ISSN:0975-9646

A Survey on Sequential Pattern Mining Algorithms

Vishal S. Motegaonkar, Prof. Madhav V. Vaidya

Department of Information Technology,
SGGS IE & T, Nanded, Maharashtra, India-431606

***Abstract*- Sequential pattern mining is a very important mining technique with broad applications. It found very useful in various domain like natural disaster, sales record analysis, marketing strategy, shopping sequences , medical treatment and DNA sequences etc. It discovers the subsequence's and frequent relevant pattern from the given sequences. That we have provided the sequence database having sequences, in which each sequence is a list of the transactions ordered by the transaction time. Each transaction consists of the number of the items. The problem is to discover the all sequential pattern who satisfy the user specified constraint, from the given sequence database. There are various sequential pattern mining algorithms proposed earlier, some of them are GSP, SPADE, SPAM and Prefixspan. They are proposed to find the relevant frequent pattern from the sequences. In above algorithms the old dataset is deleted while some other dataset are updated. In these algorithm the timestamp is an important attribute of each dataset, and also it is important in the process of data mining for giving the more accurate and useful information. The detail survey of these entire algorithms is presented in this paper. The survey of all these algorithms done with various research perspectives. First we categorized these algorithms by their used approaches to solve the mining problem and then we have compared each one with another by their various provided features and performance point of view.**

***Keywords*- Sequential Pattern Mining, Sequence Database, Apriori, GSP, SPADE, Prefixspan.**

I. INTRODUCTION

The sequential pattern mining is a very important concept of data mining, a further extension to the concept of association rule mining [1]. That has a wide range of real-life application. This mining algorithm solves the problem of discovering the presence of frequent sequences in the given database [2]. The database given to this algorithm is set of sequences called as data-sequences. Each data-sequence is a list of customer transaction, and each transaction is a set of items. There is transaction time which is associated with the each transaction in the sequence database. The sequential pattern mining is almost similar to the association rule mining, but the difference is that the events are linked with time. The sequential pattern mining discovers the correlation between the different transactions, but in the case of association rule mining it discovers the relationship of items in the same transaction.

In case of association rule mining, it discovers that which different items are brought with each other frequently, all these items must have come under same transaction. Instead in case of sequential pattern mining, it discovers which items are brought in a particular order by a single customer, having those items come from various transactions. The sequential pattern mining is very useful for the marketing person to determine which item is brought one after another in sequence by particular customer. Sequential Pattern Mining is defined as discovering the whole set of frequent subsequence in the set of sequential transactional database. The resulting pattern found after mining is the sequence of item sets that normally found frequent in specific order. In a single transaction all items have the same transaction time value. Each sequence is the ordered list of the different transactions and every transaction in it is a collection of the items. The ordering of the transaction in a sequence is induced by the absolute timestamps associated with that transaction. Generally all the events of a customer are together viewed as sequence, known as customer-sequence, where as each event is presented in the form of item set in that sequence and all the events are listed in a specific order with reference to the event-time. The process of finding sequential pattern from sequence transaction database is described below-

Problem Definition: Let E be set of customer transaction in which every transaction T have customer_id, a time at which transaction takes place and a set of items involved in the transaction. Let A={i1,i2,.....,ij} be a set of items. An item set A is a non-empty set. A sequence s is set of item sets and ordered it according to time-stamp associated with them. The sequence s is denoted as <s1,s2,.....,sl>, where sk, k€1.....l, an item set. The j-sequence is the sequence having j-items (of length j). The sequence <s1,s2,....,sm> is the sub-sequence of other sequence <s'1,s'2,.....,s'n> if there is integer i1<i2<.....<ij.....<im such that the s1⊆s'1, s2⊆s'2,.....,sm⊆sn' [3]. The problem of discovering the sequential pattern is to find all that sequences s such that support(s)≥min_support for database E, where the support threshold value is min_support.

The task of finding all frequent sequence in huge database is more challenging because the search space is large. For instance, with the m attribute there may be $O(m^n)$ frequent sequences of length n. The things that the responsible for the sequential pattern mining algorithm so difficult and time-consuming one are as follows. First is that the information of a pattern is not just related to single item but to the item sets. Second, pattern can be formed by any permutation, of any possible combination items in the sequence database. Third, the number of item sets in a pattern and the number of items in the item set is unknown prior to the mining. The sequential pattern mining concept is introduced 1995 by Agrawal et al [2]. The problem of finding the sequential pattern has taken more attention. After their work, there have done many studies on the sequential pattern mining and their applications, and the

efficiency and accuracy of mining the entire sets of sequential pattern is improved too much till date. In many situation sequential pattern mining still have some challenges in both effectiveness and performance. On the other side, there may be a huge number of sequence patterns in large database. A user is often interested in only small patterns. To present the whole set of sequential pattern might be cause user hard to interpret and hard to use.

II. LITERATURE REVIEW

In the sequential pattern mining concept there various study and proposal presented in literature till date. In which some are constraint based sequence pattern mining and some are incremental sequential pattern mining. The study and review on some latest researches related to the incremental sequential pattern mining is presented below. In the past times, the improving the strategy and concept for incremental mining of constraint-based pattern mining has comes as very important issue for day to day life application.

Ching-Yao Wang [5] has proposed an algorithm for sequential pattern mining based on the incremental mining concept. This algorithm uses the concept of Pre-Large sequence to minimize the need for rescanning the original databases. By applying the lower support threshold and upper support threshold it defines the Pre-Large sequence that act as gap to resist the movement of sequence from large to small and from small to large. This algorithm does not perform the rescanning of the database until the new customer sequence is added. That is when database size get larger, the number of new transactions allowed before the database rescanning required also grow.

Chi-Yao Tseng [6] have proposed general model for sequential pattern with the changing database, while the data in the database can be fixed, added or removed. Also they presented the progressive algorithm called PISA which is stands for Progressive mIning of Sequential pAttern which discover the sequential pattern in fixed time interest in progressive manner. The period of interest is the time period continuously moving forward with time goes by. In PISA algorithm, to efficiently maintain the recent data sequences it utilizes a progressive sequence tree. It finds outs the whole set of up-to-date sequential pattern and remove obsolete data and pattern as per require. The size of the sequential pattern tree created was depending on the length of the period of the time window. So that effectively minimizing the memory required by algorithm that is very less than the memory required by other methods.

Jiaxin Liu [4] have proposed a data storage structure, known as frequency sequence tree, and gives the generation method for the frequent sequence tree called con FST. At the root node of this frequent sequence tree stored the support for frequent sequence tree and the path from the node to the any outer node represents a sequential pattern in the database. The sequential pattern whose support meets the frequent sequence tree support threshold is stored in frequent sequence tree, so as the support changed, the algorithm which uses FST as the storage structure could find the entire sequential pattern without mining the entire original database. Vincent Shin-Mu Tseng [7] have proposed the rule growth, the method for mining the sequential rules same for several sequences. Unlike the other algorithms rule growth is based on the pattern-growth approach for finding sequential pattern rules such that it can be better and scalable. They performed test of the rule growth with other some algorithm on the public datasets. It found that the rule growth clearly outperforms the other algorithms, for these datasets under low support and fixed threshold.

Jiaxin Liu [8] have proposed that the structure of sequence tree based on the projected database, known as sequence tree, for the construction of this sequence tree they proposes the steeps algorithm. Sequence Tree was the structure of data storage. It is similar in structure to the prefix tree. But, it stores all the sequence in the original database. The path from the root node to any leaf node is a sequence in the database. The structural characteristics of the sequence tree make it suitable for the increment pattern mining. From the experimental analysis showed that the increment mining method of sequential pattern which uses the sequence tree as the storage structure for sequence pattern performed best than the prefixspan in memory use cost on condition that support threshold was smaller. To take the dynamic nature of data addition and deletion.

III. COMPARISON OF DIFFERENT EXISTING SEQUENTIAL PATTERN MINING ALGORITHMS

In recent years many approaches in sequential pattern mining have been proposed, these studies cover broad portion of issues [11]. In general there are two important concerns in sequential pattern mining.

(1) The first one and very important one is to improve the performance or efficiency and accuracy in sequential pattern mining process.
(2) Extend the mining of sequential pattern to the time related constraint.

A. Improve the Performance by designing suitable algorithms.

As per the research done till date on the sequential pattern mining, the algorithms differs in two categories [12].

(1) The way by which candidate sequences are created and stored in memory. The important target for this category of algorithms is to minimize the number of candidate sequences generated so that to minimize the IO cost.
(2) The way by which the support value is calculated and how the candidate sequences are tested by using these support value for frequency. The main idea here is to delete the any database record or data structure that has to be maintained over the time of support of counting purpose only. On the basis of these criteria's sequential pattern mining is classified broadly into two groups:
    - Apriori Based.
    - Pattern Growth Based.

a) Apriori Based Algorithms

The Apriori and AprioriAll algorithms set the basic for a set of algorithms that depends largely on the appriori property and use the appriori- generate joint procedure to generate the candidate sequences. As per the apriori statement property all the nonempty subset from the frequent item set much also be the frequent. That is also be referred as (downward-closed) in that if a sequence cannot satisfy the minimum support test, than its entire super sequence will also fail the test/condition.

Important terms of the apriori -based algorithms are [12]:

1) *Breadth-first search technique used*: Basically the apriori based algorithms are work on the breadth-first search technique (level-wise), because the sequences, in jth iteration of the algorithm as they traverse the search space.
2) *Generate-and-Test*: This kind of feature is used by the very early algorithms from initials research done in sequential pattern mining algorithms which rely on this technique only shows the inefficient pruning method and create huge number of candidate sequences and then test each one sequentially for satisfying some user specified constraints consuming a lot of memory in the early stage of mining.
3) *Multiple scan of the database*: This feature is very undesirable because it requires the lots of processing time and IO cost.

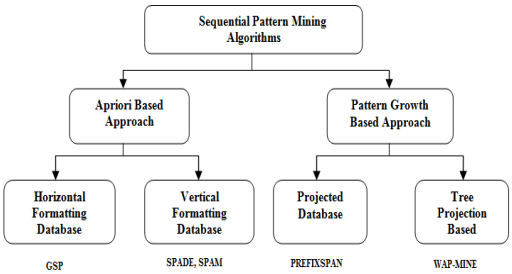

Fig- Classification of Sequential Pattern Mining Algorithm

GSP (Generalized Sequential Pattern)- algorithms is described by Agrawal and Shrikant [14] makes the multiple passes on the data. This algorithm is more faster than the AprioriAll algorithm. In the GSP algorithm the two steps are involved, one is candidate generation and candidate pruning method. The algorithm is not a main memory algorithm generates only as many candidates as will fit in memory and the support of the candidate is find out by scanning the dataset. Frequent Sequences from these candidates are written to disk and the candidates which are without minimum support are deleted. The same step is repeated until every candidate has been counted. The GSP algorithm finds all the length-1 candidates (using one database scan) and orders them by their support value ignoring whose support<min_support. Then for each level (i.e. sequences of length k) the algorithm scans the dataset to collect the support count of the each candidates sequence and generates candidates of length (k+1) sequence from length-K frequent sequences using apriori. This step is continued until no frequent sequence or no candidates can be found.

This algorithm has a very good scale up properties with respect to the number of transaction per data sequence and number of items per transaction. But this algorithm is less than efficient where the mining in large sequencing of databases having numerous pattern or long patterns as it cannot generates any more candidates sequence and also multiple scans of database is needed because the length of each candidates grows by one at each database scan.

SPIRIT - The basic concept behind this algorithm is to use the regular expression at flexible tool for the constraint specifications [13]. It provides the generic user specified regular expression constraint on the mined pattern, for providing the more powerful restriction. There are many versions in the algorithm. The selection of the regular expression as a constraint specification tool is considered on the basic of two important factors. The first regular expression is the simple form and natural syntax for specification of families of sequential pattern and second it has the more power for specifying huge range of interesting pattern constraints.

SPADE - As like horizontal formulating methods (GSP) the sequential dataset can be transformed into a vertical dataset format consisting of item id-lists [15]. 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 a 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.

SPAM - SPAM combines the ideas of GSP, SPADE, and FreeSpan [16]. This algorithm uses the vertical bitmap data structure representation of database which is similar to the given id-list of SPADE. The whole algorithm with its data structure fits in the main memory. For the performance increase the SPAM use the depth-first traversal fashion. SPAM is similar to SPADE, but it uses the bitwise operations instead of the regular and temporal join when the comparison of SPAM and SPADE is consider the SPAM outperform more than SPADE, while the SPADE algorithm is more SPACE-efficient than SPAM.

CloSpan- CloSpan (Closed Sequential Pattern Mining) algorithm mines the frequent closed sub sequences only [16]. That is, those containing no super-sequences with the same support when mining long frequent sequence. The performance of algorithms degrades dramatically. This algorithm creates the less number of sequences than the other algorithms.

CMDS - (Closed Multidimensional Pattern Mining) is an combines method of closed- item set pattern mining and closed sequential pattern mining [17]. It consist of mainly two steps-

- Combination of closed sequential pattern mining with closed item set pattern mining.
- Removal of redundant pattern.

The number of pattern in CMDS is less than the number of pattern in multidimensional pattern mining. The set of CMDS pattern can cover the set of MDS pattern.

b) Pattern-growth Sequential Pattern Mining Algorithms

The Pattern Growth algorithm comes in the early 2000s, for the solution to the problem of generates and test. The main concept is for to avoid the candidate generation step altogether, and to concentrate the search on a specified portion of the initial database. In this kind of the algorithm the technique of search space partitioning is an important role in pattern-growth. In this kind of algorithm initiates by building a representation of the database to be mined, and after that defines the way to partition the search space and generates the candidates' sequences by growing on the initially mined frequent sequences. The initial algorithm started by using projected databases, which is free-span, prefix span with latter one being most influential.

PrefixSpan- The PrefixSpan (Prefix Projected Sequential pattern Mining) algorithms presented by Jian Pei, Jiavei Han and Helen Pinto [19] is the only projection based algorithms from all the sequencing pattern mining algorithms. It performs better than the algorithm like apriori, freespan, SPADE (vertical data format). This algorithm finds the frequent items by scanning the sequence database once. The database is projected into several smaller databases according to the frequent items. By recursively growing subsequence fragment in every projected database, we got the complete set of sequential pattern. The main concept behind the prefixspan algorithm to successfully discovered patterns is employing the divide-and-conquer strategy. The prefixspan algorithm requires high memory space as compare to the other algorithms in the sense that it requires creation and processing of huge number of projected sub-databases.

FREESPAN- The freespan algorithm reduces the cost require to candidate generation and testing of apriori, with satisfying its basic feature [18]. In short, the freespan algorithm uses the frequent items to iteratively project the sequence database into projected database while growing subsequence's frequently in each projected dataset. Every projection divides the database and confines further testing to progressively smaller and more manageable units. The important issue is to considerable amount of sequences can appear in more than single projected database and the size of database decreases with each iteration.

WAP-MINE- This is pattern-growth based algorithm with tree-structure mining technique on its WAP-tree data structure. In this algorithm the sequence database is scanned twice to build up the WAP-tree from the frequent sequences by their support values. Here header table is maintained first to point that where is first occurrence of the each item in a frequent item set which can be helpful to mine the tree for frequent sequences built up on their suffix. It found in the analysis that the WAP-MINE algorithm have more scalability than GSP and perform bitterly by marginal points. Although this algorithm scans the database twice only and avoids the problem of generating huge candidate as in case of apriori-based approach, the WAP-MINE faces the problem of memory consumption, as it iteratively regenerate n increase automatically.

B. Extension to the Time-Related pattern on Sequential Pattern Mining

In recent years the sequential pattern mining has been intensively studied. There exists a large verity of algorithms for sequential pattern mining, along with that motivated by the various applications for sequential pattern. Following extension of the initial definition have been proposed which may be related to other types of time-related patterns or to the addition of time constraint. Some extension of those algorithms for special need such as time interval, multidimensional, constraint based and closed sequential pattern mining are studied in following section.

a) Developing the Sequential Pattern based on the Constraints

Even the efficiency of the finding the whole set of sequential pattern is improved drastically; in many situation the sequential pattern mining is facing tough challenge in both correctness and the performance. Instead, there could be a huge number of sequential patterns in large database. That the user is actually interested in only a small subset of such patterns. It may hard to interpret or hard to use the result when to presenting the complete set of sequential pattern. To address this problem Jiawei Han and Wei Wang [20] have systematically presented the method of adding different rules deep into sequential pattern mining known as constraints using pattern-growth methods. As the constraints usually represent user's interest and focus, this approach may overcome the obstacle of effectiveness and efficiency of sequential pattern mining. The user defined constraint will limits the pattern to be found to a particular subset satisfying some strong condition. The pei han and Wang mention the seven categories of constraints.

1) **User Defined Item Constraint:** This user defined constraint specifies the subset of item that should not be present in the resultant pattern.
2) **User Defined Super Pattern Constraint:** The super patterns are once those contain at least one of particular set of pattern as sub-pattern.
3) **Defined Regular Expression Constraint:** The regular expression constraint CRE is a constraint specified regular expression on the set of item with the help of established set of operators like disjunction, union and kleens closure.
4) **User Defined Gap Constrain:** This kind of constraint is defined only for sequence databases in which each transaction in every sequence has a timestamps. According to gap constraint the sequence pattern in the sequential database must have the property such that the timestamps difference between every two adjacent transaction must be shorter or longer than the user defines gap.
5) **User Defined Length Constraints:** User can specify the length of the pattern where length can be in the form of number of occurrence of item or the number of the transactions.
6) **User Defined Aggregate Constraints:** This user defined constraint is on an aggregate of items in a pattern, where as the aggregate function may be sum, max, min, avg, standard deviation etc.

7) **User Defined Duration Constraints:** The duration constraint defined by user is can only apply to the sequence databases where each transaction in every sequence has a timestamps. According to the duration constraint it requires that the sequential pattern in the database should have the property such that the timestamps difference between initial and final transaction in a sequential pattern must be shorter or larger than given period.

IV. COMPARATIVE ANALYSIS OF SEQUENTIAL PATTERN MINING ALGORITHMS

This survey of the sequential pattern mining algorithm is completed on the basis of their various important features. For the comparison sequential pattern mining algorithm is categorized into two broad categories, as apriori based and pattern growth based algorithm. The all features used to classify these algorithms are discussed first and then comparison is done for the following algorithms.

G.S.P.: Generalized sequential pattern.

SPADE: Use of the equivalence classes for the discover of the sequential pattern.

SPAM: Sequential Pattern Mining.

Freespan: Finding the sequential pattern by projecting the frequent pattern in sequence database.

PrefixSpan: By prefix-projected sequential pattern mining.

WAPMINE: Web access pattern mining from sequential dataset which contains web click in the sequential format by timestamps.

SPIRIT: By formulating the constraint using regular expression the sequential pattern mining.

A. Features of Sequential Pattern Mining Algorithms are:

1) *Breadth-First Search Based Approach vs Depth First Search Based Approach:* In the breadth-first search traversal technique level-by-level search is conducted to find the complete set of pattern i.e. All the inner node are processed before moving to the next level. Instead in the depth-first search traversal technique, all the inner-node must be explored before in the path moving to the next one. The depth first search is that it can reach very quickly to large frequent fragments and therefore some expansion in the other path in the tree can be avoided.
2) *Apriori-Based vs Pattern-growth Based:* In the category of apriori based type algorithm the main theme is to candidate-generate and test which uses the downward closer property. If an item set α is frequent, then and then only the superset of α is frequent, otherwise if not be frequent either. Pattern-growth strategy takes better approach in creating possible frequent sequences, and uses the divide-and-conquer approach. For the reduce of search space this pattern-growth algorithm do the projection on the database.
3) *Top-Down Search vs Bottom-up search:* The apriori based algorithms uses a bottom-up search by ensuring each single frequent sequence. It means that for the produce a frequent sequence of length 1, all 21 subsequence's have to be generated. From that it can be stated that this exponential complexity is limiting at the apriori based algorithms to find out only short pattern, since they just find the subset infrequent pruning by deleting any candidate sequence for which there exist a subsequence that does not belongs to the set of frequent sequences. In case of the top-down approach the subset of sequential pattern can be mined by generating the relative set of projected databases and mining each recursively for top to bottom.
4) *Anti-Monotone vs. Prefix-Monotone Property:* According to the property of anti-monotone it states that the each non-empty subsequence of the sequential pattern is a sequential pattern. And in the prefix-monotone states that every sequence which is having α as a prefix satisfies the constraints if α sequence satisfy the constraint.
5) *Regular Expression Constraints:* The number of state changes in the relative deterministic finite automata help to calculate the complexity of regular expression constraints. It has the nice property known as growth-based anti-monotonic if it satisfy the following property. The sequence must be reachable by growing from any component which matches the part of the regular expression when it satisfies the constraints first. From our comparative study we found that prefixspan algorithm uses depth-first search based approach. Top-down technique is efficient technique to find frequent subsequence's as sequential pattern from the large database. Also the regular expression constraint and prefix monotone property is use by prefixspan algorithm, which makes this algorithm best choice for applying user defined constraint for mining only concerned sequential pattern.

B. EXPERIMENTAL STUDY DONE BY RESEARCHERS

To analyse the correctness and performance of various sequential pattern mining algorithm, a performance study is done on the four algorithms, on GSP, Freespan, prefixspan and SPADE. On real and synthetic datasets.

Dataset used for the Analysis

Synthetic dataset is actually used for the performance study. The synthetic dataset we have use in the analysis study are C20T16S818, C100T1.25SIo.75 and C100T2.5S5I1.25 where-

C=Number of customer.
T= Average number of items per transactions.
S= Average number of transactions per sequence
I= Average item set in maximum sequence.

That we have assumed that number of items is 10000 and on average a frequent sequence pattern contains as many four transactions.

1) *Analysis on the memory uses by algorithms:* From the experimental result, it is found that the prefixspan is better stable in memory usage than the other algorithms SPADE and GSP. At the support value of 0.25 present, the SPADE enters in the crash or error message while GSP use about 375 MB memories, while the prefixspan only uses less than the one third of the memory used by

the GSP algorithm. In the analysis done by the researchers it found that the prefixspan algorithm needs memory space to just hold the sequence datasets and a set of header tables and pseudo projection table.

2) *Comparison on the basis of time complexity of the Algorithms:* From the survey of all these three algorithms it is found by the experiments that the both of the pattern growth algorithm Freespan and prefixspan are time efficient than apriori based algorithms.

3) *Comparison on the basis of ability to scale-up property:* By the survey done by researchers it is found that prefixspan has better performance than the other algorithms and it scales with the database size linearly.

The SPADE and GSP algorithms needs memory space to hold the candidate sequence pattern as well as the sequence database instead for the case of prefixspan it needs memory space to fit the sequence database and a set of header table and pseudo projection table. From the above performance analysis it is found that the prefixspan is the best among all the other tested algorithms. The prefixspan performs better than the other algorithms that the reasons discussed below –

a) *Use of Pattern-growth approach without candidate generation:* As like the traditional apriori based approach in which generates the candidate and test is used, the prefixspan does not perform the candidate generation and test. It only calculates the frequency of local 1-itemset.

b) *Partitioning-based approach as best mean for data reduction:* The prefixspan algorithm creates the longer sequential pattern from the smaller or shorter one by partitioning the search space and concentrating only on the subspace after supporting the pattern-growth. The search space of this algorithm is concentrate and continued to only a set of projected databases. So the projected dataset for the subsequences α contains all and only the required information for mining the super pattern that can grow up from α. The size of the projected database goes on decreasing as per the mining of longer sequential patterns. In other hand the algorithm which is based on the apriori algorithms works on the entire or whole database once for all iterations during the mining process. Many insignificant results have to be examined and checked which leads to increase the overhead; this may results in the performance degradation.

c) *Prefixspan requires comparatively stable and less memory space:* For the algorithm based on the apriori approach they require the candidate generation and test method as well. For both GSP and SPADE requires a huge amount of memory when the support qualify value goes low, since it need to hold a huge number of candidate sets. Instead for the prefixspan, it doesn't generate any candidates and explores the divide-and-conquer methodology, so it requires the constant memory space over the mining process.

V. CONCLUSIONS

In this paper, we discussed what is sequential pattern mining and various types of their algorithms. This concept is being introduced in 1995, has gone through remarkable advancement in few years only. Initial work on this topic is concentrated on improvement of the performance of algorithms by using different data structure or different representation. So, on the basis of these problems the sequential pattern mining is categorized into two main groups, Apriori approach based algorithms and pattern growth approach based algorithms. From our comparative survey and previous some studies by various researchers on sequential pattern mining algorithms it is found that the algorithm which are based on the approach of pattern growth are better in terms of scalability, time-complexity and space-complexity.

REFERENCES

[1] J. Han and M. Kamber, “Data Mining: Concepts and Techniques”, Morgan Kaufman publishers, 2001.

[2] R. Agrawal and R. Srikant, “Mining Sequential Patterns”, In Proceedings of the 11th International Conference on Data Engineering, pp. 3-14, Taipei, Taiwan, 1995.

[3] Florent Masseglia, Pascal Poncelet and Maguelonne Teisseire, "Incremental mining of sequential patterns in large databases", Data & Knowledge Engineering, Vol. 46, No.1, pp. 97-121, 2003.

[4] Jiaxin Liu, “The design of storage structure for sequence in incremental sequential patterns mining,” Networked Computing and Advanced Information Management (NCM), pp. 330 - 334, 2010.

[5] Tzung-Pei, Hong,Ching-Yao Wang and Shian-Shyong Tseng, “An Incremental Mining Algorithm for Maintaining Sequential Patterns Using Pre-large Sequences,” Journal Expert Systems with Applications, Vol. 38, Issue 6,p p.7051-7058, 2011.

[6] Jen-Wei Huang, Chi-Yao Tseng, Jian-Chih Ou, Ming- Syan Chen, "A General Model for Sequential Pattern Mining with a Progressive Database," IEEE Transactions on Knowledge and Data Engineering, vol. 20, No. 9, pp. 1153-1167, 2008.

[7] Philippe Fournier,Viger,Roger Nkambou and Vincent Shin-Mu Tseng, “RuleGrowth: Mining Sequential Rules Common to Several Sequences by Pattern-Growth,” Symposium on Applied Computing, pp . 951-960, 2011.

[8] Jiaxin Liu, “The design of frequent sequence tree in incremental mining of sequential patterns,” Software Engineering and Service Science (ICSESS), pp. 679-682, 2012.

[9] M. Zaki, “Scalable data mining for rules”, Technical Report Ph.D. Dissertation, University of Rochester, New York, 1998.

[10] V. Chandra Shekhar Rao and P.Sammulal,"Survey On Sequential Pattern Mining Algorithms". International Journal of computer application(0975-8887),Vol 76-No.12, August 2013

[11] J.Pei, J.Han, B.MortazaviAsl, J.Wang, H.Pinto, Q.Chen, U.Dayal and M.-C.Hsu, ―Mining sequential patterns by pattern-growth: The PrefixSpan approach ‖, IEEE Transactions on Knowledge and Data Engineering, vol.16, no.11, 2004, pp. 1424-1440.

[12] NIZAR R. MABROUKEH and C. I. EZEIFE, A Taxonomy of Sequential Pattern Mining Algorithms, ACM Computing Surveys, Vol. 43, No. 1, Article 3, Publication date: November 2010.

[13] M. Garofalakis, R. Rastogi, and K. Shim, "SPIRIT: Sequential pattern mining with regular expression constraints", VLDB'99, 1999.

[14] Jian Pei, Jiawei Han and Wei Wang, "Constraint-based sequential pattern mining: the pattern-growth methods", Journal of Intelligent Information Systems, Vol:28, No: 2 ,pp:133-160, 2007.

[15] Mohammad J. Zaki,- SPADE: An Efficient Algorithm for Mining Frequent Sequences, Kluwer Academic Publisher. Machine Learning,42,31-60,2001.

[16] X. Yan, J. Han andR. Afshar, “CloSpan: Mining closed sequential patterns in large datasets”, Third SIAM International Conference on Data Mining (SDM), San Francisco,pp. 166–177, 2003.

[17] C.-C. Yu and Y.-L. Chen, “Mining Sequential Patterns from Multi-Dimensional Sequence Data”, IEEE Trans. Knowledge and Data Eng., Vol. 17, No. 1, pp. 136-140, Jan. 2005.

[18] Han J., Dong G., Mortazavi-Asl B., Chen Q., Dayal U., Hsu M.-C., Freespan: Frequent pattern-projected sequential pattern mining, Proceedings 2000 Int. Conf. Knowledge Discovery and Data Mining (KDD’00), 2000, pp. 355-359.

[19] J. Pei, J. Han, B. Mortazavi-Asi, H. Pino, "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix- Projected Pattern Growth", ICDE'01, 2001.

[20] Jian Pei, Jiawei Han, Wei Wang, －Constraint-based sequential pattern mining: the pattern growth methods, J Intell Inf Syst , Vol. 28, No.2, ,2007, pp. 133 － 160.