Analysis of Sequential Pattern Mining Algorithms

Prof. Alpa Reshamwala, Ms. Neha Mishra
Assistant Professor MPSTME NMIMS, M.Tech Student MPSTME NMIMS
alpa.reshamwala@nmims.edu, misniya@gmail.com

Abstract- Sequential pattern mining is an important data mining problem with broad applications. Most of the previously developed sequential pattern mining methods, such as SPAM and SPADE, explore a candidate generation-and-test approach [12] which reduces the number of candidates to be examined. In this paper, we have implemented SPADE, SPAM and Prefixspan algorithm on the two databases. One database is sign database which is taken from ASL (American sign language database) [11]. The second dataset is Kosarak dataset containing 10000 sequences of click-stream data from an hungarian news portal. Sign dataset forms the dense dataset with few distinct items and Kosarak forms the sparse dataset with maximum distinct items. From the experimental results, SPADE performs better in both the dense as well as sparse dataset taken for simulation study. Performance of SPAM is worst when executed on sparse dataset. The number of sequences generated is same in both the dataset by all the mentioned algorithms. For dense dataset prefixsapn uses less memory whereas in sparse dataset it utilizes the most. In Dense dataset SPAM and SPADE are utilizing approximately constant memory. In sparse dataset minimum utilization of memory is by SPADE.

Index Terms- Prefixspan, SPAM, SPADE, Kosarak dataset, Sign dataset, Frequent Sequences

I. INTRODUCTION

Data mining is the process of finding the useful and previously unknown information from the databases. The discovered information can be helpful in applications of the data mining such as DNA analysis, stock exchange and so on. In the field of the data mining, to find the sequential patterns is a tremendous task. Sequential pattern mining is the task of finding the complete set of frequent subsequences given a set of sequences. A huge number of possible sequential patterns are hidden in databases [2]. Sequential pattern mining algorithms [4] are very important to efficiently deal with such amounts of information and to deliver the results in an acceptable timeframe required by various real-world applications. With the help of Sequential pattern mining algorithms data analysts decide that which sequences are frequently occurred in the sequential database. For mining frequent patterns in large data sets [1], three pattern mining methods are used and evaluated:

A mining algorithm should be able to find the complete set of patterns, when possible, satisfying the minimum support (frequency) threshold, highly efficient, scalable, involving only a small number of database scans and to incorporate various kinds of user-specific constraints[14]. To overcome these problems, a parallel Prefixspan approach can be proposed. Mining tasks are decomposed to many small tasks, the Map function is used to mine each Prefix-Projected sequential pattern, and the projected databases will be constructed parallelly. It will simplify the search space and will acquire a higher mining efficiency. Then the intermediate values will be passed to a Reduce function which will merge together all these values to produce a possibly smaller set of values. Theoretical analyses shows that Parallel-PrefixSpan will reduce the time of scanning database. It will also solve the problem of mining massive data effectively, has considerable speedup and scaleup performances with an increasing number of processors.

II. RELATED WORK

The main challenge towards the problem of mining sequential patterns is the high processing cost due to a large amount of data [1]. Many algorithms have been proposed to speed up the mining process. The representative ones are SPAM [3][14][15], SPADE [5], and Prefixspan [6]. PrefixSpan algorithm is used for predicting DoS attacks sequences on KDD cup 99 training dataset which is efficient. which is then compared with SPAM (Sequential Pattern Mining) algorithm which uses vertical bitmap data layout allowing for simple, efficient counting described in [19]. Apriori a candidate generation algorithm and SPAM (Sequential Pattern Mining) algorithm on Yahoo! Music KDD Cup 2011 are compared. From these discovered patterns, we can know what patterns or music sequences which are frequently heard and in what order they are recommended. Experimental results have shown that SPAM performs well for large datasets like Yahoo! Music dataset is due to the bitmap representation of the data for efficient counting [20].

SPAM [2] (Sequential PAttern Mining) assumes that the entire database (and all data structures used for the algorithm) completely fit into main memory [13]. With the size of current main memories reaching gigabytes

and growing, many moderate-sized to large databases will soon become completely memory-resident [11][16]. Considering the computational complexity that is involved in finding long sequential patterns even in small databases with wide records, this assumption is not very limiting in practice [4][17]. The SPADE algorithm adopts a bottom-up approach to generate frequent sequences with different lengths [8]. By iteration, this approach computes the support count of a candidate k-sequence generated by merging the ID-lists of any two frequent (k-1)-sequences with the same (k-2)-prefix. The SPADE algorithm costs a lot to repeatedly merge the ID-lists of frequent sequences for a large number of candidate sequences [9][18].
On the other hand, Pei et al. [6] employ the projection scheme in the Prefixspan algorithm to project the customer sequences into overlapping groups called projected databases such that all the customer sequences in each group have the same prefix which corresponds to a frequent sequence [10]. The Prefixspan algorithm [14] first scans the database to find the frequent 1-sequences [12]. After that, this algorithm generates the projected database for each frequent 1-sequence. For the projected database, the Prefixspan algorithm continues the discovery of frequent 1-sequences to form the frequent 2-sequences [7].

III. ALGORITHMS

Sequential pattern mining is the mining of frequently occurring ordered events or subsequences as patterns literature [5,6].Compared with projected databases and subsequence connections, PrefixSpan [11] was more efficient than SPADE and SPAM. PrefixSpan does not require candidate generation, also it can reduce the scale of projected databases substantially relative to the original sequence database, and the major cost of PrefixSpan is the construction of projected databases. In addition, scanning projected databases repeatedly also reduce the efficiency of the algorithm. Generally speaking, reducing both the scale of projected databases and the time of scanning projected databases are the main ways of improving PrefixSpan [7,8,9]. However, when mining long frequent concatenated sequences, this method is inefficient. Therefore, it is impractical to apply PrefixSpan to mine long contiguous sub-sequences from sequencial database. There are some challenges in sequential pattern mining which are [9]: (1) A huge number of possible sequential patterns are hidden in databases (2) A mining algorithm should -find the complete set of patterns, when possible, satisfying the minimum support (frequency) threshold, It should be highly efficient, scalable, involving only a small number of database scans and It should be able to incorporate various kinds of user-specific constraints.The sequential pattern mining extract various patterns from the sequential database. The algorithms which are implemented are explained below:

SPADE Algorithm: The SPADE algorithm [5] utilizes combinatorial properties to decompose the original problem of finding frequent patterns into smaller sub-problems that can be independently solved in the main-memory using efficient lattice search techniques and simple join operations.
SPAM Algorithm: SPAM assumes that the entire database (and all data structures used for the algorithm) completely fit into main memory. The SPAM uses a vertical bitmap data layout allowing for simple, efficient counting.
The Prefixspan Algorithm: The Prefixspan algorithm [17] first scans the database to find the frequent 1-sequences. After that, this algorithm generates the projected database for each frequent 1-sequence. In this way, the Prefixspan algorithm recursively generates the projected database for each frequent k-sequence to find frequent (k+1)-sequences [8].
The performance of Prefixspan Pattern mining method [6] in terms of the execution time and memory was evaluated on sign and Kosarak Databases and compared with SPADE and SPAM algorithm of frequent pattern mining [7] .

IV. EXPERIMENTAL RESULTS

For implementation of the pattern mining algorithm, a system having windows 7, Intel Core i3 processor 2.10 GHz with 2 GB RAM with JDK 1.7 is used. The datasets are downloaded from SPMF (Sequential Pattern Mining Framework) which is implemented by Phillipe Fournier-Viguera [15] and available at http://www.philippe-fournier-viger.com/spmf/.
The first dataset is taken from the ASL (American Sign Language Database)[11] which, contains a collection of utterance. Each utterance associates a segment of video with a detailed transcription. For each utterance, a number of ASL gestural and grammatical fields (e.g. eye-brow raise, head tilt forward, wh-question), each one occurring over a time interval are considered. Major focus of the implementation is to detect all frequent arrangements of temporal intervals, where each interval label corresponds to a grammatical/syntatic/gestural field of ASL. The second dataset is Kosarak dataset which was provided by Ferenc Bodon to FIMI repository [21] and contains (anonymized) click-stream data of a Hungarian on-line news portal. From the Table 1, the number of distinct items in kosarak dataset is more than the number of distinct items in sign dataset when compared with the number of sequences in the

dataset respectively. Hence, Sign dataset forms the dense dataset with few distinct items and Kosarak forms the sparse dataset with maximum distinct items.

TABLE 1: STATISTICAL PARAMETERS OF DATABASES

Sr. No	Statistical Parameters	Sign	Kosarak
1	Number of sequences	730	10000
2	Number of distinct items	267	10094
3	Average number of itemsets per sequence	51.99	8.14
4	Average number of distinct item per sequence	51.99	8.14
5.	Largest item id	310	10094
6.	Average number of occurrences for each item in a sequence	1	1
7.	Average number of items per itemset	1	1

For the Dense dataset such as sign dataset, when the minimum support threshold is low, the runtime of the SPADE is lowest as compared to the SPAM and SPADE but as the value of the supported threshold increases, the runtime of the Prefixspan is almost same as SPAM and SPADE, which is shown in fig 1. SPADE performs better than the other two.

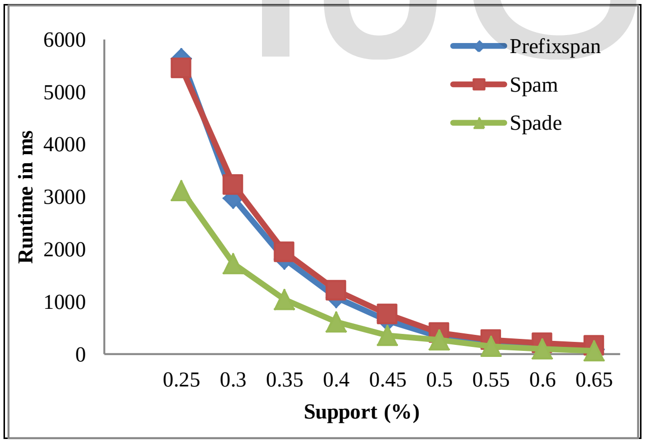

Fig 1: Runtime for Sign Dataset

In the terms of memory, when the support is low, the memory used by Prefixspan algorithm is low, but as the support is increased, the memory usage is also less. When the value of support threshold is increased, the prefixspan algorithm performs better than the SPAM and SPADE algorithm for pattern mining which is shown in fig 2. SPADE and SPAM algorithm maintains a constant memory usage for the Sign dataset.

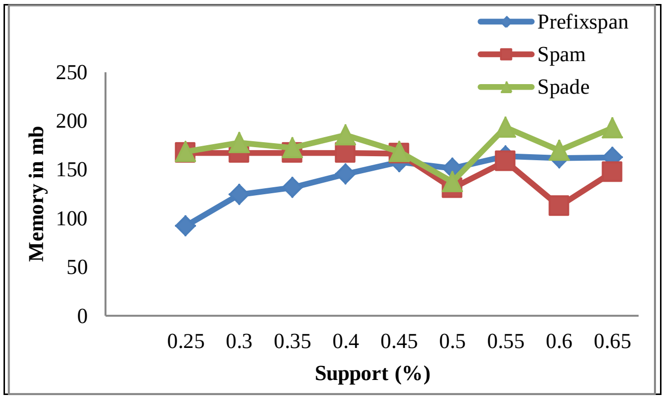

Fig 2: Memory used for Sign Dataset

In the terms of sequences, the prefixspan, SPAM and SPADE discovers almost same number of sequences for the sign database as shown in fig 3.

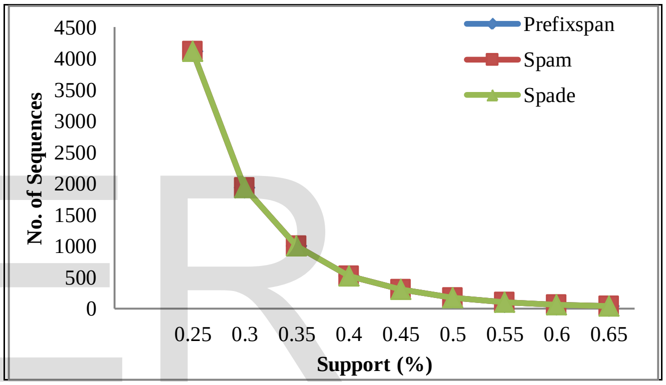

Fig 3: Sequences for Sign Dataset

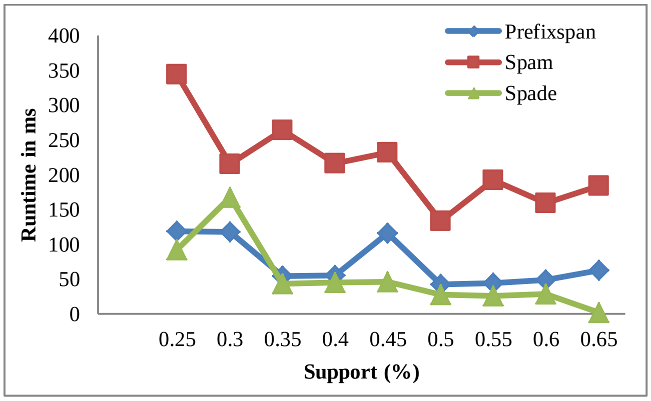

Fig 4: Runtime for Kosarak10k Dataset

For the Kosarak 10k Dataset, when the minimum support threshold is low, the runtime of the Spade is lowest as compared to the SPAM and Prefixspan. The SPAM algorithm takes the maximum time from the low to high support values which are shown in fig 4.

In the terms of memory, the PrefixSpan algorithm consumes more memory as compared to the Spade and SPAM algorithm. Spade algorithm uses less memory in the low support values but after 0.35 it increases which is shown in fig 5. SPADE uses the least memory in sparse dataset such as kosarak 10k. Prefixspan algorithm maintains an average usage of the memory. For the Kosarak database, Prefixspan, Spade and Spam generates same number of sequences shown in fig 6.

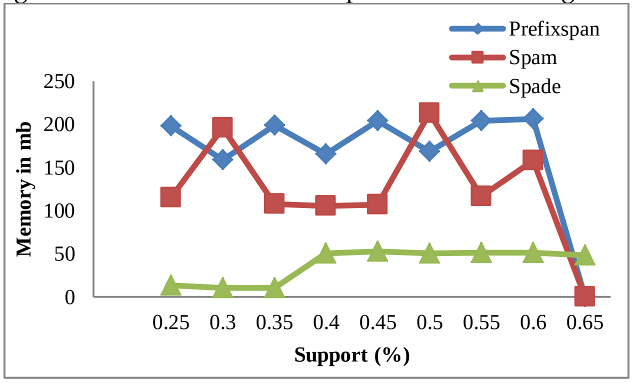

Fig 5: Memory used for Kosarak Dataset

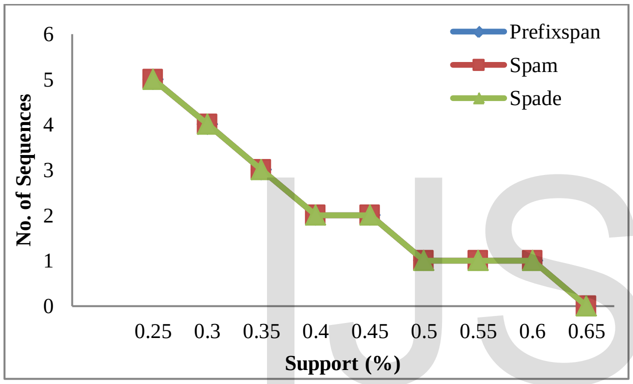

Fig 6: Sequences of Kosarak Dataset

From fig 1 and 4, SPADE performs better in both the dense as well as sparse dataset such as sign and kosarak10k respectively. Performance of SPAM is worst when executed on sparse dataset. In fig 4, Prefixspan is approximately approaching the performance of SPADE in sparse dataset whereas the performance of SPAM and Prefixspan is same in dense dataset. From fig 3 and 6 the number of sequences generated is same in both the dataset. From fig 2, for dense dataset prefixspan uses less memory whereas in sparse dataset it utilizes the most. From fig 2, in dense dataset SPAM and SPADE are utilizing approximately constant memory. From fig 5, in sparse dataset minimum utilization of memory is by SPADE.

V. Conclusion

The comparison study of SPADE, SPAM and the Prefixspan algorithm is done on the results collected. As per the results, SPADE performs better in both the dense as well as sparse dataset such as sign and kosarak10k respectively. Performance of SPAM is worst when executed on sparse dataset. Prefixspan is approximately approaching the performance of SPADE in sparse dataset whereas the performance of SPAM and Prefixspan is same in dense dataset. The number of sequences generated is same in both the dataset. For dense dataset prefixsapn is uses less memory whereas in sparse dataset it utilizes the most. In Dense dataset SPAM and SPADE are utilizing approximately constant memory. In sparse dataset minimum utilization of memory is by SPADE.

REFERENCES

[1] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," Proc. 1994 Int'l Conf. Very Large Data Bases (VLDB '94), pp. 487- 499, Sept. 1994.

[2] R. Agrawal and R. Srikant, "Mining Sequential Patterns," Proc.1995 Int'l Conf. Data Eng. (ICDE '95), pp. 3-14, Mar. 1995.

[3] Srikant R. and Agrawal R., _Mining sequential patterns: Generalizations and performance improvements, Proceedings of the 5th International Conference Extending Database Technology, 1996, 1057, 3-17.

[4] 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.

[5] M. Zaki, "SPADE: An efficient algorithm for mining frequent sequences", Machine Learning, 2001.

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

[7] Ching-Yao-Wang, Tzung-Pei Hong, S.S.T. 2001 "Maintainance of sequential pattern for record deletion", International conference on data mining pp. 536-541.

.[8] Ayres, J., Flannick, J., Gehrke, J., And Yiu, T., Sequential pattern mining using a bitmap representation, In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining-2002.

[9] C. Antunes, A. L. Oliveira, "Generalization of Pattern-growth Methods for Sequential Pattern Mining with Gap Constraints", Machine Learning and Data Mining in Pattern Recognition, Third International Conference, MLDM 2003, Leipzig, Germany, July 5-7, 2003, Proceedings.

[10] Y. L. Chen, M. C. Chiang, and M. T. Kao, "Discovering time-interval sequential patterns in sequence databases", Expert Systems with Applications, Vol. 25, No. 3, pp. 343-354, 2003.

[11] Show-Jane Yen and Yue-Shi Lee, "Mining Sequential Patterns with Item Constraints", DaWaK 2004: data warehousing and knowledge discovery: International conference on data warehousing and knowledge discovery, Zaragoza, ESPAGNE, vol. 3181, pp. 381-390, 2004.

[12] N Ju-Dong Ren, Yin-Bo Cheng, Lung-Lung Yang, "An Algorithm for Mining Generalized Sequential Patterns", Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 August 2004.

[13] Ming-Yen Lin and Suh-Yin Lee, "Incremental update on sequential patterns in large databases by implicit merging and efficient counting", Information Systems, Vol. 29, No. 5, pp. 385-404, 2004.

[14] Yen-Liang Chen, Ya-Han Hu, "The consideration of recency and compactness in sequential pattern mining", In Proceedings of

the second workshop on Knowledge Economy and Electronic Commerce, Vol. 42, Iss. 2, pp. 1203-1215, 2006.

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

[16] Jean Francois Boulicaut, “If constraint based mining is the answer: what is the constraint? (Invited Talk)”, IEEE International conference on data mining workshops, 2008.

[17] Shigeaki Sakurai, Youichi Kitahata and Ryohei Orihara, “Discovery of Sequential Patterns based on Constraint Patterns”, international journal of computational intelligence, 2008.

[18] WEI Yong-qing, LIU Dong, DUAN Lin-shan, “Distributed PrefixSpan Algorithm Based on MapReduce”, International symposium on information technology in medicine and education, 2012.

[19] Alpa Reshamwala and Dr. Sunita Mahajan, “Prediction of DoS attack Sequences”, International Conference on Communication, Information and Computing Technology (ICCICT-2012), Mumbai, October 18-20, 2012.

[20] Sunita Mahajan, Alpa Reshamwala, Nisha Sharma, Divya Vineet, Akshay Sharma, Parshwa Shah, “Prediction of Yahoo! Music Sequences on user’s musical taste”, International Conference on Advances in Information Technology- 2012 – 23rd June 2012, Bangkok, Thailand, pp 6-9,DOI- 10.3850/978-981-07-2683-6 AIT-102.

[21] FIMI-Frequent Itemset Mining Implementations Repository, (Software developed by Ferec Bodon, URL: http://fimi.cs.helsinki.fi/src)