & Electronics (ETCEE-2015)
International Journal of Advance Engineering and Research Development (IJAERD)
e-ISSN: 2348 - 4470 , print-ISSN:2348-6406,Impact Factor:3.134

A data mining approach to discovering reliable Incremental sequential patterns

Nirali Parmar[1], Trupti Kodinariya[2]

[1,2]*Computer Department, Gujarat Technological University*
Address

[1]np2492@gmail.com

[2]trupti.kodinariya@gmail.com

***Abstract*— Sequential pattern mining is one of the data mining method for obtaining frequent sequential patterns in a sequential database. Generally sequential data mining methods could be divided into two categories: Apriori-like methods and pattern growth methods. In any sequential pattern, probability of time between two adjacent events could provide valuable information for decision-makers. As we know, there has been no methodology developed to extract this probability in the sequential pattern mining process. Here we extend the IncSpan algorithm and propose a new sequential pattern mining approach: T-IncSpan. This approach imposes minimum time-probability constraint, so that fewer but more reliable patterns will be obtained. T-IncSpan is compared with IncSpan in terms of number of patterns obtained and execution efficiency. Our experimental results show that T-IncSpan is an efficient and scalable method for sequential pattern mining.**

***Keywords*— Data mining, Sequential patterns, Inter-arrival time probability**

I. INTRODUCTION

Sequential pattern mining, a well-studied and important issue in data mining field, is for analyse frequent subsequences as patterns in a sequence database. It is also an active and important research topic in data mining, with so many applications, such as mining web logs, customer shopping transaction analysis, mining DNA sequence etc. Data mining known as knowledge discovery of database(KDD) is an efficient way of extracting required knowledge from given or available large amount of dataset. It becomes very helpful in analysing various itemset and various patterns.

A well-known algorithm IncSpan [1], based on Prefixspan approach was proposed that has interesting properties. When mining the original database, IncSpan buffers the frequent sequences, as well as the sequences that are semi-frequent, which are likely to become frequent in the new version. Afterward, when mining the new version, only sequences with support over a certain threshold are likely to become frequent, out of un-buffered sequences. It divides the database into smaller projected database and solves them recursively. Since no candidate sequence needs to be generated, the database need not be scanned multiple times, thus making it faster than Apriori-like algorithms. The algorithm has improved performance due to early candidate pruning, effective implicit merging and efficient separate counting.

Mostly sequential pattern mining algorithms do not address the intervals between consecutive items, it mine conventional sequential patterns including only the orders of items. A sequential pattern including probabilities of time between two consecutive items can produce more valuable information than a conventional sequential pattern. It is required in many real-world applications. Take one example of it, a physician would like to know the probability that symptom B will appear within few days, once a patient has symptom A. While previous studies have considered several variations, sequential patterns with probability of time are not revealed.

II. RELATED WORK

Data mining provides a proficient way to extract useful information and helpful knowledge from large amount of data that are explored in a gigantic manner day by day. There are so many tasks that can be performed with the help of data mining. Like, Incremental sequential pattern mining association rule mining classification, clustering etc. From all these important tasks, the paper provided here focuses on incremental sequential pattern mining with various approaches of finding frequent pattern. Agrawal and srikant proposed an algorithm, name of it is AprioriAll for obtaining frequent sequential patterns. With their approach, candidate frequent sequential patterns can be obtained by joining shorter frequent sequential patterns.

Following AprioriAll, proposed a new algorithm named GSP(Generalizations and performance improvements), which uses a bottom-up method and breadth-first search to obtain the frequent subsequences. It also considers mining sequential patterns with timing constraints regarding the maximal time gap, minimal time gap, and sliding window size. The consideration of timing constraints is depending on customer behaviours. Another advantage of the inclusion of timing constraints is that patterns not satisfying the timing constraints are filtered out. Thus, the number of candidate sequential patterns would be reduced.GSP met real-world requirements than AprioriAll. When the database or the number of possible items grows, the extensive candidate sequential patterns and the increased number of database scans have a huge impact on performance. furthermore, the methodology cannot find a pattern whose interval between two consecutive items is not in the range, and the sequential patterns include only the temporal order of the items. Agrawal and Srikant [2] formulated GSP to address three cases: (1) The presence of time constraints that specify a maximum time between adjacent elements in a pattern. The time window can apply to items that are bought in the same transaction or in different transactions. (2) The relaxation of the restriction that the items in an element of a sequential pattern must come from the same transaction. (3) Allowing elements in a sequence to consist of items that belong to taxonomy.

SPADE (Sequential pattern discovery using equivalent class) is an algorithm for fast discovery of sequential patterns using a depth-first search and bottom-up method. As like horizontal formulating methods (GSP) the sequential dataset can be transformed into a vertical dataset format consisting of item id-lists. The vertical dataset list is the list of(sequential-id, timestamps) pair indicating the occurring timestamps of the item in that sequence. The searching in the format of dataset is done by the id-list interaction, this SPADE algorithm complete the mining in total three passes of database scanning. In addition to this the computation time requires to transform in the horizontal dataset to vertical dataset and also require additional storage space several times larger than that of the original sequence database.

Sequential pattern mining(SPAM) is provided in ayers et al.[3] integrates the ideas of GAP, SPADE, and FreeSpan. The entire algorithm with its data structures fits in main memory, and is claimed to be the first strategy for mining sequential patterns to traverse the lexicographical sequence tree in depth-first fashion. Each node in the tree has sequence-extended children sequences generated in the S-Step of the algorithm, and itemset-extended children sequences generated by the I-Step of the algorithm at each node. SPAM traverses the sequence tree in depth-first search manner and checks the support of each sequence-extended or itemset-extended child against min support recursively. If the support of a certain child s is less than min support, there is no need to repeat depth first search on s by the apriori property. Apriori-based pruning is also applied at each S-Step and I-Step of the algorithm, minimizing the number of children nodes and making sure that all nodes corresponding to frequent sequences are visited. For efficient support-counting, SPAM uses a vertical bitmap data structure representation of the database similar to the id list in SPADE. Each bitmap has a bit corresponding to each element of the sequences in the database. Each bitmap partition of a sequence to be extended in S-Step is first transformed using a lookup table, such that all the bits after the index of the first “1” bit(call it index y) are set to one and all the bits with index less than or equal to y are set to zero. When SPAM was compared to SPADE, it was found to outperform SPADE by a factor of 2.5, while SPADE is 5 to 20 times more space efficient than SPAM, making the choice between the two a matter of a space-time trade-off. Here, Ayres et al.[3] propose to use a compressed bitmap representation in SPAM tp save space, but do not elaborate on the idea.

Another algorithm is Freespan [4], is based on the following property: If an itemset X is infrequent, any sequence whose projected itemset is a superset of X cannot be a sequential pattern. Freespan mines sequential patterns by partitioning the search space and projecting the sequence sub databases recursively based on the projected itemsets. Given the database S and min_support θ1, Freespan first scans S, collects the support for each item, and finds the set of frequent items. Frequent items are listed in support descending order.

Later on proposed Prefixspan for dealing with the problem of freespan: that is the length of original database sequences could not be shortened during the mining process. Their principle is to check the frequency of patterns from the prefix sub-sequences. For those prefixes that pass the frequency threshold, the suffix sub-sequences of each original sequence would be inserted into their projected database. Each projected database would then generate its frequent patterns recursively. The advantage of partitioning the search space into the projected database is that each projected database would contain only the required mining information with respect to the prefix. Along with the growth of the frequent patterns, the projected database will shrink, making prefixspan perform better than Freespan in general. However, since the generated projected databases occupy a lot of memory space, When the database or the number of items is huge, the memory space may be insufficient to store the projected databases. The algorithm CFR-postfixspan improved prefixspan by adding timing requirements of regency and compactness. The added constraints could filter out less important patterns and reduce the memory space required in storing projected databases.

SuffixTree techniques were proposed [5] in which deal with incremental sequential pattern updating. SuffixTree has only to maintain the data reading after the update, for this reason it is a very appropriate method for incremental sequence extraction. But, this algorithm presents the complexity in space which depends on the size of the database, which presents the main limitations of this method. Further, the sensitivity of the position to the update operation makes Suffixtree very expensive for dynamic strings.

Another Incremental based algorithm is Incspan [6] is used for incremental mining over multiple database increments. Incspan algorithm development is based on two novel ideas. The first idea which presents a several good properties and lead to efficient practices is the use of a set of “almost frequent” sequences as the candidates in the updated database. The second idea is constituted by two optimization techniques designed to improve the performance, which are reverse pattern matching and shared projection. The first technique is used for matching a sequential pattern in a sequence. Reverse pattern matching can prune additional search space, while the appended transactions are at the end of a sequence. Shared projection is intended to reduce the number of database projections for some sequences having a common prefix. Empirical study shows that Incspan is better than ISM and Prefixspan on incrementally updated databases by a wide scope.

Incspan+ proposed by [7] to improve Incspan. The authors agree that the algorithm Incspan cannot find the complete set of frequent sequential patterns in the updated database D’, that means, it violates the correctness condition. The proposed algorithm ensures the correctness of mining result in the updated database. Incspan+ ensures two important tasks: (1) The discovery of the complete FS', which guarantees the correctness of the mining result and (2) the discovery of the complete SFS', which is helpful in incrementally maintaining the frequent patterns for further database updates.

Although consecutive events in the sequential patterns obtained from previous data mining algorithms reveal their timing orders, the time between events is not determined. To solve the problem,[8] in proposed work present a new pattern, 'the time-interval sequential pattern', which includes not only the order of the events but also the time intervals between two consecutive events? They divided the complete time domain into several fixed and no overlap time ranges. A time interval sequential pattern has a form like (E1,I1,E2,I2,E3), meaning that event E1 happens first, followed by events E2 and E3 after time intervals of I1 and I2, respectively. Both I1 and I2 are predetermined time ranges.

In this research, we build an efficient algorithm called I-Incspan for sequential pattern mining. Between two consecutive events in a frequent sequential pattern, not only the time interval but also its probability will be revealed in the mining process. It is developed by extending the well known Incspan algorithm and assuming that the time between two consecutive items or events follows an exponential distribution with constant arrival rate.

III NOTATIONS

In the paper of Pei et al.[8], a sequence is defined as an ordered list of itemsets, denoted as <I1, I2, . . ., In>. Items in the same itemset will be enclosed by the parentheses '('and')'. For example,<(bc),a,d,(aef),c> denotes the sequence of the following itemsets: {b,c}, {a}, {d}, {a,e,f}, and {c}. To simplify the above presentation, when there is only one item in an itemset, the parentheses will be omitted. In this study, each of the items is attached to a transaction time and the items in an itemset share the same transaction time. A sequence is represented as <[q1,t1], [q2,t2], . . ., [qn,tn]>,where qj is an item and tj stands for the transaction time at which qj occurs, $1 \leqslant j \leqslant n$, and $t1 \leqslant t2 \leqslant \cdots \leqslant tn$. <α> = <α1, α2, . . ., αm> denotes the sequential pattern. With the above representation, the following definitions are defined.

Definition 1 . For a sequence Q = <$[q_1, t_{q1}]$, $[q_2, t_{q2}]$, . . ., $[q_n, t_{qn}]$> and a pattern P = <p_1, p_2, . . ., p_m>, P is a time-relaxation prefix of Q if $p_1= q_1$, $p_2= q_2$, . . ., $p_m= q_m$ where $m \leqslant n$.

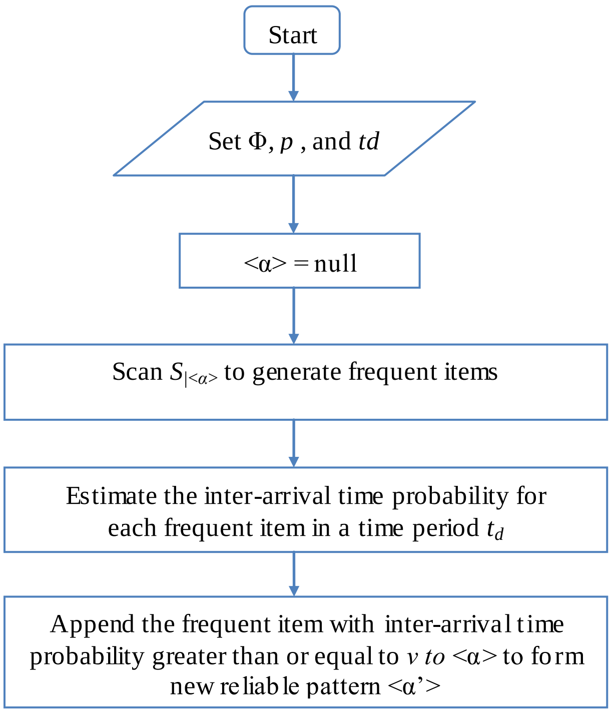
Definition 2. A sequence P = <$[p_1, t_{p1}]$, $[p_2, t_{p2}]$, . . ., $[p_m, t_{pm}]$> is a time-relaxation subsequence of Q = <$[q_1, t_{q1}]$, $[q_2, t_{q2}]$, .., $[q_n,t_{qn}]$>, if there exist integers $1 \leqslant j_1 < j_2, . . ., < j_m \leqslant n$, such that $p_1=q_{j1}$, $p_2= q_{j2}$, . . . , $p_m= q_{jm}$. And Q is defined as a time-relaxation super sequence of P.

Definition 3. A time-relaxation subsequence P'of sequence P is called a projection of P with respect to the time-relaxation prefix R if (1) P' has time-relaxation prefix R and (2) there exists no proper time-relaxation super sequence P'' of P' such that P' is a time-relaxation subsequence of P and has a time-relaxation prefix R. If one removes directly the time-relaxation prefix R from the projection P', the new sequence is called the postfix of P with respect to time-relaxation prefix R. For example, in sequence <[a,5], [c,6],[d,10], [c,10], [e,12]>,the time-relaxation subsequences P^1= <[_,6], [d,10],[c,10],[e,12]> and P^2= <[_,10],[e,12]> are postfixes of P with respect to time-relaxation prefix <c>. Note that [,6] and [,10] means that the transaction times of time-relaxation prefix in P^1 and P^2 are 6 and 10, respectively.

Definition 4. Let T_{max} denote the occurrence time of the last transaction in a sequence database S. Q = <$[q_1, t_{q1}]$, $[q_2,t_{q2}]$, . . , $[q_n, t_{qn}]$> is a sequence in S. {$Q(t_1,t_2)$} represents all the items that exist in Q between transaction time $t_1$and t_2. Assume that P = <$[p_1, t_{p1}]$, $[p_2, t_{p2}]$, . . ., $[p_m, t_{pm}]$> is a time-relaxation subsequence of Q, and β is an item which does not belong to {$Q(t_{pm},t_{qn})$}.According to the current sequence Q, we do not know the exact transaction time of β. To analyze the probability of time interval of consecutive items, the potential censoring time for item β with respect to P is defined as $T_{max} - t_{pm}$.

Keywords used in steps of T-IncSpan

Φ - The user-specified minimum support threshold
p - user-specified minimum probability threshold
t_d -predefined time period
<α> - frequent pattern
|<α >| - the length of <α>
$S_{|<\alpha>}$ - the projected database of S with respect to <α>


Start
Set Φ, *p* , and *td*
<α> = null
Scan $S_{|<\alpha>}$ to generate frequent items
Estimate the inter-arrival time probability for each frequent item in a time period t_d
Append the frequent item with inter-arrival time probability greater than or equal to *v to* <α> to form new reliable pattern <α'>

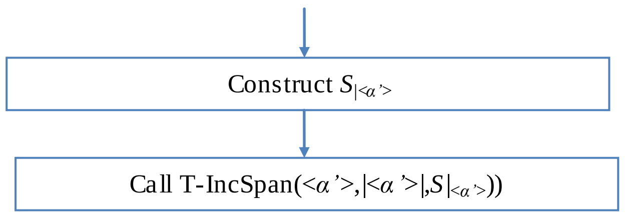

Construct $S_{|<\alpha'>}$
Call T-IncSpan(<α'>,|<α'>|,$S_{|<\alpha'>}$))

Fig.1 The steps of T-IncSpan

A sequential pattern <α> is called a frequent pattern if the percentage of transactions in S containing < α> is greater than or equal to the user-specified minimum support threshold - Φ.In this paper, a frequent pattern < α> is called a reliable pattern if the estimated inter-arrival time probability of any two consecutive items in < α > in a predefined time period t_d is larger than the user-specified minimum probability threshold - p. The searched sequential patterns should be frequent and reliable patterns in the proposed algorithm. It should be noted that if p is set to be 0, then all frequent patterns are reliable patterns. In that case, T-IncSpan and IncSpan will obtain the same results.

IV PROPOSED ALGORITHM

The T-Incspan algorithm is developed by modifying the well-known Incspan algorithm. Given a sequence database S and a pattern<α>, we use projected database $S|_{<\alpha>}$ to denote the collection of postfixes in S with respect to < α>. The length of pattern < α> represents the number of items in < α>, denoted as |< α>|.

Consider the sequence database S shown in Table 1, where ID denotes the identity of sequence and T^{max} = 18. Four sequences are observed. Suppose Φ (minimum support threshold) = 2, p (minimum probability threshold) = 0.3, and t_d (expected time period) = 7.Then, the probability of the occurrence of a frequent item following pattern <α> within 7 days must be greater than or equal to 30%.

```
Input:
        S - sequence database
        Φ - The user-specified minimum support threshold
        p - user-specified minimum probability threshold
        $t_d$ -predefined time period
Output:
        A complete set of reliable patterns
Subroutine:
        T-IncSpan(<α>,|< α >|,S|< α >)
Parameters:
        < α > : a time-relaxation IncSpan
        |< α >|: the length of < α >
        $S_{|<\alpha>}$: the projected database of S with respect to <
                α >
Algorithm:
        Step 1: ScanS|< α > one time. Find all frequent items
                in $S|_{<\alpha>}$
        Step 2: If |< α >| = 0, then For each frequent item ,
                append to < α > as < α'>;
        Step 3 : If |< α >| > 0,then For each frequent item β
                (a) λ = arrivalRate (<α>, S|<α >, β )
                (b) IATP = 1 - eλtd
                (c)If IATP > p then append β to <α> as
                <α'>
        Step 4: For each < α'> Construct S|< α'>, the
                projected database of < α'>
                Call T-IncSpan (<α>,|< α'>|,S|< α'>)
```

In the beginning, <α> is set to be null. Scan the sequential database once. Five length-1 frequent items – a, b, c, d, and f are found. Appending all the frequent items to <α> yields five different <α'>, and T-IncSpan() must be called again with new parameters for each <α'>.Consider the case with <α'> = <α>. The projected database S|<α> will be constructed and is shown in Table 2.

Table 1 Sequence database

ID	Sequence
0	<[b,2],[c,2],[a,4],[d,7],[a,8],[f,8],[c,15]>
1	<[a,0],[d,5],[f,5],[b,12]>
2	<[a,1],[b,1],[c,1],[f,4],[a,6],[c,6],[b,8],[c,9]>
3	<[c,5],[a,7],[b,15],[d,18],[f,18]>

Table 2 Projected database for <α>

ID	Projected (postfix) database
0	<[,4],[d,7], [a,8], [f,8],[c,15]> <[,8],[f,8], [c,15]>
1	<[,0],[d,5],[f,5],[b,12]>
2	<[,1],[b,1],[c,1],[f,4],[a,6],[c,6],[b,8],[c,9]> <[,6],[c,6],[b,8],[c,9]
3	<[,7],[b,15],[d,18],[f,18]>

By mining the projected database with the predefined Φ, the frequent tems in $S_{|<\alpha>}$ can be found. They are a, b, c, d, and f. In the next step, for each frequent item, the probability of inter-arrival time with respect to <α> will be estimated. Take frequent item b for example. The arrival rate of b can be calculated by the following equation:

$$\lambda_{b|(a)} = \frac{\gamma}{\sum_{\forall s\in S_1} T_{b|(a)} + \sum_{\forall s\in S_2} T^{+}_{b|(a)}}$$

$$= \frac{4}{(12-0)+(1-1)+(8-6)+(15-7)]+[(18-4)+(18-8)}$$

= 0.087

Then the probability of inter-arrival time being less than or equal to t_d= 7 can be approximated as

$$P(T^b \leqslant t_d) = 1 - P(T^b > 7) = 1 - e^{-0.087\times 7} = 0.456 > p = 0.3.$$

Since p < 0.456, frequent item b can be appended to <α> to form a new reliable pattern <α> with length 2. In the same manner, the arrival rate and the approximated probability can be calculated for the above projected database and the results are shown in Table 3.Given time relaxation prefix <α>, three different reliable patterns <a,b>,<a,c>, and <a,f> are derived. Recursively finding the reliable pattern in S|<α> finally yields all the reliable patterns in S.

The application of inter-arrival time probability in the real life could be explained in the following scenario. Suppose while promoting the item a, a retailer would like to make a decision on which item to promote next within 7 days.

Table 3 Arrival rate and inter-arrival time probability

Pattern	**Arrival rate**	**Inter-arrival time Probability**	**Large or equal to p**
<a,a>	0.033	0.208	No
<a,b>	0.087	0.456	Yes
<a,c>	0.072	0.399	Yes
<a,d>	0.045	0.273	No
<a,f>	0.142	0.632	Yes

By using T-IncSpan, it can be found from Table 3 that the probability of purchasing a, or d is lower than 30%, while that of f is greater than 50%. Therefore, item f would be a better choice for the next marketing campaign.

V. CONCLUSION

Most studies of sequential pattern mining concentrate on symbolic patterns, whereas numerical analysis usually belongs to the scope of trend analysis and forecasting. This paper proposed a new algorithm, T-IncSpan, for mining reliable sequential patterns, which concentrate on symbolic patterns and numerical analysis simultaneously. The reliable sequential pattern can yield information not only of the order of frequent items but also of the time probability of arrival items. The information, which provides a more detailed description about the behaviour of derived patterns, is crucial to decision makers. When decision-makers specify the time period of concern to them, the inter-arrival time probability for every two consequent items in a reliable pattern can be estimated. In practice, reliable sequential patterns are useful for analysing the purchase behaviour of customers, the symptom behaviour of disease, and many others.

REFERENCES

[1] Ayres, Jay, Jason Flannick, Johannes Gehrke, and TomiYiu. "Sequential pattern mining using a bitmap representation." In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 429-435. ACM, 2002.ISBN:1-58113-567-X

[2] ANKHBAYAR YUKHUU," Mining Sequential Patterns by PrefixSpan algorithm with approximation", Department of Computer Science, Sun Moon University, 100 Kalsan-ri, Tangjeongmyeon, Asan ; Chugnam, 336-708 SOUTH KOREA

[3] Chen, Y.L., Chiang, M.C., Ko, M.T., 2003. Discovering time-interval sequential patterns in sequence databases. Expert Systems with Applications 25, 343–354.

[4] Hong Cheng, Xifeng Yan, Jiawei Han "IncSpan: Incremental Mining of Sequential Patterns in Large Database"

[5] K.Wang, J.Tan .Incremental discovery of sequential patterns. In Proc of Workshop on Research Issues on Data Mining and Know Discovery. June 1996, Montreal, Canada. 95–102 10414

[6] Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.C., 2001.Prefixspan: mining sequential patterns efficiently by prefix-projected pattern growth. International Conference on Knowledge Discovery in Databases and Data Mining, 215–224.

[7] S.N.Nguyen, X.Sun and M.E.Orlowska. Improvements of IncSpan: Incremental mining of sequential patterns in large database. In the Proceedings of the 9th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD 2005). 9th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD 2005), Hanoi, Vietnam, (442-451). 18-20 May2005.

[8] Yan, X., J. Han, and R. Afshar, CloSpan: Mining Closed Sequential Patterns in Large Datasets, Proceedings of 2003 SIAM nternational Conference Data Mining, 2003.