Fast Mining Maximal Sequential Patterns

NANCY P. LIN[1], WEI-HUA HAO[1], HUNG-JEN CHEN[1,2],
HAO-EN CHUEH[1], CHUNG-I CHANG[1]

[1]Department of Computer Science and Information Engineering
Tamkang University,
151 Ying-Chuan Road, Tamsui, Taipei,
TAIWAN, R.O.C.
[2]Department of Industrial Engineering and Management
St. John's University,
499, Sec. 4, Tam-King Road, Tamsui, Taipei,
TAIWAN, R.O.C.

Abstract: - Sequential patterns mining is now widely used in many areas, such as the analysis of e-Learning sequential patterns, web log analysis, customer buying behavior analysis and etc. In the discipline of data mining, runtime and search space are always the two major issues. In this paper, we had study many previous works to analyze these two problems, and propose a new algorithm with more condense structure and faster process to find out the complete frequent sequential patterns.

Key-Words: - Data mining, Sequential patterns, Maximal sequential patterns, Lattice structure, Sequence data base

1 Introduction

Sequential Patterns has divers' applications in many field recently. Han and Kamber[17] defined :*A sequence database consists of sequences of ordered elements or events*. And it is one of the most important domains of Data Mining. The major problem in previous works of this field is that generate too many candidates sequences during the mining process, which has increase the requirement of hardware and system runtime. In this paper we propose a new algorithm, Fast Mining Maximal Sequential Patterns(FMMSP), to alleviate this problem. Mining sequential patterns is a task of finding the full set of frequent sequences that satisfy a given minimum support in a sequence database.

Sequential pattern mining has gradually become an inportant data mining task, with broad applications, including market and customer analysis, web log analysis, and mining XML query access patterns.

In this paper, we propose a solution to mine maximal sequences, rather than mining the full set of frequent sequences.

The reason why we mine maximal sequential patterns is that they are compact representations of frequent sequential patterns. Sequential Patters Mining was first introduced by Agrawal and Srikant in [1]: *Given a set of sequences, where each sequence consists of a list of itemsets, and given a user-specified minimum support threshold (min support), sequential pattern mining is to find all frequent subsequences whose frequency is no less than min support*. This mining algorithm has a consequence of the following problems: sequential pattern mining often generates huge number of candidate patterns in an exponential curve, which is inevitable when the database consists of long frequent sequential patterns. For example, assume the database contains a frequent sequence $\langle i_1,\ldots,i_k \rangle$, k=20, it will generate 2^{20} -1 frequent subsequences which are essentially redundant patterns.

Mining sequential patterns with maximal sequential patterns may largely reduce the number of patterns generated in the process and without losing any information because it can be used to derive the complete set of sequential patterns.

2 Preliminary

The problem can be described as follows: Assume that $I=\{ i_1, i_2,\ldots, i_{|I|}\}$ is a finite items set and D is a data set containing n transactions, each transaction $s \in D$ is a sequence of distinct items $s=\langle i_1,\ldots,i_{|s|} \rangle$, in

which i,j∈I. Let S be a k-items sequence, where S = 〈i_1,…,i_k〉 is a sequence of k distinct items i, j∈I. Given a k-items sequence s, let its support be supp(s) which is defined as the number of transactions in D that include s. To mine all the frequent sequences from D requires finding all the sequences that support no less than the minimum support and this has to search through the huge search space which is given by the power set of I.

3 The FMMSP algorithm

FMMSP is composed of 3 phases: Growing Phase, Pruning Phase and Maximal Phase. In Growing Phase, sequences are read from database into a lattice structure with all its subsequences, if there is any. Second, in Pruning Phase, prune off those infrequent sequences in the lattice. Finally, Maximal Phase, delete each node's subsequences to find out the maximal frequent sequences. According the Closed-downward principle, all subsequences of maximal frequent sequences are also frequent. A structure of maximal frequent sequences is a lossless method to contain original information.

```
==================================
//Input: D, minsup
//Output: Maximal Sequences

// Growing Phase
Initiate Lattice  // create root node
ConstructLattice(D,minsup){
   Repeat steps listed below until the end of D{
      read sequence SP from D
      if SP ∈ Lattice {
         search node that node.sequence=SP;
         if node.frequent=FALSE
            node.count++;
         if node.count≥minsup
         For this node and all of it's subsequence
                  node.frequent=TRUE;
      }
      else {
         new node;
         node.sequence=SP;
         node.count=1;
         //construct all subsequences nodes of this new
         node;
         ConstructLattice(subsequences of node,
         minsup);
      }
     }
     }
//Pruning upward
For (k=1:k≤
LengthOfLongestFrequenceSequence-1:k++)
   For all nodes of length k
      If node.frequency < minsup{
         Delete node;
          Delete all supersequences of node
      }
// Maximal sequence phase
For each node
         Delete all subsequences nodes
Maximal sequences = all nodes remain in Lattice
==================================
```

Fig 1 FMMSP algorithm

4 example

We will demonstrate how the FMMSP is capable to minimize the searching space and accelerate runtime with example. In table 1 is a simple sequence database table. SID represents Student Identifier. The itemset including A, B, C, D and E.

Table 1 sequence database

SID	Sequence
1	ACD
2	ABCE
3	BCE
4	BE

GROWING PHASE: read in a sequence 〈ACD〉 from database. Link the sequence node to root node and all its subsequence are linked to this node and so on so forth. This lattice is growing into a 8 nodes lattice including root node, 〈ACD〉, 〈AC〉, 〈AD〉, 〈CD〉, 〈A〉, 〈B〉 and 〈C〉. Each node's accumulator has increase 1 from 0.

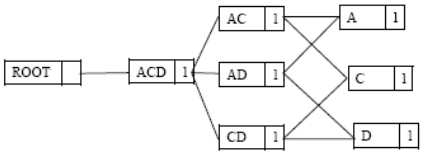

Fig. 2 Lattice contains 〈ACD〉

After that, continue read in 〈ACD〉 and 〈ABCE〉 as show in Fig. . The accumulator of 〈AC〉 has reached the minimum support, it is qualified being a frequent sequence, node are mark with a wide red frame, so does nodes of 〈A〉 and 〈C〉 due to the Closed-Downward principle. The number in the middle of linking line represents SID number. Wide arrow represents closed-downward relationship.

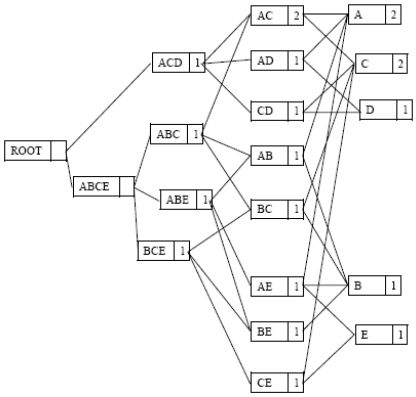

Fig. 3 Lattice including〈ACD〉and 〈ABCE〉

The lattice after reading〈BCE〉and〈BE〉is shown in fig. White nodes are denote as frequent sequence and all it's subsequences pointed with arrow are frequent sequences as well. Hence, FMCSP can accelerate the runtime of constructing Lattice.

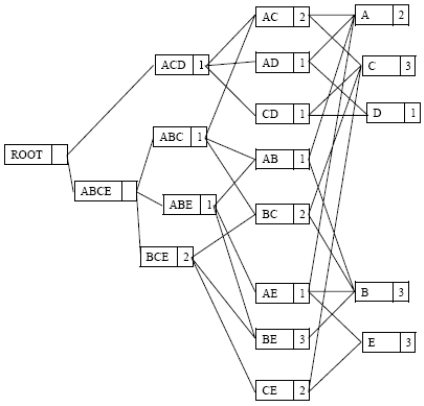

Fig. 4 Complete Lattice

Pruning Phase: Scan Lattice from short sequence nodes, delete infrequent sequence nodes and its suppersequences. A pruned lattice is shown in Fig 5。

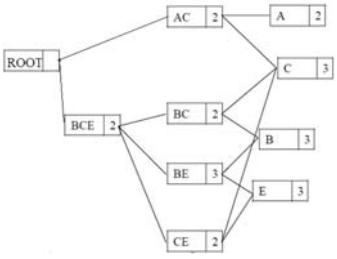

Fig. 5 Lattice of frequent sequence

Maximal Sequences: Scan from long sequence nodes, delete each node's subsequence. The remaining sequences are〈BCE〉and 〈AC〉called maximal sequences. The remaining lattice is called maximal sequence lattice, shown as Fig. 6.

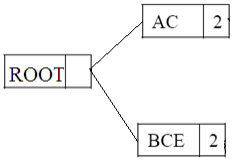

Fig.6 Maximal Sequence Lattice

Compare to previous works the advantages are: smaller search space and faster search speed via the preknowledge of minimum support.

5 Conclusion

Unfortunately, Apriori-like algorithms may fail to extract all the frequent sequences from dense data sets, which contain strongly correlated sequences and long frequent sequential patterns. Such data sets are, in fact, very hard to mine since the Apriori closed-downward principle does not guarantee an effective pruning of candidates, while the number of frequent sequences grows up very quickly as the minimum support threshold is decreased.

Many studies have incept the concept to elaborate all frequent pattern mining to more compact results and significantly better efficiency of memory usage. Our study shows that this is usually true when the number of frequent patterns is extremely large, in this case the number of frequent maximal sequential patterns is also tend to be very large. In this paper, we proposed FMMSP, a novel algorithm for mining frequent maximal sequential sequences. It has improved the drawback of the *candidate maintenance-and-test* paradigm, constructing more compact searching space compare to the previously developed closed pattern mining algorithms. FMMSP adopts a breadth-first method can output the frequent closed patterns online.

References:

[1] R. Agrawal and R. Srikant. Mining sequential patterns. In Proc. 1995 Int. Conf. Data Engineering (ICDE'95), pages 3–14, Taipei, Taiwan, Mar. 1995.

[2] C. Lucchese, S. Orlando and R. Perego, Fast and Memory Efficient Mining of Frequent Closed Itemsets, IEEE Transactions on Knowledge and Data Engineering, Vol. 18, No. 1, January 2006.

[3] P. Songram, V. Boonijin and S. Intakosum, Closed Multidimensional Sequential Pattern Mining, Proceeding of the Third Conference on Information Technology: New Generations (ITNG'06).

[4] J. Han, J. Pri, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M.-C. Hsu, FreeSpan: Frequent Pattern-Projected Sequential Pattern Mining, Proc. 2000 ACM SIGKDD Int'l Conf. Knowledge Discovery in Database (KDD '00), pp. 355-359, Aug. 2000.

[5] J. Han, J. Pei and Y. Yin, "Mining Frequent Patterns without Candidate Generation", Proc. 2000 ACM-SIGMOD Int'l Conf. Management of Data (SIGMOD '00), pp.1-12, May 2000.

[6] Wang, J.; Han, J.a, "BIDE: efficient mining of frequent closed sequences", Data Engineering, 2004. Proceedings. 20th International Conference on 30 March-2 April 2004 Page(s):79 – 90.

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

[8] 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.

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

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

[11] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Janyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal, Mei-Chun Hsu, Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach, IEEE Transactions on Knowledge and Data Engineering, vol. 16, No. 11, November 2004.

[12] M. Zaki. SPADE: An efficient algorithm for mining frequent sequences. Machine Learning, 40:31–60, 2001.

[13] Maged El-Sayed, Carolina Ruiz, Elke A. Rundensteiner, Web mining and clustering: FS-Miner: efficient and incremental mining of frequent sequence patterns in web logs Proceedings of the 6th annual ACM international workshop on Web information and data management, November 2004.

[14] 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. 1994.

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

[16] Fast Accumulation Lattice Algorithm for Mining Sequential Patterns, Proceedings of the 6th WSEAS International Conference on Applied Computer Science (ACOS'07), pp. 230-234, Hangzhou, China, April 15-17, 2007.

[17] Jiawei Han and Micheline Kamber, "Data Mining, Concepts and Techniques", 2nd edition, Morgan Kaufmann Published, 2006.