ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.6, Pages : 156-160 (2013)
Special Issue of ICETEM 2013 - Held on 29-30 November, 2013 in Sree Visvesvaraya Institute of Technology and Science, Mahabubnagar – 204, AP, India

Fast and Efficient Technique for Mining Frequent Closed Sequences

P.LAKSHMI DEEPTHI[1]
Assistant Professor
Dept. of CSE
MLRIT, Dundigal.
Contact: 9573864306
Email.id: deepthi1988@gmail.com

N.ARAVIND KUMAR[2]
Assistant Professor
Dept. of CSE
MLRIT, Dundigal.
Contact: 9704391437
Email.id: aravind.n1205@gmail.com

ABSTRACT

Frequent patter mining has been around ever since the data mining domain came into popularity. However, there was an argument in this domain that one should min only closed frequent patterns instead of mining all frequent patterns. The reason being that the former gives rise to very compact and efficient result set. The drawback of most of the existing closed sequence mining algorithms is that they rely on an approach known as candidate maintenance-and-test. This approach is costly in terms of memory usage and the time taken. To overcome these drawbacks, this paper presents a new algorithm that can effectively mine closed frequent sequence patterns without the need for maintaining candidates. It also prunes search space for reducing cost and time of the operations. Extensive experiments are made with many real world datasets. The experimental results reveal that the proposed algorithm and pruning techniques are efficient.

INTRODUCTION

Among many techniques of data mining sequential pattern mining is one of the important tasks. It was introduced in [1]. It has many applications in real time including discovering patterns from protein sequences, analyzing web log data, analyzing markets and customers etc. besides its ability to mine XML query access patterns for the purpose of caching. Sequential patterns are the patterns that contain set of items repeated sequentially in the given dataset. By mining the sequential patterns it is possible to use the results for analysis and taking well informed decisions. The result of such data mining is the required business intelligence that leads to profits to enterprises. Previously many researchers studied various efficient mining techniques. They include sequential patter mining techniques, long sequential pattern mining in noisy environment, partial periodical pattern mining, temporal relation mining, cyclic association rule mining, frequent episode mining and constraint –based sequential pattern mining. Extensive research has been made on general sequential pattern mining and presented in [2], [3], [4] and [5]. The research of constraint based sequential pattern mining is focused in [6], [7] and [8]. Frequent episode mining also became famous which is presented in [9]. A variant of association rule mining namely cyclic association rule mining is presented in [10]. Mining which considers temporal information known as temporal relation mining is found in [11]. Periodic patterns might be found partially such technique is discussed in [12] while in [13] long sequential pattern mining is discussed in noisy environment.

Recently an argument came into existence in data mining domain that is it is not good to mine all frequent patterns. Instead of that, min only closed ones as they provide more compact and complete with better efficiency. This argument was focused in [14], [15], and [16]. For mining closed sequential patterns, very less number of methods is available. The reason for this might be the complexity of this problem of closed sequential patterns mining. CloSpan is a well known method for the same purpose. This method was described in [17]. Like many other algorithms, it also follows a paradigm known as “candidate maintenance – and – test. It does mean that it performs mining and maintains candidates which are the result of already mined closed sequence patterns. Those candidates are used to effectively prune the search space and determine the new patterns of sequential in nature are closed or not. Despite its effectiveness, the closed pattern mining algorithms have less scalability with respect to number of closed patterns. The reason for this is that it needs more memory and involves larger search space. From this problem it is essential to find a way that eliminates the maintenance of candidates which appears to be a difficult task. This paper provides a good solution for this using an algorithm that is capable of mining frequent closed sequences effectively. In the proposed algorithm, the historical patterns are not kept track of for the purpose of checking new patterns. This leads to reducing search space or pruning more search space by using optimization techniques. The empirical results revealed that the proposed algorithm is very effective in terms of memory and time.

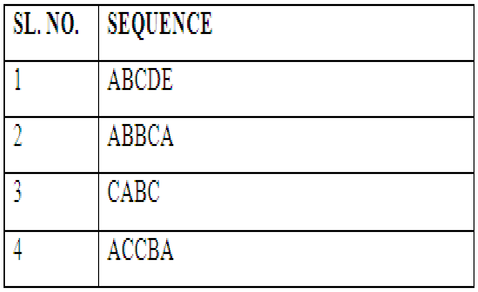
RELATED WORK

The problem of sequential pattern mining was initially proposed in [1] which was later refined by the same authors and presented in [18]. These implementations were done by using Apriori property [19]. For performance improvement many such algorithms came into existence. Among them some of the interesting ones are SPAM [2], PrefixSpan [20] and SPADE [5]. Lattice –theoretic approach is used by SPADE where the search space is divided into small pieces. Representation of horizontal format dataset was adapted by PrefixSpan which makes use of pattern – growth approach for mining sequential patterns. To deal with long sequential patterns, it uses prefix pattern. When compared with other approach by name GSP, the SPADE and PrefixSpan are much better ones. Long sequential patterns are obtained by SPAM. When it comes to mining long sequential patterns, the SPAM is better than the other two approaches such as PrefixSpan and SPADE.

The technique of frequent closed item set mining was introduced in [14]. Afterwards many such algorithms came into existence namely CLOSET+ [15], CHARM [16], A-Close [21]. For checking patterns, they use already mined frequent closed patterns. Hash indexed result tree is used by algorithms like CLOSET+2 and TFP [21] for the purpose of reducing search space for pruning and memory usage. Out of all frequent closed sequence techniques, CloSpan [17] is the recent one whose approach to solve the problem is candidate maintenance-and-test. According to that approach, it generates candidate sequences first and then follows certain post pruning techniques. Pruning methods used by this technique are Backward Sub-Pattern Pruning and Common Prefix. However, the CloSpan consumes more memory and deal with huge search space thus results in poor scalability. Out contributions in this paper include a new algorithm for frequent closed sequences; optimization techniques like ScanSkip and BackScan pruning methods; performance study to evaluate the algorithm and pruning techniques.

Problem Definition

In many real world applications, sequence patterns play an important role. In a data set an item can occur multiple times sequentially. The number of times the item set/item occurs sequentially is known as the length of the sequence. Table1 shows sample database with sequences.



SL. NO.	SEQUENCE
1	ABCDE
2	ABBCA
3	CABC
4	ACCBA

Table 1 – Sample sequence database

The purpose of the algorithm proposed in this paper is to find out closed sequence patterns from the given dataset. The proposed algorithm is meant for mining only closed frequent sequence patterns.

PROPOSED ALGORITHM

The proposed algorithm finds closed frequent sequence patterns effectively. It can also prune search space and also reduce the cost of computations. The proposed algorithm follows the steps given below.

1. First of all, it scans the database to find the frequent 1-sequences.
2. For each such sequence, it builds pseudo projected database and treats each sequence as a prefix.
3. BackScan pruning approach is used to find each one can be pruned.
4. If that can't be proved, it then computes the number of backward-extension-items

ISSN 2278-3091

and then the following steps are recursively executed.

5. Projected database is scanned to find local frequent patterns.
6. Calculates the number of forward-extension-items
7. Calculate the number of backward-extension-items
8. If there is not forward-extension-items and also backward-extension-items then
9. Output the prefix as frequent closed sequence
10. Then build pseudo projected database for the new prefix
11. Check if the new prefix can be pruned. If not repeat the process from step 5.

EXPERIMENTAL RESULTS

A prototype application is built to demonstrate the efficiency of the proposed algorithm and pruning methods. The environment used for the application development includes Windows XP OS, JDK 1.6, Net Beans with a PC containing 2GB RAM and 2.93 GHz. Other algorithms' results are compared with proposed algorithm. Three different datasets are used in the experiments. They are Gazelle, Snake and Pi.

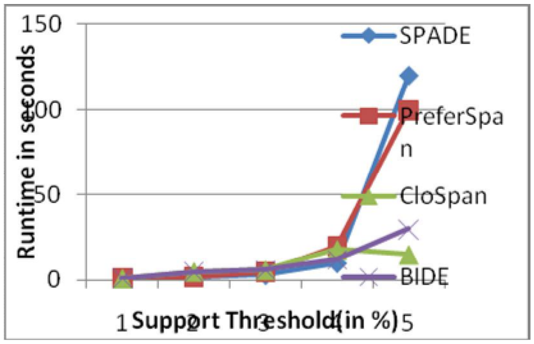


Fig. 1 – Comparison among algorithms

As can be seen in fig. 1, the proposed algorithm is compared with other algorithms such as SPADE, PerferSpan, and CloSpan. The horizontal axis represents support threshold while the vertical axis shows the runtime in milliseconds. These experiments are made on Gazelle dataset.

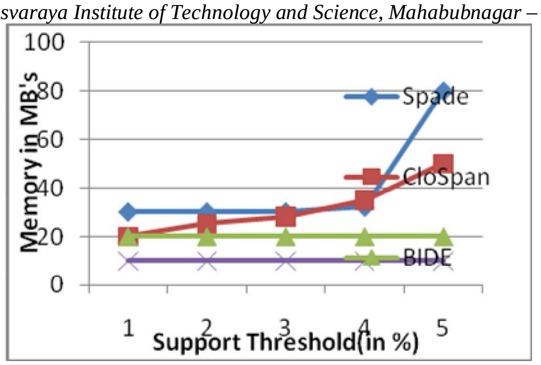


Fig. 2 – Comparison among algorithms

As can be seen in fig. 2, the proposed algorithm is compared with other algorithms such as SPADE, PerferSpan, and CloSpan. The horizontal axis represents support threshold while the vertical axis shows the memory used in MBs. These experiments are made on Gazelle dataset.

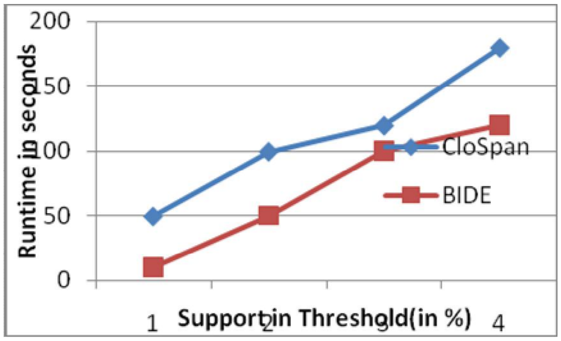


Fig. 3 – Comparison between CloSpan and Proposed Algorithm

As can be seen in fig. 3, the proposed algorithm is compared with CloSpan. The horizontal axis represents support threshold while the vertical axis shows the runtime in milliseconds. These experiments are made on Snake dataset.

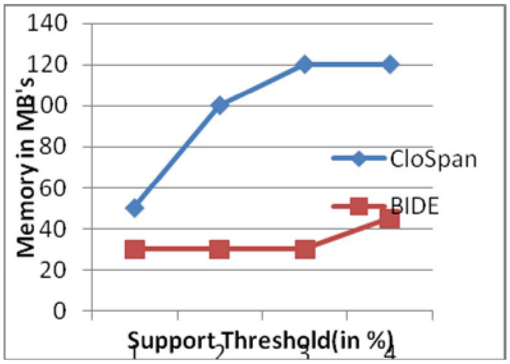


Fig. 4 – Comparison between CloSpan and Proposed Algorithm.

As can be seen in fig. 4, the proposed algorithm is compared with CloSpan. The horizontal axis represents support threshold while the vertical axis shows the Memory in MBs. These experiments are made on Snake dataset.

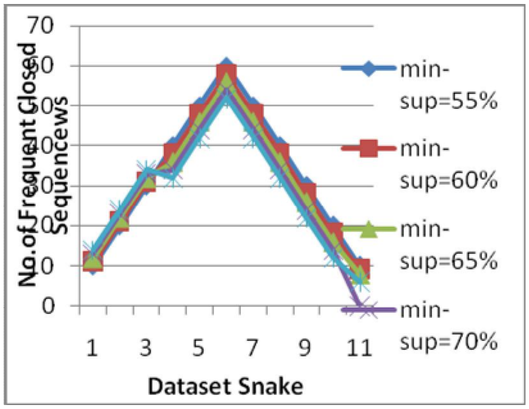


Fig. 5 – Experimental Results with Snake Dataset

As can be seen in fig. 5, the proposed algorithm is compared with CloSpan. The horizontal axis represents Snake dataset while the vertical axis shows the number of frequent closed sequences. These experiments are made with various minimum support values.

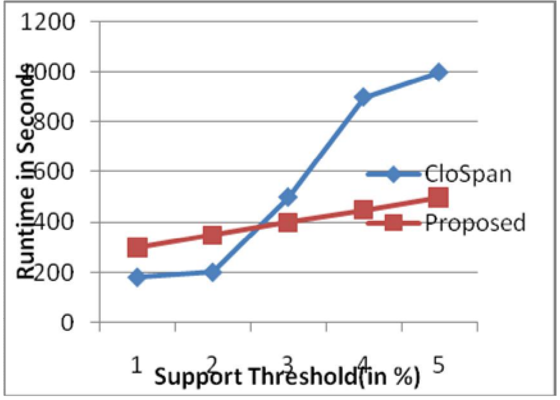


Fig. 5 – Comparison between CloSpan and Proposed Algorithm using Snake Dataset

As can be seen in fig. 5, the proposed algorithm is compared with CloSpan. The horizontal axis represents support threshold while the vertical axis shows the runtime in milliseconds. These experiments are made on Pi dataset.

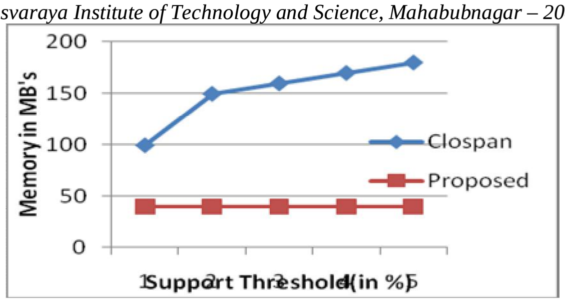


Fig. 6 – Comparison between CloSpan and Proposed using Snake Dataset

As can be seen in fig. 6, the proposed algorithm is compared with CloSpan. The horizontal axis represents support threshold while the vertical axis shows the memory in MBs. These experiments are made on Pi dataset.

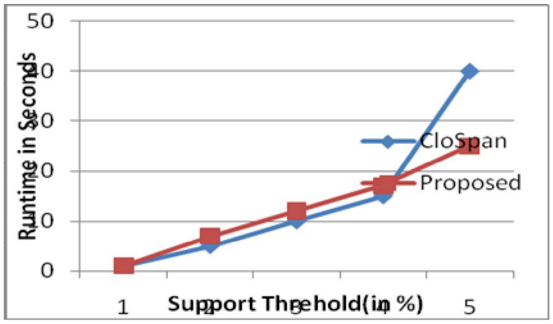


Fig. 7 – Comparison between CloSpan and Proposed using Pi Dataset

As can be seen in fig. 7, the proposed algorithm is compared with CloSpan. The horizontal axis represents support threshold while the vertical axis shows the runtime in milliseconds. These experiments are made on Pi dataset.

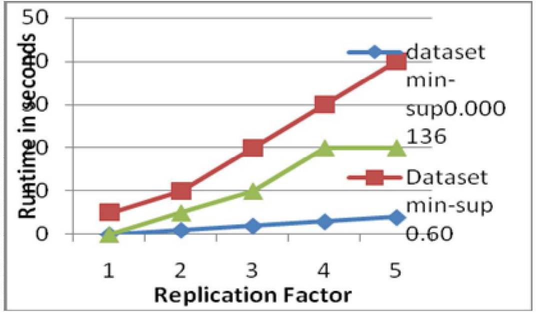


Fig. 8 – Results of Scalability Test

As can be seen in fig. 8, the proposed algorithm is tested for its scalability. The horizontal axis represents replication factor while the vertical axis shows the runtime in milliseconds. These experiments are made on different datasets with different minimum support values.

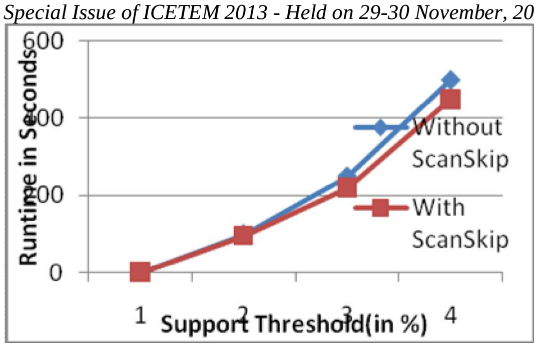


Fig. 9 – Results of ScanSkip Optimization

As can be seen in fig. 9, the ScanSkip optimization performance is presented. The horizontal axis represents support threshold while the vertical axis shows the runtime in milliseconds. These experiments are made on different datasets with different minimum support values.

CONCLUSION

This paper presents an efficient closed frequent pattern mining algorithm that proves the fact that the closed frequent pattern mining is as effective as all frequent pattern mining and also provides result set which is more impact and effective. The existing closed frequent pattern mining algorithms are costly and consume more memory as they maintain candidates while processing. This paper proposed an algorithm that eliminates the needs for such approach. Thus the proposed algorithm and pruning techniques reduce the time taken and also the computational cost. The experiments have been made with three different real world data sets. Those datasets are both sparse and dense. A prototype application is built to demonstrate the efficiency of the proposed algorithm. The results revealed that the proposed approach is more efficient than many existing algorithms of same kind.

REFERENCES

[1] R. Agrawal, and R. Srikant, *Mining sequential patterns*. In ICDE'95, Taipei, Taiwan, Mar. 1995.

[2] J. Ayres, J. Gehrke, T. Yiu, and J. Flannick, *Sequential PAttern Mining using a Bitmap Representation*. In SIGKDD'02, Edmonton, Canada, July 2002.

[3] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M.C. Hsu, *FreeSpan: Frequent pattern-projected sequential pattern mining* . In SIGKDD'00, Boston, MA, Aug. 2000.

[4] F. Masseglia, F. Cathala, and P. Poncelet, *The psp approach for mining sequential patterns*. In PKDD'98, Nantes, France, Sept. 1995.

[5] M. Zaki, *SPADE: An Efficient Algorithm for Mining Frequent Sequences*. Machine Learning, 42:31-60, Kluwer Academic Pulishers, 2001.

[6] M. Garofalakis, R. Rastogi, and K. Shim, *SPIRIT: Sequential PAttern Mining with regular expression constraints*. In VLDB'99, San Francisco, CA, Sept. 1999.

[7] J. Pei, J. Han, and W. Wang, *Constraint-based sequential pattern mining in large databases*. In CIKM'02, McLean, VA, Nov. 2002.

[8] M. Seno, G. Karypis, *SLPMiner: An algorithm for finding frequent sequential patterns using lengthdecreasing support constraint*. In ICDM'02,, Maebashi, Japan, Dec. 2002.

[9] H. Mannila, H. Toivonen, and A.I. Verkamo, *Discovering frequent episodes in sequences* . In SIGKDD'95, Montreal, Canada, Aug. 1995.

[10] B. Ozden, S. Ramaswamy, and A. Silberschatz, *Cyclic association rules*. In ICDE'98, Olando, FL, Feb. 1998.

[11] C. Bettini, X. Wang, and S. Jajodia, *Mining temporal relationals with multiple granularities in time sequences*. Data Engineering Bulletin, 21(1):32-38, 1998.

[12] J. Han, G. Dong, and Y. Yin, *Efficient mining of partial periodic patterns in time series database*. In ICDE'99, Sydney, Australia, Mar. 1999.

[13] J. Yang, P.S. Yu, W. Wang and J. Han, *Mining long sequential patterns in a noisy environment*. In SIGMOD' 02, Madison, WI, June 2002.

[14] N. Pasquier, Y. Bastide, R. Taouil and L. Lakhal, *Discoving frequent closed itemsets for association rules*. In ICDT'99, Jerusalem, Israel, Jan. 1999.

[15] J. Wang, J. Han, and J. Pei, *CLOSET+: Searching for the Best Strategies for Mining Frequent Closed Itemsets*. In KDD'03, Washington, DC, Aug. 2003.

[16] M. Zaki, and C. Hsiao, *CHARM: An efficient algorithm for closed itemset mining*. In SDM'02, Arlington, VA, April 2002.

[17] X. Yan, J. Han, and R. Afshar, *CloSpan: Mining Closed Sequential Patterns in Large Databases*. In SDM'03, San Francisco, CA, May 2003.

[18] R. Srikant, and R. Agrawal, *Mining sequential patterns: Generalizations and performance improvements*. In EDBT'96, Avignon, France, Mar. 1996.

[19] R. Agrawal and R. Srikant. *Fast algorithms for mining association rules*. In VLDB'94, Santiago, Chile, Sept. 1994.

[20] J. Pei, J. Han, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M.C. Hsu, *PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth*. In ICDE'01, Heidelberg, Germany, April 2001.

[21] J. Han, J. Wang, Y. Lu, and P. Tzvetkov, *Mining Top- K Frequent Closed Patterns without Minimum Support*. In ICDM'02, Maebashi, Japan, Dec. 2002.