Systematic Analysis of Sequential Pattern Mining Algorithms And Future Research Directions

Pragya Goel
M.Tech. Student, Department of Computer Science & Applications, Kurukshetra University, Haryana, India
Email: gpragya1992@gmail.com
Rajender Nath
Professor, Department of Computer Science & Applications, Kurukshetra University, Haryana, India
Email: rnath2k3@gmail.com

----------**ABSTRACT**----------
Web Usage Mining (WUM) is one of the key areas of Web Mining that deals with the discovery of user access patterns from weblog files. Pattern discovery phase in WUM attempts to discover hidden patterns from preprocessed weblog files by the application of various data mining techniques. Association rule mining is used in Pattern discovery phase to uncover associations among a set of frequently accessed web pages but does not take into account the sequence in which the web pages are accessed. Sequential pattern mining (SPM) takes care of the time order and thus it can be seen as a generalized model of association rule mining due to which more candidates are generated. SPM is one of the various techniques that have earned particular attention from the web mining community as these algorithms can determine the web page traversal sequences of the customers by analyzing the weblog files. SPM algorithms discovers the existent maximal frequent sequential patterns from the given sequence database. This paper intends to review sequential pattern mining algorithms systematically and draw research directions in this field.

Keywords: Apriori, Data mining, Pattern Growth, Sequential Pattern Mining, Web Usage Mining

1. Introduction

Discovering hidden knowledge from huge amount of data through the application of various techniques such as statistical methods, clustering, classification, association rule mining, sequential pattern mining etc is termed as data mining. Web mining is one of the main areas of data mining and is defined as the application of data mining techniques to either web log files or contents of the web documents or to the web document's hyperlink structure in order to extract from them the unknown knowledge and potentially valuable patterns. Web mining has been classified into three categories- web usage mining, web structure mining and web content mining. This paper deals with Web Usage Mining (WUM) that is an important area of web mining first proposed in [5] and deals with automatic discovery of user access patterns from weblog files that are generated whenever a user visits any website. Sequential pattern mining (SPM) is one of the key areas of research in the field of WUM as massive amount of data is being collected and stored in the form of web log files and many companies are increasingly becoming interested in mining sequential patterns from weblogs for analyzing the behavior of their customers. The problem of mining sequential patterns is to discover all sequential patterns with a user specified minimum support where the support of a pattern is number of transactions that contain the pattern. The SPM problem is concerned with inter-transaction patterns as opposed to

association rule mining that considered only intra- transaction patterns. The SPM can be utilized in WUM by analyzing the weblogs of customers and determining the sequence of web pages visited by a particular user. The patterns discovered from web logs are the sequences of most frequently accessed pages at a particular site. The task of discovering all frequent sequential patterns from sequential database or weblog file can be quite challenging and a number of algorithms have been proposed for mining interesting patterns in sequence database[15][16][17]. This paper intends to review SPM algorithms systematically and draw research directions in this field.

Rest of the paper is organized as follows: Section 2 presents the research methodology, section 3 presents the literature survey of SPM algorithms, section 4 presents a comparative analysis of SPM algorithms, section 5 outlines areas of future scope and gives concluding remarks.

2. **Research methodology**

Firstly, research papers relating to SPM will be collected from various sources such as ACM, Springer and IEEE etc. Then these research papers will be classified into different categories based on the SPM techniques used in the papers. After that representative research papers along each class will be critically analyzed and research directions will be drawn.

3. **Sequential Pattern Mining**

Total 12 research papers were collected and were classified into two categories viz. apriori based algorithms and pattern growth based algorithms. Eight research papers were found under the category of apriori based algorithms and four under pattern growth based algorithms. The research papers under each category are discussed and analyzed below.

3.1 Apriori-based Algorithms

Most of the earlier algorithms of SPM used Apriori-based approach. Apriori algorithm was first proposed in [1] according to which all subsequences of a frequent sequence must also be frequent. It was also described as antimonotonic or downward closed since if a sequence could not pass the minimum support test, its entire super sequences would also fail the test. Apriori based algorithms are further classified into two types based on database format-horizontal database format and vertical database format.

3.1.1 Horizontal Database format

In horizontal database format, data set was represented as pairs of <sequence id: sequence of objects>. Some of the early Apriori based algorithms [2][3][4][9]used horizontal database format, which are discussed and analyzed below. AprioriAll, AprioriSome and DynamicSome were three algorithms first proposed in [2] for mining sequential patterns. AprioriAll was a three phase algorithm. It first used Apriori property to find all frequent itemsets, then replace each transaction by the set of all frequent itemsets contained in the transaction, then made multiple passes over the database to generate candidates and finally counted the support of candidates to discover sequential patterns. . The first algorithm AprioriAll discovered all the patterns while the latter two algorithms discovered only maximal sequential patterns. The performance of AprioriAll algorithm was experimentally found better than other two algorithms. However this approach almost doubled the disk space requirement that could be troublesome for large databases.

GSP (Generalized Sequential Pattern) algorithm proposed in [3] needed to scan the database multiple times. It also incorporated certain constraints into the mining process (i) time constraints that specified minimum or maximum gap between adjacent elements in a pattern (ii) sliding window constraint i.e. a set of items can be taken as in the same transaction if the distance between minimum and maximum transaction time of these items was not bigger than sliding window. (iii) item taxonomies that generated multilevel sequential patterns.

The PSP algorithm [4] for discovering sequential patterns was widely inspired from GSP but made some improvements that made it possible to perform retrieval optimizations. The main algorithm was same as that of GSP using candidate generation and prune approach;

however, to improve the efficiency of retrievals, PSP used a different hierarchical structure than in GSP for organizing candidate sequences. GSP used hash tables at each internal node of the candidate tree whereas PSP used prefix tree that organized the candidates according to their common elements that resulted in faster retrievals as well as lower memory overhead.

Table1: Horizontal Database Format Apriori Based Algorithms

Algorithm name	Key features	Weaknesses
AprioriAll	BFS based approach, first algorithm for mining sequential patterns	Multiple database scans, exponential growth of candidate sequences, double disk space requirement.
GSP	BFS based approach, use of hash tables to reduce the number of candidates, incorporated constraints, 20 times faster than AprioriAll algorithm	Multiple database scans, High computational and I/O costs, difficulty in mining long sequential patterns, Not a main memory algorithm.
PSP	Prefix tree structure, retrieval optimization, less memory required than GSP	Multiple database scans still required
MFS	Used modified version of GSP candidate generation function, lower I/O cost compared to GSP, early support checking for longer sequences, used sampling	As the sample size is increased, more work is done on candidate counting and CPU cost also increased.

Table 1 summarizes the key features and the weaknesses found during analysis of the Horizontal database format apriori based algorithms.

The candidate generating function used by GSP was extended in [9] by proposing a novel two stage algorithm called MFS that reduced the I/O cost needed by GSP. While GSP discovered frequent sequences of the same length in each database scan, MFS, on the other hand used a refinement approach. First of all, a rough estimate of the set of all frequent sequences was computed as a suggested frequent sequence set and then candidate generation function of GSP was generalized to maintain the set of maximal frequent sequences known so far. Thus, longer sequences could be generated and counted early, which was the major source of efficiency improvement of MFS over GSP.

3.1.2 Vertical Database Format

In vertical database format, rows of the database consisted of object time stamped pairs associated with an event. Using vertical database format provided the benefit of generating patterns and calculating their support count without performing costly database scans. A number of Apriori based algorithms [10][11][12][14] were based on vertical database format.

A novel method called SPADE (Sequential Pattern Discovery using Equivalence classes) was proposed in [10] for mining sequential patterns based on vertical database format. SPADE decomposed the original problem into smaller sub problems and used efficient lattice search techniques to solve them independently in main memory. Only three database scans were required to discover all the sequences. The major performance improvement was due to the use of ID lists for each candidate due to which the support count was calculated from its ID list, thus reducing the cost of scanning. The authors decoupled the problem decomposition from pattern search that reduced both computational and input output costs. Experiments showed that SPADE was twice as fast as GSP, the reason

being the use of more efficient support counting method based on idlist structure.

SPAM (Sequential PAttern Mining using bitmap representation) was proposed in [11] that integrated variety of old and new algorithmic approaches into a practical algorithm. The authors claimed SPAM to be the first depth first search strategy for sequential pattern mining and it encoded the ID lists from SPADE to a vertical bitmap data structure and put them in memory that made joining operation between ID lists extremely fast. On scanning the database for the first time, a vertical bitmap for each item in the database was constructed with each bitmap having a bit corresponding to every element of the sequence in the database. SPAM was similar to SPADE but it used bitwise operations rather than temporal joins.

An algorithm called GO-SPADE was proposed in [12] that extended SPADE to incorporate generalized occurrences. The motivation behind GO-SPADE was that many sequential databases could contain repetition of items that caused performance degradation in the traditional SPADE approach. The authors introduced the concept of generalized occurrences which were compact representations of several occurrences of a pattern and described corresponding primitive operators to manipulate them. Using such a representation reduced the size of ID lists significantly if large number of consecutive occurrences appeared in the database. The authors claimed that this approach not only reduced the memory space used during the process of extraction but also significantly reduced the join cost and therefore, the overall execution time.

Table 2: Vertical Database Format Apriori Based Algorithms

Algorithm name	Key features	Weaknesses
SPADE	Support counting from ID-lists, Prefix based equivalence class, Minimize computational costs by using lattice based approach for search space partitioning ,Minimize I/O cost by reducing database scans to three	Three database scans can still take lot of time in case of huge databases.
SPAM	First algorithm to use DFS based approach, bitmap representation, time efficient algorithm	Required to fit all data in main memory
GO-SPADE	Incorporate generalized occurrences, reduce memory space and join costs since GOID list's size is smaller than normal ID lists	Efficient only with databases containing consecutive repetition of items
bitSPADE	Combination of SPADE and SPAM, semi vertical database, vertical bitmap representation, uses lattice concept of SPADE	In terms of speed, bitSPADE is still slower than SPAM algorithm

Table 2 summarizes the key features and the weaknesses found during analysis of the Vertical database format apriori based algorithms.

A novel algorithm bitSPADE was presented in [14] that combined the best features of SPADE, one of the most memory efficient algorithm and SPAM, the fastest algorithm. The authors used the concept of semi vertical database using bitmap representation of SPAM and combined this semi vertical database with SPADE's lattice decomposition into independent equivalence classes that allowed fast and efficient enumeration of frequent sequences. A new pruning strategy was also presented that could be applied independently to each equivalence class.

3.2 Pattern growth-based algorithms

The main overhead in Apriori based algorithms was the generation of candidate sequences. Therefore to improve efficiency, pattern growth algorithms were proposed [6][7][8][13] that avoid the candidate generation step. Therefore

pattern-growth algorithms however, are more complex to develop and maintain as compared to Apriori based algorithms but were faster when given large amounts of data.

Free Span was proposed in [6] that was projection based as opposed to previous algorithms that were Apriori based. Free Span integrated the mining of frequent sequences with frequent patterns and used projected database that confined the search and growth of subsequent fragments. This method greatly reduced the generation of candidate subsequences by using projected databases. The authors also found experimentally that Free Span was considerably faster than Apriori based GSP algorithm.

Table 3: Pattern Growth Based Algorithms

Algorithm name	Key features	Weaknesses
WAP-mine	Two database scans, WAP tree data structure, better scalability than GSP	Memory consumption more.
FreeSpan	DFS based approach, three database scans, projected sequence database	Size and number of projected databases is very large
PrefixSpan	DFS based approach, two database scans, projected prefix database, reduce size and number of projected databases through pseudo projection	Pseudo projection technique consumes considerable amount of memory due to the use of in-memory sequence database
PLWAP	Position coded version of WAP tree, less memory consumption than WAP-mine	Memory still needed to store the position code.

Table 3 summarizes the key features and the weaknesses found during analysis of the pattern growth based algorithms.

WAP-mine algorithm proposed in [7] was major contribution made as a pattern growth technique for efficient mining of access patterns from web logs. This algorithm scanned sequence database only twice to build the WAP (Web Access Pattern) tree from frequent sequences along with their support threshold, a header table is also maintained that pointed at first occurrence of each item in frequent itemset and was later tracked in a threaded way to mine the tree for frequent patterns, building on suffixes. The algorithm required only two database scans: first scan that found frequent-1 sequences and second scan builds WAP tree with only frequent sequences. To mine the tree, WAP mine algorithm was proposed that had better scalability than GSP but suffered from memory consumption problem, as it recursively reconstruct many intermediate WAP trees during mining as the number of mined frequent patterns increased.

Prefix Span algorithm was proposed in [8] and was built around the concept of Free Span. The authors found that the major cost of Free Span was to deal with projected databases and if the pattern appeared in all sequences of the database then projected databases does not shrink. The main idea of Prefix Span was that instead of projecting sequence databases, only the prefix subsequences were examined and only their corresponding postfix subsequences were projected into the projected databases. To reduce the cost of construction and scanning of projected databases, another projection method called bi-level projection was used.

PLWAP [13] used a binary code assignment algorithm for constructing WAP tree that was pre-ordered and position-coded linked in which each node was assigned a binary code used during mining that determined which sequences were the suffix sequences of the last event and also to find the next prefix for mined suffix without the need to reconstruct intermediate

WAP trees, thus solving the problem posed in [7].

4. Comparative analysis

A comparative analysis of different sequential pattern mining algorithms is carried out to find future research directions. The different SPM algorithms discussed above are compared on five parameters- Search approach used, No of Database scans, Constraints used, Search space partitioning and Main memory algorithm.

4.1 Apriori Based Algorithms

Table 4: Comparison of Horizontal Database Format Based Algorithms

Evaluation parameters	AprioriAll	GSP	PSP	MFS
Search approach used	BFS	BFS	BFS	BFS
No of DB scans	Many	Many	Many	Many
Constraints	No	Yes	Yes	No
Search space partitioning	No	No	No	No
Main memory algorithm	No	No	No	No

Table 5: Comparison of Vertical Database Format Based Algorithms

Evaluation parameters	SPADE	SPAM	G0-SPADE	bit-SPADE
Search approach used	BFS and DFS	DFS	BFS and DFS	BFS and DFS
No of DB scans	3	2	3	3
Constraints	No	No	No	No
Search space partitioning	Yes	No	Yes	Yes
Main memory algorithm	No	Yes	No	No

4.2 Pattern Growth Based Algorithms

Table 6: Comparison of Pattern Growth Based Algorithms

Evaluation parameters	WAP Mine	Free Span	Prefix Span	PL-WAP
Search approach used	-	DFS	DFS	-
No of DB scans	2	3	2	2
Constraints	No	No	No	No
Search space partitioning	Yes	Yes	Yes	Yes
Main memory algorithm	Yes	Yes	Yes	No

Discussion: Analysis results have shown that most of the algorithms make use of BFS approach. Vertical database format apriori based algorithms and pattern growth based algorithms requires two to three database scans which are much more efficient than horizontal database format apriori based algorithms that requires multiple database scans. A few algorithms incorporate constraints into the mining process. Most of the algorithms make use of search space partitioning that allows partitioning of large search space of candidate sequences for efficient memory management. All the pattern growth based algorithms are main memory algorithms.

From the currently available algorithms, the best Apriori-based algorithm in terms of the number of database scans require three database scans and the best pattern growth-based algorithm requires two database scans. Therefore, a technique needs to be developed that will further reduce the number of database scans since database scan is very costly in time.

5. Conclusion And Future Research Directions

This paper has analyzed the existing SPM algorithms systematically in order to draw future research directions. All the research papers collected were classified under two broad categories viz. apriori based and pattern growth based and under each category, the

representative research papers were analyzed and their limitations were found. Though a number of algorithms have been proposed for efficiently discovering existent maximal frequent sequences from sequence databases, there are few demanding issues in SPM that can be comprehensive for future research such as developing a strategy to reduce the number of scans in sequential database. That is, an algorithm needs to be developed that should be able to process the huge search space and reduce the recurring scanning of database during the mining process.

References

[1] Agrawal, R., & Srikant, R. Fast algorithms for mining association rules. ACM, (pp. 487-499). Santiago, Chile, 1994.

[2] Agrawal, R., & Srikant, R. Mining sequential patterns. IEEE, (pp. 3-14). Taiwan, 1995.

[3] Srikant, R., & Agrawal, R. Mining sequential patterns: Generalizations and performance improvements. 5th International Conference on Extending Database Technology: Advances in Database Technology (pp. 3-17). London: Springer-Verlag, 1996.

[4] Masseglia, F., Cathala, F., & Poncelet, P. The PSP approach for mining sequential patterns. 2nd European Symposium on Principles of Data Mining and Knowldege Discovery (pp. 176-184). London: Springer-Verlag, 1998.

[5] Cooley, R., & Mobasher, B., Data preparation for mining World Wide Web browsing patterns.Knowledge and Information Systems, 5-32, 1999.

[6] Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., & Hsu, M.-C. FreeSpan:frequent pattern-projected sequential pattern mining . 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 355-359). New York: ACM, 2000.

[7] Pei, J., Han, J., Mortazavi-Asl, B., & Zhu, H. Mining Access Patterns Efficiently from Web Logs.4th Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 396-407). London: Springer-Verlag, 2000.

[8] Pei, J., Han, J., Moratzavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., et al. PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern growth.17th International Conference on Data Engineering (pp. 215-224). IEEE,2001.

[9] Zhang, M., Kao, B., Yip, C. L., & Cheung, D. A GSP based efficient algorithm for mining frequent sequences. International Conference on Artificial Intelligence. Las Vegas,2001.

[10] Zaki, M. J. SPADE: An Efficient algorithm for mining frequent sequences. Machine Learning, 31-60,2001.

[11] Ayres, J., Flannick, J., Gehrke, J., & Yiu, T. Sequential Pattern Mining using a bitmap representation. 8th ACM SIGKDD International Conference on Knowledge Discovery and data mining (pp. 429-435). New York: ACM,2002.

[12] Leleu, M., Rigotti, C., Boulicaut, J.-F., & Euvrard, G. GO-SPADE: Mining Sequential Patternsover Datasets with Consecutive Repetitions. 3rd International Conference on Machine Learning and Data Mining in Pattern Recognition (pp. 293-306). Leipzig: Springer Berlin Heidelberg, 2003.

[13] Ezeife, C. I., Lu, Y., & Liu, Y. PLWAP sequential mining: open source code. 1st

International workshop on open source data mining: frequent pattern mining implementations (pp. 26-35). New York: ACM,2005.

[14] Aseervatham, S., Osmani, A., & Viennet, E. bitSPADE:A Lattice-based Sequential Pattern Mining Algorithm Using Bitmap Representation. 6th International Conference on Data Miniing (pp. 792-797). Hong Kong: IEEE,2006.

[15] Mabroukeh, N. R., & Ezeife, C. I. A taxonomy of sequential pattern mining algorithms. ACM Computing Surveys,2010.

[16] Boghey, R., & Singh, S. Sequential Pattern Mining: A Survey on approaches. International Conference on Communication Systems and Network Technologies (pp. 670-674). Gwalior: IEEE,2013.

[17] Mooney, C. H., & Roddick, J. F. Sequential Pattern Mining-approaches and algorithms. ACM Computing Surveys,2013.