Hybrid Technique for Frequent Pattern Extraction from Sequential Database

Rajalakshmi Selvaraj[1], Venu Madhav Kuthadi[2], and Tshilidzi Marwala[3]

[1] Faculty of Engineering and the Built Environment, University of Johannesburg, South Africa & Department of Computer Science, BIUST, Botswana
[2] Department of AIS, University of Johannesburg, South Africa
[3] Faculty of Engineering and the Built Environment, University of Johannesburg, South Africa

Abstract. Data mining has became a familiar tool for mining stored value from the large scale databases that are known as Sequential Database. These databases has large number of itemsets that can arrive frequently and sequentially, it can also predict the users behaviors. The evaluation of user behavior is done by using Sequential pattern mining where the frequent patterns extracted with several limitations. Even the previous sequential pattern techniques used some limitations to extract those frequent patterns but these techniques does not generated the more reliable patterns .Thus, it is very complex to the decision makers for evaluation of user behavior. In this paper, to solve this problem a technique called hybrid pattern is used which has both time based limitation and space limitation and it is used to extract more feasible pattern from sequential database. Initially, the space limitation is applied to break the sequential database using the maximum and minimum threshold values. To this end, the time based limitation is applied to extract more feasible patterns where a bury-time arrival rate is computed to extract the reliable patterns.

Keywords*:* Sequential pattern, Data mining, Hybrid pattern mining.

1 Introduction

In the Sequence database, the sequential pattern mining establishes the frequent subsequences as patterns and also it is an essential data mining issue in the broad applications which includes the learning of client purchase behavior, DNA sequences, Disease treatments, Web access patterns, Scientific experiments, Natural disasters etc,.[1]. More efforts are put up to develop an effective algorithm for the purpose of searching frequent sequential patterns. Existing sequential pattern mining algorithm is divided into two approaches and they are of Apriori which is of candidate-generation [2] [3] [4] [5] and pattern-growth approach [6] [7] [8] [9].

Initially, the issues of sequential pattern mining were proposed by Srikant and by Agrawal in [3]. “A set of sequences is mentioned in which each sequences has a list of elements and those elements has a set of items and also an user least support threshold is given. The sequential pattern mining is to identify the entire frequent

S.C. Satapathy et al. (eds.), *Proc. of the 3rd Int. Conf. on Front. of Intell. Comput. (FICTA) 2014 – Vol. 1*, Advances in Intelligent Systems and Computing 327, DOI: 10.1007/978-3-319-11933-5_29

subsequences i.e. "Those Subsequences which has occurrence frequency but not less than the minimum support in the set of sequences".

Many sequential pattern mining algorithms won't deal with the intervals among consecutive item. The mine conventional sequential patterns deals particularly with the items order [10]. The sequential pattern which includes possibilities of time among two consecutive items gives essential information rather than the convectional sequential pattern. So, it is very important in more real time applications.

Considering an example i.e. a product of items A and B are sold in a supermarket. Also, the supermarket transaction is identified the possibility of the product that are been sold within 1, 2, 3 days and rest of the days once a customer has product A. As a result, existing research work are taken with various variations, sequence patterns with their probabilities of time is not shown.

Prefix Span Algorithm [9]: It is a recognizable pattern growth approach and it separates the database in to small probable databases and resolves them as it doesn't require to create any candidates sequence and doesn't require to scan the database at many times. Therefore it is faster than the Apriori-like Algorithm. Moreover, it is very clear that the Traditional Prefix Span algorithm functions slowly when there are huge numbers of frequent subsequences [9].

P-Prefix Span algorithm [6]: It is also a recognizable pattern growth approach. In this the frequent patterns are identified as per the possibility of inter-arrival time. Moreover, the frequent occasion of sequence pattern affect the sequence database, so this problem can increase the difficultly to obtain the end results.

This paper introduces a latest technique known as Hybrid Pattern mining algorithm in order to solve the mining problems. This hybrid pattern mining algorithm will extract the feasible patterns from sequential database by different specified limitation of the user. At first, the proposed algorithm will divide the sequential database according to the user specified max-min space limitations.

The sequence database is classified by hybrid pattern mining technique which is been identified as a continuous sequential patterns using probability of buying time of frequent items. The proposed algorithm is executed as to reestimate and support to search for frequent patterns using the estimated probability of end time. A new metric namely called probability-based pattern evaluation is introduced in this paper. The main aim of sequential pattern mining is to obtain whole patterns and their probability from the given set of transactions. The probabilities are obtained from a predefined time-period length that must be greater than least probability threshold. Also, to extract those type of sequential patterns, the probabilities should be verified when a item is mutated to sequential pattern such as if frequent item χ is to be combined to a frequent sequential pattern (ω), T represents the duration of time among the two repeated items as well as the least-probability threshold and is described as $P(T \leq 20) = 50\%$. Let $T_{\chi|(\omega)\ is}$ to explain the time period among the occurrence of frequent item χ and the last occurrence item in (ω). If $P(T_{\chi|<\omega>} \leq 20) > 50\%$, then frequent item χ will be combined to (ω) creates a novel sequential pattern (ω'), or else, frequent item χ which cannot be create a novel pattern. The estimation of time probability is depended on the space constraints and Least-Prop-Threshold, however proposed algorithm requires less calculation time than existing techniques and searched patterns are more reliable.

The remaining paper is framed as: In section 2, the related work is discussed. In section 3, the proposed algorithm is discussed. In section 3, the result of proposed algorithm is discussed. At last in section 4, conclusion is discussed.

2 Related Work

The important problem of sequential pattern mining is the huge processing cost because of extracting the patterns from huge amount of data. Numerous of algorithms are introduced to expedite mining process. In that, SPAM [8], P-Prefix span [6], GSP [3], Prefix span [9], AprioriAll [2] and SPADE [11] are the representatives.

AprioriAll [2] is well-known three phase algorithm where it initially identifies the entire item sets with min support (frequent itemsets) and alters the database so that every transaction is altered with the set of entire frequent itemsets and then it identifies the sequential database. Actually, there are two issues in this approach. The first issue is that it is very expensive to perform the data transformation on the fly at every pass while finding the sequential database and also to modify the database and to store the modified database is unfeasible or else impractical in lots of applications as it doubles the disk space needs which are too expensive for huge database. The second issue is that it doesn't show any feasible to integrate sliding windows if it is probable to enlarge this algorithm in order to manage the time limitation and taxonomies.

In GSP algorithm, Agrawal and Srikant [2] implemented a bottom up method where this method creates initially frequent 1-sequences after that it creates frequent 2-sequences until the pattern has discovered. Also, this method creates the candidate n-sequences which are too from the frequent (n-1) –sequences in the every iteration. Also, it is according to the anti-monotone property where entire subsets of a frequent item sequence must be in frequent at all times. The candidate n-sequences are determined by their frequent which is measured by their supportive counts where in the situation of all iteration. For the purpose of supportive counts, this algorithm is expensive in order to decompose the item sequences.

Huge number of candidate sequences is the other issue of this GSP algorithm. In order to solve this issue [11] proposed the lattice idea using with the algorithm namely called SPADE where it differentiate the candidate sequence into several set of items, also is stored the every sets successfully in the primary memory itself. Moreover, the algorithm namely called GSP that utilizes ID List method in order to minimize the cost of measuring supportive counts. The sequence of ID pairs is kept in the ID list where it signifies the exact position in the database. Moreover, it doesn't create reliable patterns.

The FP-tree [7] is a well-known short come of AprioriAll to generate the huge sequential patterns. This method is utilized a FP-tree based data structure. The tree structure which is utilizing the prefix method for reorder the all transaction or frequent items. In subsequences, if the sequence of various items orders is larger than its re-arrangement method of the sequences doesn't execute properly. For the purpose of dealing the FP-Tree issues, Free Span [8] is utilized. The main concept of this Free

Span is to create the projected or prefix based sequential databases of particular frequent item patterns. Then, these frequent patterns have the capability to grow by just linking the frequent subset items from the small projected or prefix databases. This Free Span can get the entire frequent patterns by this way. The sequence of created candidate patterns are very less than the received candidate patterns from the combining technique as well as the database scanning cost is very cheap.

Many researchers attempted to solve many realistic requirements in the recent years. In [13] [14] [15] for an example, they are analyzed the entire issues of sequential patterns mining along with their quantities. Most of the real-time applications contain the entire size information which is obtained in data. Moreover, this information may be neglected by many more existing algorithms.

3 Proposed System

For the purpose of sequential pattern mining, an effective algorithm known as Hybrid Pattern Technique is proposed in this paper. In the frequent sequential pattern among two consecutive events, not exactly the time-period even its possibilities are revealed in mining process. It considers the space limitation and the time-period among two consecutive items or events because of every infinite sequence itemsets occurs in the sequence database.

Notations

In this research, a sequence of items is described as an ordered itemsets in list manner also it is indicated as (I_1, I_2, in). The parentheses '('and ')' are used to enclose the itemset which are from the same items. Consider an example, {(2,3),1,4,(1 ,5, 6), 3} is the sequence of items then it also have the following itemsets:{2,3*}*, *{*1*}*, *{*4*}*, *{*1,5,6*}*, and *{*3*}*. As a result, if one item is present in the item set then the parentheses can be neglected. Also, in this research, an itemset is indicated as flowing like the every item is combined with a transaction time and the frequent items also occurred in the similar transaction time. A item sequence is indicated as (s_1,t_1), (s_2,t_2), . . ., (s_n, t_n), where s_j is a item and t_j indicates the transaction time of s_j occurs, $1 \leq j \leq n$, and $t_1 \leq t_2 \leq \cdots \leq t_n$. Also, a sequential pattern can be represented as $_(\omega) = (\omega_1, \omega_2, \omega_n)$. The following definitions are described with help of the above mentioned representation,

Definition-1: A sequence is S= {[s_1, t_{s1}], [s_2, t_{s2}]… [s_n, t_{sn}]}, and a pattern P = {p1, p2, pm}, p is time-rest prefix of S and m≤n

Definition-2: A sequence database is DS= {S_1, S_2… S_∞} i.e. a sequential database having

Definition-3: A time-rest subsequence P′ of sequence P is indicates a prefix or projection of P w.r.t the time-rest prefix R if (a) P′ has time-rest prefix R and (b) there presents not in proper time-rest super sequence P″ of P′ such that P′ is a time-rest subsequence of P and has a time-rest prefix R. If one eliminates direct manner, the time-rest prefix R from the projection P′, the new sequence is called the postfix of P with respect to time-rest prefix R.

Definition 4: Let T^{Last} indicates the final transaction occurrence of a sequence. Database DS. S = {[s_1, t_{s1}], [s_2, t_{s2}] . . . [s_n, t_{sn}]} is a sequence in DS. *{*S (t_1, t2)*}* indicates the entire items that represent as sequence S which are occur among transaction time t_1 and t_2. Assume that, a time-rest subsequence of S is represented as P = {[p1, tp1], [p_2, tp_2, [p_m, t_{pm}]} and χ is a one of the item that do not depended on {S (t_{pm},t_{qn})*}*. The correct transaction time of χ does not know to us based on the present sequence S. The probable censoring time of item χ w.r.t pattern p which is described as T^{Last} – tpm to realize the time interval probability of two frequent items.

Table 1. Notations utilized in Hybrid algorithm

Parameter	Expansion
<ω>	A time-rest prefix
|(ω)|	Length <ω>
DS_{exis}	Continuous sequence database
DS_{new}	Divided Database **of** DS_{exist}
$DS\|_{<\omega>}$	Projected Database of DS_{new} w.r.t <ω>
μ	Least-Prop-Threshold
T_p	Time Duration
Þ	Least probability threshold
Min_s	Minimum space threshold
Max_s	Maximum space threshold

3.1 Hybrid Mining Algorithm

Space Constraints

The space constraint is applied on the item sequence id which is from sequential consecutive pattern. It remove out the sequence id of frequent pattern then it attempt to discover the given minima and maxima sequence id based frequent pattern. Consider an example, 0 to 10 sequential id which is sequential database and then specify lower space 1 and maximum space is 5. Then they obtained pattern can be be 012345 sequence id.

Time Constraints

A time constraint is work on the sequential frequent pattern i.e. time constraints remove the unnecessary subsequence

Pattern from the extracted subsequence patterns based on Least-Prop-Threshold and time probability threshold. The following steps are used to extract the feasible patterns from the sequential database.

3.1.1 Steps for Hybrid Mining Process

Step 1: Divide the sequential database space using minima and maxima threshold values
Step 2: Scan the sequential database for generating frequent itemsets
Step 3: Evaluate the arrival rate between each items
Step 4: Evaluate the Bury-Arrival time probability for frequent items at each time t.
Step 5: The Bury-arrival probability is greater than the time probability threshold, append the frequent item into pattern.
Step 6: Repeat the step-2 to 5, until obtain the specified reliable patterns

3.1.2 Hybrid Mining Algorithm

Consider a sequential database DS and pattern pattern (ω), we use projected or prefix appended database DS | (ω)| to indicate the list of postfixes in DS w.r.t pattern (ω). The quantity of items represents the size of pattern (ω) which is indicated | (ω)|. The following pseudo code has represents our proposed hybrid pattern mining algorithm.

Algorithm: Hybrid Algorithm
Input: $DS_{exist,}$ DS_{new}, μ, $T_{p,}$ Þ, $Min_{s,}$ $Max_{s;}$
Output: A set of feasible frequent patterns
Iteration: Hybrid (<ω>, |(ω)|, $DS|_{<\omega>}$)
1. Sequence count= $Min_{s;}$
2. For (sequence_count; sequence_count<$Max_{s,}$ sequence_count++)
3. {
4. Select sequence pattern from DS_{exist}
5. Insert sequence pattern into DS_{new}
6.}
//end for
7. Scan $DS|_{<\omega>}$ one time, Extract all frequent items
8. Total_item=All_items_present in DS
9. if (|(ω)|=0)
10. {
11. Every frequent item χ, combine χ to <ω> as <ω′>
12. }
//end if
13. if(|(ω)|>0)
14. {
15. for (χ;χ≤ Total_item; χ++)
16. δ=**arrival_rate(<ω>,$DS|_{<\omega>}$,χ)**
17. BATP=1-$e^{\delta td}$
BATP- Burry-Arrival Time Probability
18. if(BATP> Þ)
19. {
20. Combine χ to <ω> as <ω′>
21 }
//end if
22. }

end for
23. for (Until all <ω′>)
24. {
25. Construct Projected Database $DS|_{<\omega'>}$ w.r.t <ω′>
26. Recall Hybrid (<ω′>, |(ω′)|, $DS|_{<\omega'>}$)
27. }

Iteration: Arrival rate(<ω>, $DS|_{<\omega>,\chi}$)
Input: <ω>, $DS|_{<\omega>,\chi}$
Output: δ-arrival rate of item χ occurrence after final item in <ω> χ-frequent item
1. tf= the transaction time of final item in <ω>
2. r=0;
r-total number of sub sequence
3. α_1=0;
α_1- Difference between transaction time of final time in (ω) and initial item χ.
4. α_2=0;
α_{2-} the censoring time of censored item χ
5. for (every postfix in $DS|_{<\omega>}$)
6. if (χ ∉ items in present postfix)
7. {
8. $\alpha_{2=}\alpha_2$+ (Transaction time of Final occurrence item-Transaction time of initial occurrence item)
10. }
11. else
12. {
13. r=r+1;
14. t_χ =transaction time of item χ
$\alpha_1=\alpha_1+(t_{\chi-}$ Transaction time of initial occurrence next item)
15. } //end if
16. } //end for
17. δ= $r/(\alpha_1+\alpha_2)$
18. return δ;

4 Performance Evaluation and Experimental Results

Hybrid Algorithm is implemented in the Java language and also it is tested with AMD Sempron (tm) 1.60 GHz CPU, MS Windows XP operating system and 1GB memory on a computer system in order to calculate the performance of the proposed work. To store the datasets My SQL server 2005 is utilized.

The following are the threshold values that are applied in the Hybrid Technique.

1. Minimum space =1
2. Maximum space =4
3. Least-Prop-Threshold =2
4. Least Time Probability=0.3
5. Expected Time Period =7 days

Table 2 represents the infinite datasets which are occurred in the Sequential Database. Table 3 represents the total number of patterns, arrival rate, Bury-arrival rate and selected frequent patterns which are established by the Proposed Hybrid algorithm. This proposed Hybrid algorithm provides better performance than the other various pattern mining algorithms like a-priori, FP-tree, Prefix-Span and P-Prefix-Span. Table 3 represents the performance of Proposed and Existing approach.

Table 2. ∞ - Sequential Pattern Database

Sequence Id	Datasets
0	{(4,12),(3,4),(5,6),(2,9)}
1	{(2,2),(3,2),(1,4),(4,7),(1,8),(3,15)}
2	{(1,1),(4,5),(6,5),(2,12)}
3	{(1,1),(2,1),(3,1),(6,4),(6,6),(2,8),(3,9)}
4	{(3,5),(1,7),(2,15),(4,18),(6,18)}
5	{(3,5,(4,6),(6,7)}
∞	….

Table 3. Extracted Patterns and their Burry Arrival Time Probability

Patter n	Arrival rate	Bury-Arrival Time Probability	Patterns Extraction
{1,1}	0.033	0.208	-
{1,2}	0.087	0.456	Extract
{1,3}	0.072	0.399	Extract
{1,4}	0.045	0.273	-
{1,6}	0.142	0.632	Extract

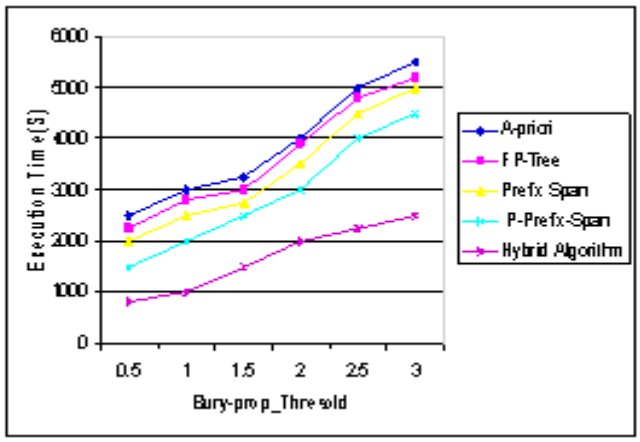

Fig. 1. Least-Prop-Threshold Vs Execution Time

Figure 1 represents the entire relationship among the Least-Prop Threshold of pattern generation and also their execution time. When the threshold value is increased, then the execution time is also increased. It means that the number of items mutation is increased. When compared with other existing algorithm, the proposed approach takes very less execution time.

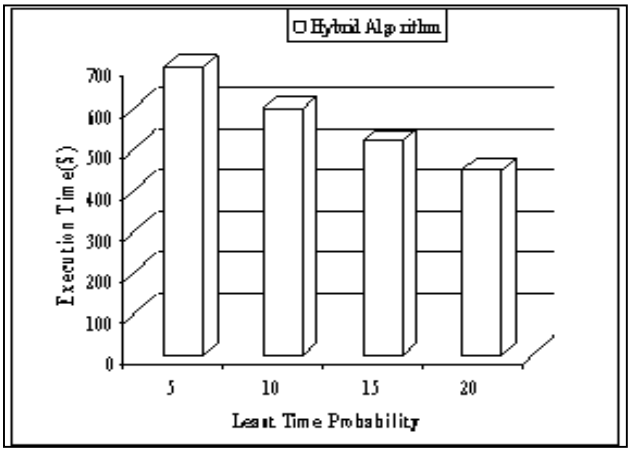

Fig. 2. Least-Time-Threshold Vs Execution Time

Figure 2 represents the relationship of proposed approach among the least time probability threshold of pattern extraction and also their execution time. The high probability can be reduced the execution time of pattern generation because of the generated Bury-prop-arrival rate is greater than the least time probability.

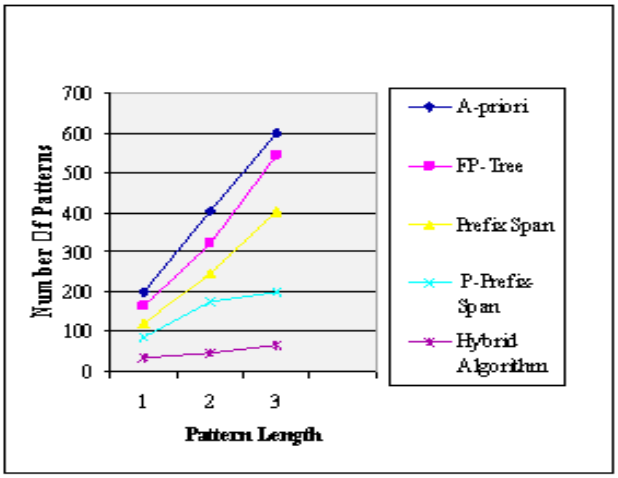

Fig. 3. Comparison of pattern generation Hybrid and Other Algorithm

Figure 3 represents the number of pattern generation among the proposed Hybrid Algorithm and other existing Algorithm. The proposed Hybrid Algorithm creates very limited amount of frequent patterns than other existing algorithms. If the pattern length increases, then the amount of patterns is also increased Figure 4 represents the execution time of Proposed Hybrid Algorithm and other existing algorithms.

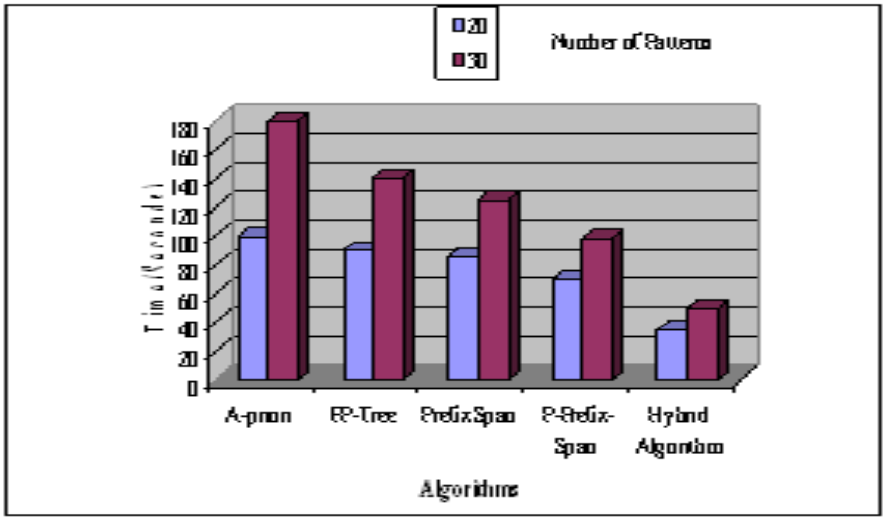

Fig. 4. Comparison of Time execution of Hybrid and Other Algorithm

The proposed Hybrid Algorithm takes very less time when compared to other algorithm like A-priori, FP-tree, Prefix Span and P-Prefix-Span. If the number of pattern increase then the time also increases as the number of patterns is propositional to time. The performance of proposed Hybrid Algorithm is very proficient in computation speed, storage space and time.

5 Conclusion

In sequential pattern mining, many researches focuses only on the Symbolic Patterns where the numerical investigation is primarily belongs to the range of forecasting investigation and trend analysis. A latest algorithm namely called Hybrid Algorithm is proposed in this paper which is mainly utilized for extracting realistic sequential patterns where it focus on symbolic patterns at the same time it also focus on the numerical analysis. As well as this algorithm uses minimum and maximum space of sequence database where it can increases the accuracy of final output. The reliable sequential pattern not only yields the information of ordered frequent items, it also obtains the detail probability of arrival items time. This information shows the brief explanation of the derived patterns characteristics which are very critical for the decision makers.

The users can identify the Least-Prop-Threshold and Least time probability threshold to establish the feasible sequential patterns as per the algorithm. The proposed algorithm minimizes candidate patterns amount with help of the threshold of least time probability which makes the proposed technique is greater than the other mining algorithms.

The experimental results represents that the Proposed Hybrid algorithm is very effective and very suitable technique for mining the sequential pattern. Thereafter, the proposed approach can also be utilized with more constraints for extracting the feasible patterns.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: 20th VLDB Conference, Chile, pp. 487–499 (1994)
2. Agrawal, R., Srikant, R.: Mining sequential patterns. In: 11th International Conference on Data Engineering, Taiwan, pp. 3–14 (1995)
3. Ayres, J., Flannick, J., Gehrke, J., And Yiu, T.: Sequential pattern mining using a bitmap representation. In: 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 429–435. ACM, New York (2002)
4. Chen, Y.L., Chiang, M.C., Ko, M.T.: Discovering time-interval sequential patterns in sequence databases. Expert Systems with Applications 25, 343–354 (2003)
5. Chen, Q., Dayal, U., Han, J., Hsu, M.C., Mortazavi-Asl, B., Pei, J.: FreeSpan: frequent pattern-projected sequential pattern mining. In: International Conference on Knowledge Discovery and Data Mining, USA, pp. 355–359 (2000)
6. Chen, Q., Dayal, U., Han, J., Hsu, M.C., Mortazavi-Asl, B., Pei, J., Pinto, H., Wang, J.: Mining sequential patterns by pattern-growth: the prefixspan approach. IEEE Transactions on Knowledge and Data Engineering 16, 1424–1440 (2004)
7. Jou, C., Shyur, H.-J., Chang, K.: A data mining approach to discovering reliable sequential patterns. The Journal of Systems and Software 86, 2196–2203 (2013)
8. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Proceedings of ACM SIGMOD International Conference of Data, Dallas, TX, pp. 1–12 (2000)
9. Pei, J., Han, J., Pinto, H.: PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. In: 11th IEEE International conference on Data Engineering, Germany, pp. 215–224 (2001)
10. Kim, C., Lim, J.H., Ng, R.T., Shim, K.: SQUIRE: sequential pattern mining with quantities. Journal of Systems and Software 80, 1726–1745 (2007)
11. Orlando, S., Perego, R., Silvestri, C.: A new algorithm for gap constrained sequence mining. In: ACM Symposium on Applied Computing, New York, pp. 540–547 (2004)
12. Toroslu, I.H.: Repetition support and mining cyclic patterns. Expert Systems with Applications 25, 303–311 (2003)
13. Zaki, M.: SPADE: An efficient algorithm for mining frequent sequences. Machine Learning 42(2), 31–60 (2001)
14. Kuthadi, V.M.: A new data stream mining algorithm for interestingness-rich association rules. Journal of Computer Information Systems 53(3), 14–27 (2013)
15. Selvaraj, R., Kuthadi, V.M.: A modified hiding high utility item first algorithm with item selector for hiding sensitive item sets. International Journal of Innovative Computing, Information and Control 9(12), 4851–4862 (2013)