Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Engineering

Elixir Comp. Engg. 76 (2014) 28702-28704

ISSN: 2229-712X

Finding closed sequential patterns in sequence databases

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

[1] Research Scholar, Department of CSE, Acharya Nagarjuna University, Guntur, India.

[2] Department of IT, S.R.K.R. Engineering College, Bhimavaram, India.

ARTICLE INFO

Article history:
Received: 2 October 2014;
Received in revised form:
10 November 2014;
Accepted: 21 November 2014;

Keywords
Data mining,
Sequential pattern mining,
Closed sequential pattern mining,
Sequence database.

ABSTRACT

Sequential pattern mining has been a focused theme in data mining. Sequential pattern mining algorithms provide better performance for short sequences but are inefficient at mining long sequences, since long sequences generate a large number of frequent subsequences. To avoid the limitations of sequential pattern mining algorithms, closed sequential pattern mining algorithms were developed. Closed sequential pattern mining produces less number of patterns and works more efficiently than sequential pattern mining. In this paper, we propose an efficient algorithm CSPgrow to find out closed sequential patterns. To improve the performance, we developed an Extension Checking pruning method. The results indicate that the proposed algorithm CSPgrow outperforms ClaSP.

Introduction

Data mining has become an important subject and attracted the researchers in recent years due to the availability of large amounts of data and the need for converting such data into useful knowledge. Sequential pattern mining was first introduced by R. Agrawal and R. Srikanth in [1]. It aims at discovering frequent subsequences as patterns in a sequence database. Since then, sequential pattern mining has become an important data mining task.

Sequential pattern mining algorithms are mainly classified into Apriori based methods and Pattern growth methods. Apriori based methods require frequent scans of database, generation of candidate sequences and testing. Pattern growth based methods eliminate the above problems and operate on projected database which minimizes the search space.

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

There are two approaches for mining closed sequential patterns. The first approach is greedily finding the final closed sequential patterns and the second approach is to generate closed sequential pattern candidate set and to do post pruning on it. Closed sequential pattern mining has a large number of applications in different domains. The major applications are mining customer shopping sequences, mining biological sequences, mining web click streams, target marketing, personalization systems and web recommender systems.

In this paper, we propose an efficient algorithm CSPgrow to find out closed sequential patterns. To improve the performance, we developed an Extension Checking pruning method. The results show that the proposed algorithm CSPgrow can find closed sequential patterns efficiently and outperforms ClaSP[2]. The rest of this paper is organized as follows. Section 2 discusses the related work. Section 3 presents the problem definition. Section 4 presents the proposed method. Section 5 reports the performance evaluation. Finally, we conclude the work in Section 6.

Related work

Closed sequential pattern mining is related to sequential pattern mining and closed itemset mining. Sequential pattern mining is used to find the complete set of frequent sequences in a sequential database. Sequential pattern mining was first proposed by R. Agrawal and R. Srikanth in [1]. The same authors also proposed a generalized algorithm for sequential pattern mining GSP [3] to reduce the search space for finding frequent sequences.

Later efficient algorithms such as SPADE [4], PrefixSpan [5] and SPAM [6] were developed to improve the efficiency of sequential pattern mining in terms of time and space complexity. SPADE adopts breadth-first search where as PrefixSpan and SPAM adopt depth-first search. SPADE uses vertical data format and mines the sequential patterns through a simple join on id-lists. PrefixSpan uses horizontal data format and generates the sequential patterns with the pattern growth paradigm. SPAM uses vertical bitmap representation and it runs faster than PrefixSpan and SPADE. But, SPAM consumes more memory space than the other two methods.

Closed itemset mining was proposed to mine closed itemsets without any supersets with the same support. Closed itemset mining can lead to orders of magnitude smaller result set than frequent itemset mining while retaining the completeness i.e., from this concise result set, it is straightforward to generate all the frequent itemsets with accurate support counts. Closed item set mining algorithms like CLOSET[7] and CHARM [8] 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.

Tele:
E-mail addresses: prajusvs@gmail.com

CLOSET+[9] combines the merits of the previously developed effective strategies and new concepts such as item skipping technique and efficient subset-checking scheme. TFP[10] generates top-k frequent closed itemsets. It employs a mixed top-down and bottom-up FP-Tree traversing strategy, a novel closed itemset checking method, a fast two level hash-indexed result tree and a set of pruning techniques to speed up the mining. In most cases, it outperforms CHARM and CLOSET+ even when they are executed with the best tuned minimum support.

In recent years, some research has started to focus on closed sequential pattern mining. There are only two popular algorithms CloSpan [11] and BIDE [12] in closed sequential pattern mining. 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. Further works have also focused on domain specific challenges, such as mining the top-k closed sequential patterns [13] and temporal patterns [14].

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 $S_1 \sqsubseteq S_2$, if there exit integers $1 \leq i_1 < i_2 < \ldots < i_m \leq n$ 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 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.

Table 1. A sample sequence database

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

Proposed Method

In this section, we discuss about our proposed algorithm CSPgrow that extends Apriori property and the depth-first pattern growth procedure to find all closed sequential patterns. CSPgrow uses the C*losure Checking* strategy to eliminate non-closed patterns and *Extension Checking* strategy to prune the search space.

Definition 1 (Closed Pattern): A pattern S is called a closed pattern in a sequence database if there is no super-pattern S' ($S \sqsubseteq S'$) such that support(S) = support(S').

Definition 2 (Pattern Growth ‘Δ’): Let $P = a_1a_2 \ldots a_m$ be a pattern, the growth of P with event e ($a_1a_2 \ldots a_me$) is called pattern growth, denoted by $P \Delta e$.

Theorem 1 (Closure Checking): In a sequence database, the pattern P is not closed if there is a super-pattern of P with the same support.

Proof: The above theorem indicates that, to verify whether a pattern P is closed or not, we only need to check whether there is a super pattern of P denoted as P′ such that support(P) = support(P′). This closure checking strategy can be used to eliminate non-closed patterns from the output. But, we cannot use it to prune the search space.

We use *Extension Checking* for pruning the search space. For a pattern $P = a_1a_2 \ldots a_m$ in sequence database and an extension to P w.r.t. some event e is denoted as P′. We can prevent growing of P in the DFS if there exists P′ such that support (P) = support (P′). Because growing P will not produce any closed patterns. This *Extension Checking* strategy is useful for pruning the search space and improves the efficiency of the algorithm.

Algorithm 1: CSPgrow

```
Input: Sequence database SD and minimum support  m_sup
Output: Closed sequential patterns.
1: S1= frequent 1-sequnces in SD
2: CSP = 0
3: for each  i in S1 do
4:     P = i
5:     CS = Generate_patterns(S1,P)
6:     CSP = CSP ∪ CS
7: end for
```

Algorithm 2: Generate_patterns (S1, P)

```
Input: Sequences S1 and pattern P
Output: Closed sequential patterns with prefix P
1: CS = 0
2: if support (P) ≥ m_sup and ExtensionCheck (P) = prune then
3:     if  ClosureCheck (P) = closed then
4:         CS = CS  ∪  {P}
5:     end if
6:     for each i in S1 do
7:         P = P Δ i
8:         Generate_patterns (S1, P);
9:     end for
10: end if
11: return CS.
```

Algorithm 1, CSPgrow, first finds all frequent-1 sequences (line 1) in the sequence database and for each frequent-1 sequence, Generate_patterns(S1, P) is called (line 5) to find all closed sequential patterns with P as their prefix. Algorithm 2, Generate_patterns(S1, P), is a DFS of the pattern space starting from P. The Apriori property and Extension Checking pruning method are applied to prune the search space(line 2). The pattern closeness is verified in line 3 to eliminate nonclosed sequential patterns. In each iteration of lines 6-9, the pattern P is extended with i (line 7), and Generate_patterns(S1, P) is called recursively(line 8) to generate all closed sequential patterns with P as prefix and stores them into set CS(line 4).

Performance Evaluation

In our experiments we used the FIFA dataset. It is a dataset of 20,450 sequences of click stream data from the website of

FIFA World Cup [15]. It has 2,990 distinct items. The average sequence length is 34.74 items with a standard deviation of 24.8 items. The characteristics of the dataset are given in Table 2.

Table 2. Characteristics of the dataset

S. No.	Characteristic	Value
1	No of sequences	20450
2	No of distinct items	2990
3	Average sequence length	34.74

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 FIFA dataset to find out closed sequential patterns. The Fig. 1 shows the performance comparison between CSPgrow and ClaSP algorithms. Our proposed algorithm CSPgrow runs faster than ClaSP.

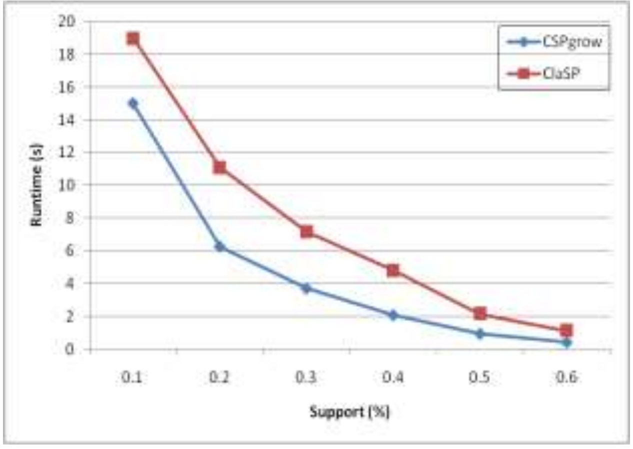

Fig. 1. Performance comparison

Conclusion

In this paper, we propose an efficient algorithm CSPgrow for mining closed sequential patterns in large sequence databases. The closed sequential pattern mining has the same expressive power of sequential pattern mining and also produces more compact result set. Our proposed algorithm CSPgrow outperforms ClaSP by an order of magnitude.

Future research in this area will be focus on improving the efficiency of the algorithms either with new structures, new representations or by managing the database in the main memory. Other interesting research problems that can be pursued include mining of closed structured patterns and closed graph patterns.

References

[1] R. Agrawal and R. Srikant, “Mining sequential patterns,” Proc. Int’l Conf. Data Engineering (ICDE ’95), pp. 3-14, Mar. 1995.

[2] Antonio Gomariz, Manuel Campos, Roque Marin, and Bart Goethals, “ClaSP: An efficient algorithm for mining frequent closed sequences,” PAKDD 2013, LNAI 7818, Part I, pp. 50–61, 2013.

[3] R. Srikant and R. Agrawal, “Mining sequential patterns: Generalizations and performance improvements,” Proc. Int’l Conf. Extending Database Technology (EDBT ’96), pp. 3-17, Mar. 1996.

[4] M. Zaki, “SPADE: An efficient algorithm for mining frequent sequences,” Machine Learning, vol. 42, pp. 31-60, 2001.

[5] 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.

[6] J. Ayres, J. Gehrke, T. Yiu, and J. Flannick, “Sequential pattern mining using a bitmap representation,” Proc. ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (SIGKDD ’02), pp. 429-435, July 2002.

[7] J. Pei, J. Han, and R. Mao, “CLOSET: An efficient algorithm for mining frequent closed itemsets,” Proc. ACM SIGMOD Workshop Research Issues in Data Mining and Knowledge Discovery (DMKD ’00), pp. 21-30, May 2000.

[8] M. Zaki and C. Hsiao, “CHARM: An efficient algorithm for closed itemset mining,” Proc. SIAM Int’l Conf. Data Mining (SDM ’02), pp. 457-473, Apr. 2002.

[9] J. Wang, J. Han, and J. Pei, “CLOSET+: Searching for the best strategies for mining frequent closed itemsets,” Proc. ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (SIGKDD ’03), pp. 236-245, Aug. 2003.

[10] J. Han, J. Wang, Y. Lu, and P. Tzvetkov, “Mining top-K frequent closed patterns without minimum support,” Proc. IEEE Int’l Conf. Data Mining (ICDM ’02), pp. 211-218, Dec. 2002.

[11] X. Yan, J. Han, and R. Afshar, “CloSpan: Mining closed sequential patterns in large databases,” Proc. SIAM Int’l Conf. Data Mining (SDM ’03), pp. 166-177, May 2003.

[12] J. Wang, J. Han, and Chun Li, “Frequent closed sequence mining without candidate maintenance,” IEEE Trans. Knowledge and Data Eng., vol. 19, no. 8, pp. 1042-1056, Aug. 2007.

[13] P. Tzvetkov, X. Yan, and J. Han, “TSP: Mining top-K closed sequential patterns,” Proc. IEEE Int’l Conf. Data Mining (ICDM ’03), pp. 347-354, Dec. 2003.

[14] F. Nakagaito, T. Ozaki, and T. Ohkawa, “Discovery of quantitative sequential patterns from event sequences,” Proc. 9th IEEE Int’l Conf. Data Mining (ICDM 2009), pp. 31–36, 2009.

[15] http://www.philippe-fournier- viger.com/ spmf /index.php ? link=datasets.php

[16] Nancy P. Lin, Wei-Hua Hao, Hung-Jen Chen, Hao-En Chueh and Chung-I Chang, “Fast mining of closed sequential patterns,” WSEAS Transactions on Computers, vol. 7, no. 3, Mar. 2008.

[17] V. Purushothama Raju and G.P. Saradhi Varma, “A framework for mining closed sequential patterns,” International Journal of Computer Science and Information Technologies, vol. 5, no.2, pp. 1864-1866, 2014.

[18]Kuo-Yu Huang, Chia-Hui Chang, Jiun-Hung Tung, and Cheng-Tao Ho, “COBRA: Closed sequential pattern mining using bi-phase reduction approach,” Springer LNCS 4081, pp. 280-291, 2006.

[19] Takei H. and Yamana H., “IC-BIDE: Intensity constraint-based closed sequential pattern mining for coding pattern extraction,” IEEE 27th International Conference on Advanced Information Networking and Applications (AINA 2013), pp. 976-983, Mar. 2013.

[20] Panida Songram and Veera Boonjing, “Closed multidimensional sequential pattern mining,” Int. J. Knowledge Management Studies, vol. 2, no. 4, pp. 460-479, 2008