ISSN:0975-9646

A Framework for Mining Closed Sequential Patterns

V. Purushothama Raju[1], G.P. Saradhi Varma[2]

[1]*Research Scholar, Dept. of CSE, Acharya Nagarjuna University*
Guntur, A.P., India

[2]*Department of Information Technology*
S.R.K.R. Engineering College, Bhimavaram, A.P., India

***Abstract*—Sequential pattern mining algorithms developed so far provide better performance for short sequences but are inefficient at mining long sequences, since long sequences generate a large number of frequent subsequences. To efficiently mine long sequences, closed sequential pattern mining algorithms have been developed. These algorithms mine closed sequential patterns which don't have any super sequences with the same support. Closed sequential patterns are more compact comparing to the patterns produced by the sequential pattern mining algorithms. In this paper, we propose a framework for mining closed sequential patterns by integrating the best features of SPAM and CHARM. Our algorithm is the first method that utilizes vertical bitmap data structure for closed sequential pattern mining.**

***Keywords*—Data Mining, Sequential Pattern Mining, Closed Sequential Pattern Mining.**

I. INTRODUCTION

Sequential Pattern Mining (SPM) was first introduced by R. Agrawal and R. Srikanth in [1] and it has become an important data mining task. Applications of SPM include mining customer shopping sequences, DNA sequences and Web click streams, finding copy-paste and related bugs in large software, mining API usages from open source software and network intrusion detection.

Several SPM algorithms were proposed to mine short sequences but they are inefficient at mining long sequences. Long sequences generate exponential number of sub sequences, for example a long frequent sequence $\{(x_1)(x_2)....(x_{50})\}$ will generate 2^{50} - 1 subsequences. The performance of SPM algorithms degrade when mining at low support values.

Closed sequential pattern mining was proposed to overcome the limitations of SPM algorithms. Closed sequential pattern mining produces more compact result set than SPM and also offers better efficiency for mining long sequences. Only a few algorithms were proposed for mining closed sequential patterns, this is due to the complexity of the problem.

There are two ways to mine closed sequential patterns. The first approach is greedily finding the final closed sequential patterns; this approach is more complicated because it is hard to verify the closeness of pattern without checking with the previously discovered patterns. The second approach is to find the closed sequential pattern candidate set and conduct post pruning on it, this approach requires storing the discovered patterns but with recent advances in technology we can store million patterns in main memory. We follow the second approach in this paper.

In this paper, we propose an efficient algorithm CSPM (closed sequential pattern miner) for mining closed sequential patterns by integrating the best features of SPAM [7] and CHARM [13]. Our algorithm is the first method that utilizes vertical bitmap data structure for closed sequential pattern mining and it outperforms CloSpan by an order of magnitude.

The rest of this paper is organized as follows. Section 2 discusses the related work. In Section 3, we present the problem definition. Section 4 presents the proposed method. Section 5 reports the performance evaluation. Finally, we conclude the work in Section 6.

II. RELATED WORK

Closed sequential pattern mining is related to sequential pattern mining and closed itemset mining. Several algorithms were proposed for sequential pattern mining, the efficient algorithms are SPADE [5], PrefixSpan [6] and SPAM [7]. SPADE adopts breadth-first search where as PrefixSpan and SPAM adopt depth-first search. SPADE adopts a vertical data format and mines the sequential patterns through a simple join on id-lists. PrefixSpan adopts a horizontal data format and mines the sequential patterns under the pattern growth paradigm. SPAM mines sequential patterns using vertical bitmap representation and it outperforms PrefixSpan and SPADE on large datasets. However, SPAM requires more memory than the other two methods.

Closed itemset mining algorithms CLOSET [12] and CHARM [13] adopt space efficient depth first search. CLOSET adopts a compressed database representation called FP-tree to mine closed itemsets. CHARM adopts a compact vertical tid list structure called diffset to mine closed itemsets.

There are only two popular algorithms in closed sequential pattern mining CloSpan [8] and BIDE[11]. CloSpan produces a candidate set for closed sequential patterns and performs post pruning on it. CloSpan requires more storage to store the closed sequence candidates when mining long patterns or the support threshold is low and it offers poor scalability. BIDE adopts the framework of PrefixSpan and uses BackScan pruning method to stop growing redundant patterns. BIDE is a computational intensive approach since it requires more number of database scans for the bi-direction closure checking and the BackScan pruning.

III. Problem Definition

Let $I = \{i_1,i_2,\ldots.,i_m\}$ be a set of all items. A subset of *I* is called an itemset. A sequence S = $(k_1,k_2,\ldots, k_n)$ $(k_i \subseteq I)$ is an ordered list of itemsets. The items in each itemset are sorted in alphabetic order. The length of the sequence is the total number of items in the sequence. A sequence $S_1 = (a_1,a_2,\ldots..,a_m)$ is a subsequence of another sequence $S_2 = (b_1,b_2,\ldots.,b_n)$, denoted as S_1 S_2, if there exit integers $1 \le i_1 < i_2 < \ldots < i_m \le n$ and $a_1 \subseteq b_{i1}$, $a_2 \subseteq b_{i2}$, . . . , and $a_m \subseteq b_{im}$. We call S_2 as a super-sequence of S_1 and S_2 contains S_1.

A sequence database, SD=$\{S_1,S_2,\ldots,S_n\}$, is a set of sequences and each sequence has an id. The size, |SD|, of the sequence database SD is the total number of sequences in the SD. The support of a sequence α in a sequence database SD is the no of sequences in SD which contain α.

Given a minimum support threshold m_sup, a sequence α is a sequential pattern on SD if support of α is greater than m_sup. We call a sequence α as a closed sequential pattern If α is a sequential pattern and there exists no proper super sequence of α with the same support. *The problem of closed sequential pattern mining is to find the complete set of closed sequential patterns above a minimum support threshold m_sup for an input sequence database SD.*

TABLE I. A Sample Sequence Database

S.Id	Sequence
1	(ab)(bcd)(de)
2	(f) (abc)(cd)
3	(bc)(abc)
4	(de)(ag)(bcd)(f)

Table 1 shows a sample sequence database. The items in each itemset are sorted in alphabetic order. If m_sup=2 , the closed sequential pattern set contains 14 sequences {**(a):4, (f):2, (ab):3, (b)(c):3, (bc):4, (d)(d):2, (de):2, (bc)(d):2, (abc):2, (bc)(c):2, (a)(cd):3, (b)(bc):2, (ab)(cd):2, (a)(bcd):2**} and the corresponding sequential pattern set contains 34 sequences. It indicates that closed sequential pattern set contains less no of sequences than sequential pattern set.

IV. Proposed method

Our proposed method integrates the features of SPAM and CHARM. SPAM is based on the Apriori property and it is developed to work with data in main memory. It uncovers all sequential patterns within a transactional database.

SPAM is the first sequential pattern mining technique that employs a depth-first approach to explore the search space. It uses an efficient pruning method that reduces the number of candidates to make it suitable for very long sequential patterns. It stores data using a vertical bitmap representation that permits efficient support counting and considerable bitmap compression.

Our proposed algorithm CSPM is shown in figure1. First it scans the database to remove infrequent items, empty sequences and sorts each itemset of a sequence in SD. Constructs a vertical bitmap for each item in the database and each bitmap contains a bit for each element in the sequence of the database. If there is an item in an element then the bit corresponding to the element of the bitmap for the item is set to *one*; otherwise, the bit is set to *zero*.

```
Algorithm: CSPM
Input: A sequence database SD and minimum support
        min_sup.
Output: The complete set of closed sequential patterns.
1. Remove infrequent items, empty sequences and sort
   each itemset of a sequence in SD.
2. Scan the database and construct vertical bitmap for each item
     in the database.
3. Initialize the bitmaps by setting the bits corresponding to the
     sequences.
4. Construct lexicographic sequence tree.
5. Perform depth first search on lexicographic sequence tree.
6. Perform S-step and I-step at each node in the
   lexicographic sequence tree and adjust bitmaps.
7. Perform S-step pruning and I-step pruning to reduce
   search space.
8. Retain sequential patterns that satisfy the min_sup.
9. Eliminate nonclosed sequential patterns.
```

Fig. 1 CSPM Algorithm

The main idea is to generate all candidate sequences by performing the depth first traversal in the lexicographic sequence tree. Each node of the tree denotes a sequential pattern α discovered so far. Super sequences of α can be produced by using either the *sequence-extension step* (*S-step*) or the *itemset-extension step* (*I-step*). In S-step, a new itemset with one item is appended at the end of α. In I-step, one new item is appended at the end of the last itemset of α.

To extend the bitmap partition of a sequence, all bits after the first bit with value one are set to one in the S-step to produce transformed bitmap. The resultant bitmap of the S-step is created by performing logical AND operation on the transformed bitmap and of the appended item bitmap. Whereas, the resultant bitmap of the I-step is created by performing logical AND operation on the bitmaps of the sequence and the appended item. The support value is obtained by counting the bitmap partitions that contain all ones.

Two Apriori-based pruning techniques S-step pruning and I-step pruning are used to reduce the search space. Both pruning techniques work as follows. The item *i* will not be extended to any super sequence of S if the result of extending an item *i* with S is infrequent based on the Apriori principle.

The depth-first search on S is not repeated if the support of a child node of S is less than *min_sup*. A newly created sequence is pruned if its support is less than *min_sup*. The sequence is stored and depth-first search is done recursively if the support of a child node S is greater than or equal to *min_sup.*

To eliminate the nonclosed sequential patterns, we have to check for each sequence S, if there exists a super sequence S′ such that support(S) = support(S′). We implement the fast subsumption checking algorithm proposed by Zaki[13] for eliminating nonclosed sequential patterns. It maintains sequences in a hash table and the employs support of a sequence as its hash function. CSPM first finds all the sequences that have the same support of S, then it checks if there is a super-sequence containing S to eliminate the nonclosed sequential patterns.

V. Performance Evaluation

In our experiments we used the MSNBC dataset. It is a click stream data obtained from the UCI repository. The original dataset contains 989,818 sequences. Here the shortest sequences have been removed to keep only 31,790 sequences. The number of distinct items in this dataset is 17. The average number of itemsets per sequence is13.33. The average number of distinct item per sequence is 5.33. The characteristics of the dataset are given in Table 2.

TABLE II. Characteristics of the Dataset

S. No.	Characteristic	Value
1	No of sequences	31790
2	No of distinct items	17
3	Average no of itemsets per sequence	13.33

The experiments are conducted on a 2GHz Intel Core2 Duo processor with 1GB main memory running Windows XP. The algorithm is implemented in Java and it is executed using different support values on MSNBC dataset to find out closed sequential patterns. The Fig. 2 shows the performance comparison between CloSpan and CSPM algorithm. Our proposed algorithm CSPM runs faster than CloSpan.

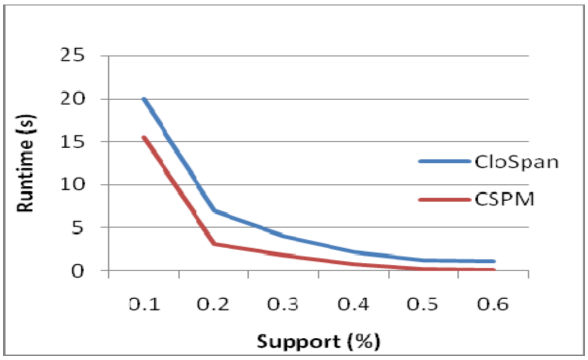

Fig. 2. Runtime v/s Support

VI. Conclusions

In this paper, we proposed an efficient algorithm CSPM for mining closed sequential patterns in large data sets. The closed sequential pattern mining has the same expressive power of sequential pattern mining and also produces more compact result set. Our proposed algorithm CSPM outperforms CloSpan by an order of magnitude. Other interesting research problems that can be pursued in this area include parallel mining of closed sequential patterns and mining of structured patterns.

References

[1] R. Agrawal and R. Srikant, "Mining Sequential Patterns," *Proceedings of ICDE '95,* pp. 3-14, Mar. 1995.

[2] R. Srikant and R. Agrawal, "Mining Sequential Patterns: Generalizations and Performance Improvements," *Proceedings of EDBT '96,* pp. 3-17, Mar. 1996.

[3] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," *Proceedings of ACM SIGMOD '00,* pp. 1-12, May 2000.

[4] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M.C. Hsu, "FreeSpan: Frequent Pattern-Projected Sequential Pattern Mining," *Proceedings of ACM SIGKDD '00,* pp. 355-359, Aug. 2000.

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

[6] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, "PrefixSpan : Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth," *Proc. Int'l Conf. Data Engineering (ICDE '01),* pp. 215-224, Apr. 2001.

[7] J. Ayres, J. Gehrke, T. Yiu, and J. Flannick, "Sequential Pattern Mining Using a Bitmap Representation," *Proceedings of ACM SIGKDD '02,* pp. 429-435, July 2002.

[8] X. Yan, J. Han, and R. Afshar, "CloSpan: Mining Closed Sequential Patterns in Large Databases," *Proceedings of SIAM's SDM '03,* pp. 166-177, May 2003.

[9] P. Tzvetkov, X. Yan, and J. Han, "TSP: Mining Top-K Closed Sequential Patterns," *Proceedings of ICDM '03,* pp. 347-354, Dec. 2003.

[10] S. Cong, J. Han, and D.A. Padua, "Parallel Mining of Closed Sequential Patterns," *Proceedings of ACM SIGKDD '05,* pp. 562-567, Aug. 2005.

[11] J. Wang, J. Han, and Chun Li, "Frequent Closed Sequence Mining without Candidate Maintenance," *IEEE TKDE.,* vol. 19, no. 8, pp. 1042-1056, Aug. 2007.

[12] J. Pei, J. Han, and R. Mao, "CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets," *Proceedings of ACM DMKD '00,* pp. 21-30, May 2000.

[13] M. Zaki and C. Hsiao, "CHARM: An Efficient Algorithm for Closed Itemset Mining," *Proceedings of SIAM's SDM '02,* pp. 457-473, Apr. 2002.

[14] J. Han, J. Wang, Y. Lu, and P. Tzvetkov, "Mining Top-K Frequent Closed Patterns without Minimum Support," *Proceedings of ICDM '02,* pp. 211-218, Dec. 2002.

[15] J. Wang, J. Han, and J. Pei, "CLOSET+: Searching for the Best Strategies for Mining Frequent Closed Itemsets," *Proceedings of ACM SIGKDD '03*, pp. 236-245, Aug. 2003.