Research article Available online www.ijsrr.org ISSN: 2279–0543

International Journal of Scientific Research and Reviews

A Survey of Sequential Rule Mining Algorithms

Sachdev Neetu and Tapaswi Namrata*

Computer Science and Engineering, Institute of Engineering and Science, IPS Academy Indore, India. E-mail: Neetusachdev01@gamil.com

ABSTRACT

The mining of frequent sequential patterns or sequential rules has been the focus of knowledge discovery in databases. This paper aims to investigate efficient algorithm for mining including association rules and sequential patterns. Mining sequential patterns or rules with time constraints, such as time gaps and sliding time-window, may reinforce the accuracy of mining results. However, the capabilities to mine the time-constrained patterns were previously available only within Apriori framework. Recent studies indicate that pattern-growth methodology could speed up sequence mining. This paper presents a review of various sequential rule mining techniques.

Keywords: *Data Mining, sequential rule mining, pattern growth approach, minimum support, confidence, sequence database, frequent itemset.*

***Corresponding author**

Dr. Namrata Tapaswi

Professor & HOD, Computer Science and Engineering

Institute of Engineering and Science, IPS Academy

Indore, India

E-mail: hod.comps@ipsacademy.org

INTRODUCTION

Recent developments in computing and automation technologies have resulted in computerizing business and scientific applications in various areas. Turing the massive amounts of accumulated information into knowledge is attracting researchers in numerous domains as well as databases, machine learning, statistics, and so on. From the views of information researchers, the stress is on discovering meaningful patterns hidden in the massive data sets. Hence, a central issue for knowledge discovery in databases, additionally the main focus of this thesis, is to develop economical and scalable mining algorithms as integrated tools for management systems.

Data mining, that is additionally cited as knowledge discovery in databases, has been recognized because the method of extracting non-trivial, implicit, antecedently unknown, and probably helpful data from knowledge in databases. The information employed in the mining method usually contains massive amounts of knowledge collected by computerized applications. As an example, bar-code readers in retail stores, digital sensors in scientific experiments, and alternative automation tools in engineering typically generate tremendous knowledge into databases in no time. Not to mention the natively computing- centric environments like internet access logs in net applications. These databases therefore work as rich and reliable sources for information generation and verification. Meanwhile, the massive databases give challenges for effective approaches for information discovery.

The discovered information will be utilized in many ways in corresponding applications. For instance, distinctive the oft times appeared sets of things in a very retail info will be used to improve the choice creating of merchandise placement or commercial. Discovering patterns of client browsing and buying (from either client records or net traversals) could assist the modeling of user behaviors for client retention or customized services. Given the specified databases, whether relational, transactional, spatial, temporal, or transmission ones, we have a tendency to could get helpful info once the information discovery method if acceptable mining techniques square measure used.

BACKGROUND AND PROBLEM DEFINITION

If a collection of data sequences is given, within which every sequence may be a list of transactions ordered by the transaction time, the matter of mining sequential patterns [3] is to get all sequences with a user such minimum support. every transaction contains a collection of things. A sequential pattern is an ordered list (sequence) of itemsets. The itemsets that area unit contained within the sequence area unit referred to as parts of the sequence. For a given database D that

consists of client transactions every group action consists of the subsequent fields: customer-ID, transaction-time, and therefore the things purchased within the group action. An item-set may be a non-empty set of things, and a sequence is an order list of item-sets. We are saying a sequence A is contained in another sequence B if there exist integer i1.

Support = $\frac{\text{The number of sequence that contains this sequence}}{\text{The total number of sequences}}$

A sequence is an ordered list of elements (transactions). Each element contains a collection of events (items). Each element is attributed to a specific time or location. Length of a sequence, |s|, is given by the number of elements of the sequence.

Table 1: A Sequence Database

ID	Sequences
1	{1,2},{3},{6},{7},{5}
2	{1,4},{3},{2},(1,2,5,6}
3	{1},{2},{6},{5},{6,7}
4	{2},{6,7},{1,2},{2,3}

Considering a minimum support = 50% and minimum confidence = 50%, we get the following sequential rules.

Table 2: Sequential rules

Id	Sequential rule	Support	Confidence
1	{1,2,3} => {5}	0.5	1.0
2	{1} => {3,5,6}	0.5	0.66
3	{1,2} => {5,6}	0.75	0.75
4	{2} => {5,6}	0.75	0.75
5	{1} => {5,6}	0.5	0.5
..	..	..	..

The goal of sequential patterns is to search out the sequences that have larger than or equal to an explicit user pre-specified support. Sometimes the method of finding sequential patterns consists of the subsequent sections: sorting phase, finding the massive item-set phase, transformation section, sequence section, and greatest phase.

RELATED WORK

As we know, data are changing all the time; especially data on the web are highly dynamic. As time passes by, new datasets are inserted; old datasets are deleted while some other datasets are refreshed. It is transparent that time stamp is an important attribute of each dataset, also it's aristocratic in the process of data mining and it can give us more accurate and useful information. For example, association rule mining does not take the time stamp in account, the rule may Buy A=⇒Buy B. If we take time stamp in account then we can get more accurate and useful rules such as: Buy A implies Buy B within two days, three days four days or a week and a month, or usually people Buy A everyday in a week. The second kind of rules, business decision can be more accurate and useful prediction and consequently make more sound decisions.

Agrawal et al.[3] The AIS (Agrawal, Imielinski, Swami) algorithm put forth by Agrawal et al.[3] was the forerunner of all the algorithms used to generate the frequent itemsets and confident association rules, the description of which has been given along with the introduction of mining problem. The algorithm comprises of two phases. The first phase constitutes the generation of the frequent itemsets. This is followed by the generation of the confident and frequent association rules in the second phase. The exploitation of the monotonicity property of the support of itemsets and the confidence of association rules led to the enhancement of the algorithm.

Agrawal et al.[4, 5] Classical Apriori-based sequential pattern mining algorithms were first introduced by Agrawal and Srikant. Given the transaction database containing customer sequences, each of which has three attributes: customer-id, transaction-time and purchased-items ordered according to purchase time, the mining process was decomposed into five phases. Sort Phase, L-itemsets Phase, Transformation Phase, Sequence Phase, and Maximal Phase.

The Apriori-like algorithm is not so efficient, but it became the basis of many efficient algorithms developed later.

Agrawal et al.[6] GSP (Generalized Sequential Pattern) was introduced by Srikant and

Agrawal, it is also an Apriori-based pattern mining algorithm. The whole algorithm has two subprocesses: candidate pattern generation and frequent pattern generation.

The candidate sequences are generated in two steps: joining phase and pruning phase. In the joining phase, candidate k-sequences are generated by joining two (k-1) sequences that have the same contiguous subsequences. When joining the two sequences the item can be inserted as a part of the element or as a separate element. Pruning phase, those candidate sequences that have a contiguous subsequence whose support count is less than the minimal support are deleted. It also uses the hash-tree structure to reduce the number of candidates to be checked in the next phase.

Harms et al.[14] SPAM (Sequential PAttern Mining) is a typical algorithm which integrates a variety of old and new algorithmic contributions. It is lexicographic tree has been used to store all the sequences. SPAM traverses the sequence tree in a standard depth-first search (DFS) manner. At each node n, the support of each sequence-extended child is tested. If the support of a generated sequence s is greater than or equal to minimum support, SPAM stores that sequence and repeats the DFS recursively on s. (Note that the maximum length of any sequence is limited since the input database is finite.) If the support of s is less than minimum support, then SPAM does not need to repeat the DFS on s by the Apriori principle, since any child sequence generated from s will not be frequent. If none of the generated children are frequent, then the node is a leaf and user can backtrack up the tree.

Pie et al. [10] The PrefixSpan (Prefix-projected Sequential pattern mining) algorithm, representing the pattern-growth methodology, finds the frequent items after scanning the sequence database once. The database is then projected, according to the frequent items, into several smaller databases. Finally, the complete set of sequential patterns is found by recursively growing subsequence fragments in each projected database. Two optimizations for minimizing disk projections were describe. The *bi-level projection* technique, dealing with huge databases, scans each data sequence twice in the (projected) database so that fewer and smaller projected databases are generated. The *pseudo-projection* technique, avoiding physical projections, maintains the sequence-postfix of each data sequence in a projection by a pointer-offset pair. Although *PrefixSpan* successfully discovered patterns employing the divide-and-conquer strategy, the cost of disk I/O might be high due to the creation and processing of the projected sub- databases.

Yan et al. [11] In order to reduce the time and space cost when generating explosive

numbers of frequent sequence patterns, CloSpan (Closed Sequential Pattern Mining) was developed. Instead of mining the complete set of frequent subsequences, it mines only frequent closed subsequences, which are the sequences containing no super sequence with the same support (occurrence frequency) .

CloSpan divides the mining process into two stages. In the first stage, a candidate set is generated. Usually this candidate set is larger than the final closed sequence set. This set is called a suspicious closed sequence set (a super set of the closed sequence set). In the second stage, a pruning method is called to eliminate non-closed sequences.

One major difference between CloSpan and PrefixSpan is that CloSpan implements an early termination mechanism to avoiding unnecessary traversing of search space.By using both backward sub-pattern and backward super-pattern methods, some patterns will be absorbed or merged, and the search space growth can be reduced.

Tzvetkov et al. [12] The TSP (TOP-K Closed Sequential Mining) algorithm is a typical sequential mining algorithm; it could find the top-k closed sequential patterns from the sequence database. It only selects the top-k wanted sequences, which could avoid the users having to set the minimal sequence support value (considered as a trivial and very difficult task).

The TSP algorithm also uses the concepts of prefix projection-based sequential pattern mining and the PrefixSpan algorithm. It uses a hash collection, called SID-Hash, as a key in the phase of verifying the closed sequential pattern or minimal generator.

Brin et al. [7] the DIC algorithm that partitions the database into intervals of a fixed size so as to reduce the number of traversals through the database. Another algorithm called the CARMA algorithm (Continuous Association Rule Mining Algorithm) employs an identical technique in order to restrict the interval size to 1.

Zaki [9] SPADE (Sequential Pattern Discovery using Equivalence classes) algorithm completed the mining in three passes of database scanning. Nevertheless, additional computation time is required to transform a database of horizontal layout to vertical format, which also requires additional storage space several times larger than that of the original sequence database.

With rapid cost down and the evidence of the increase in installed memory size, many small or medium sized databases will fit into the main memory. For example, a platform with 256MB memory may hold a database with one million sequences of total size 189MB. Pattern

mining performed directly in memory now becomes possible.

Padmaja et al. [16] In order to reduce the number of iterations, the efficient bi-directional sequential pattern mining approach namely Recursive Prefix Suffix Pattern detection, RPSP algorithm is furnished. The RPSP algorithm finds first all Frequent Itemsets (FI‟s) according to the given minimum support and transforms the database such that each transaction is replaced by all the FI‟s it contains and then finds the patterns. Further the pattern detected based on ith projected databases, and builds suffix and prefix databases based on the Apriori properties. Recursive Prefix Suffix Pattern will increase the number of frequent patterns by reducing the minimum support and vice versa. Recursion gets deleted when the detected FI set of prefix or suffix assigned database of parent database is ineffective. All patterns that correlate to a particular ith proposition database of transformed database, that formed into a set, that is dis-joint from all the other sets. The resultant set of frequent patterns is the sum of the all disjoint subsets. The proposed algorithm tested on hypothetical and sequence data and obtained results were found all satisfactory. Hence, RPSP algorithm may be applicable to many real world sequential data sets.

Fournier-Viger et al. [17] CMRules: An association rule mining based algorithm for the discovery of sequential rules. The users can specify min_sup as a parameter to a sequential pattern mining algorithm. There are two major difficulties in sequential pattern mining:

(1) effectiveness: the mining may return a huge number of patterns, many of which could be uninteresting to users, and

(2) efficiency: it often takes substantial computational time and space for mining the complete set of sequential patterns in a large sequence database.

Author	Proposed work	Conclusion
Agrawal et al. [3]	AIS Algorithm	They generate frequent association rule for small datasets.
Agrawal et al. [4,5]	Apriori Algorithm	It is not effective for sequential rule mining.
Agrawal et al. [6]	GSP Algorithm	It is better than apriori algorithm.It also uses the hash-tree structure to reduce the number of candidates.
Harms et al. [14]	SPAM Algorithm	All Sequences are store in lexicographic tree. It uses depth

		first search manner.
Pie et al. [10]	Prefix-Span	Database projected according to frequent item set. It uses divide and conquer strategy.
Yan et al. [11]	CloSpan	It avoid unnecessary traversing of search space. It is better than prefix-span.
Tzvetkov et al. [12]	TSP Algorithm	It only selects the top-k wanted sequences. And it use hash collection. it is very difficult task.
Brin et al. [7]	DIC, CARMA Algorithm	Fixed interval size to reduce the number of database. It is restricted.
Zaki [9]	SPADE Algorithm	It requires additional storage space several times larger than that of the original sequence database.
Padmaja et al. [16]	RPSP Algorithm	RPSP algorithm may be applicable to many real world sequential data sets.
Fournier-Viger et al. [17]	CMRule Algorithm	It is better than the all previous algorithm.

CONCLUSION

In this paper, we presented a review of the algorithm for mining sequential rules. It is found that most of them are based on the generate-candidate-and-test approach. The sequential rule mining is a very popular and useful aspect of data mining. By using the sequential rule mining one can find the most frequent elements which occurred in a particular sequence.

REFERENCES

1. Pujari Arun. Introduction to data mining. Universities press. 2001
2. Piatetsky-Shapiro G and Frawley W . Knowledge Discovery in Databases, AAAI/MIT Press. 1991.
3. Agrawal R , Imielminski T, Swami A . Mining Association Rules between Sets of Items in Large Databases. SIGMOD Conference. 1993 ; 207-216 .
4. Agrawal R , Srikant R . Fast Algorithms for Mining Association Rules. In Proceedings of the 20th Int. Conf. Very Large Data Bases. 1994; 487-499.

5. Agrawal R , Srikant R, . Mining generalized association rules. Proceedings of the International Conference on Very Large Databases. San Francisco, CA: Morgan Kanfman Press. 1995; 406-419.
6. Srikant R & Agrawal R . Mining Sequential Patterns: Generalizations and Performance Improvements. Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology. 1996.
7. Brin S, Motwani R , Ullman J D, & Tsur S . Dynamic itemset counting and implication rules for market basket data. In Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, volume 26(2) of SIGMOD Record. 1997; . 255–264. ACM Press.
8. Han J , Pei J, Mortazavi-Asl B , Chen Q , Dayal U, & Hsu M C. FreeSpan: frequent pattern-projected sequential pattern mining. Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.2000.
9. Zaki M J . SPADE: An Efficient Algorithm for Mining FrequentSequences. Journal of Machine Learning. 2001.
10. Pei J , Han J, Mortazavi-Asl B, Pinto H, Chen Q, Dayal U, et al. PrefixSpan: Mining Sequential Patterns by Prefix-Projected Growth. Proceedings of the 17th International Conference on Data Engineering.2001
11. Yan X , Han J , & Afshar R . CloSpan: Mining Closed Sequential Patterns in Large Datasets. Proceedings of the 2003 SIAM International Conference on Data Mining (SDM'03).2003.
12. Tzvetkov P, Yan X, & Han J .TSP: Mining top-k closed sequential patterns. Knowledge and Information Systems.2005; 7(4), 438-457.
13. Das G , Lin K-I , Mannila H , Renganathan G , & Smyth P .Rule Discovery from Time Series. In Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining. 1998; 16-22.
14. Harms S. K. , Deogun J & Tadesse T . Discovering Sequential Association Rules with Constraints and Time Lags in Multiple Sequences. In Proc. 13th Int. Symp. on Methodologies for Intelligent Systems. 2002; 373-376.
15. Mannila H , Toivonen H , & Verkano A.I . Discovery of frequent episodes in event sequences. Data Mining and Knowledge Discovery. 1997; 259-289.
16. Padmaja P, Naga Jyoti P , Bhargava M . Recursive Prefix Suffix Pattern Detection Approach for Mining Sequential Patterns. IJCA September. 2011.

17. Fournier-Viger P , Faghihi U , Nkambou R , Mephu Nguifo E . CMRules: An Efficient Algorithm for Mining Sequential Rules Common to Several Sequences. Knowledge Based Systems. 2012.