Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 15 (2011) 3653 – 3657

Procedia Engineering

www.elsevier.com/locate/procedia

Advanced in Control Engineering and Information Science

An effective algorithm for mining sequential generators

Shengwei Yi[a*],Tianheng Zhao[a],Yuanyuan Zhang[ab],Shilong Ma[a],Zhanbin Che[a] a*

[a]*State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China*
[b]*College of Software, Beihang University, Beijing 100191, China*

Abstract

Mining frequent sequences patterns invokes the interests of many searchers. However, the result set of frequent sequences is tremendous. While frequent sequential generator patterns can compact the result set of frequent sequences greatly and are superior to the frequent sequential closed patterns in classification and model selection. The existing sequential generator patterns mining algorithm fails to make full use of the relationship between a sequence and its subsequence. And when mining the frequent sequential generators, the existing algorithms are ineffective and inefficient. In order to resolve this problem, an effective and efficient algorithm is presented. A lot of experiments about the performance on datasets are conducted. The results show that the algorithm proposed is better than the existing algorithms for mining the sequential generators.

Keywords: Sequential Generators; Algorithms; Safe Pruning; Generator Checking

1. Introduction

Sequential patterns exists in many application fields such as geological disasters analysis, biological information, e-commerce applications. All the application sequence data can be analyzed to find the sequential rules implicit in the application data. Sequential patterns have the downclosure property which is that when a sequential pattern is frequent, its subsequence pattern must be frequent [1][2]. When the minimum support threshold is very low, the search space of mining sequential patterns is exponential

* Corresponding author. Tel.: +8-610-8231-7657; fax: +8-610-8231-7657.
E-mail address: yishengwei@foxmail.com.

1877-7058 © 2011 Published by Elsevier Ltd.
doi:10.1016/j.proeng.2011.08.684

explosion. The sequential closed patterns, or the sequential generators, or the maximum sequences can resolve the problem. The maximum sequences are that whether a sequential pattern is frequent or not, while the support information of a sequential pattern is lost. MSPS algorithm can mining maximum sequential patterns effectively [3]. Closed sequential patterns are that there is no supersequence of a sequence being frequent sequential pattern and the support of this sequence is equal to the support of its supersequence. CloSpan [4], BIDE [5], IMCS [6]can mining sequential closed patterns. It is preferable for sequential generator patterns to sequential closed patterns in classification and model selection. The existing algorithms for mining frequent sequence patterns, maximum sequential patterns and sequential closed patterns can not be applied to mine sequential generator patterns from large transaction databases. How to mine sequential generator patterns effectively is an interesting research topic now [7][8]. The algorithm FEAT [7] is based on sequential patterns growth with forward pruning strategy and backward pruning strategy, along with sequential generator checking technique. However, the pruning strategy has enormous time cost for pruning the non-generator sequences that should be pruned since it causes many unuseful the database projection operations and the comparison of the projected databases. Thus, the algorithm FEAT is ineffective to mine frequent sequential generators.

In order to avoid the enormous time cost for pruning, a novel approach FSGP (Frequent Sequential Generators Patterns, FSGP) based on depth first search framework is put forward. Safe pruning strategy is given on the basis of the inclusion relationship between a sequence and its subsequence. A lot of experiments have been conducted to validate the effectiveness of the algorithm FSGP.

The remaining of this paper is organized as follows. Section 2 describes the problem and properties of mining sequential generator patterns. Section 3 formulates the algorithm of FSGP. We report our performance study and analysis on a lot of experiments on datasets in Section 4. Section 5 is the conclusions.

2. Properties of sequential generators

The common properties and theorem of sequential pattern which will be used in the algorithm FSGP design are presented and proved.

Definition 1: the definition of the problem of sequential generator patterns: Given a transaction database SDB, the minimum support $\min\_\sup$, the problem of mining sequential generator patterns is to find all the set of frequent sequential generator patterns from the transaction database SDB.

Lemma 1 (sequential safe substitute): Given tow sequences S_A, S_B and a sequence S, the projected sequence of the sequence S_A on the sequence S is S_{S_A}, the projected sequence of the sequence S_B on the sequence S is S_{S_B}, if $S_{S_A} = S_{S_B} \neq \phi$ holds, then the sequence S_A can be substitute for the sequence S_B with regard to the sequence S, and vice versa.

Lemma 2 (transaction database safe substitute): Given a sequential database SDB, its size is $|SDB|$, the set of transaction is T, $\forall T_i \in SDB, 1 \le i \le |SDB|$, with regard to the two sequences S_A, S_B, the projected sequence of the sequence S_A on each transaction T_i is T_{iS_A}, the projected sequence of the sequence S_B on each transaction T_i is T_{iS_B}, if $T_{iS_A} = T_{iS_B} \neq \phi$ (the null value), then with regard to each transaction, the sequence S_B can substitute for the sequence S_A, and vice versa. And then with regard to the transaction database SDB, the sequence S_B can substitute for the sequence S_A, and vice versa.

Theorem 1 (Safe Pruning): Given a sequence $S_n = e_1 e_2 \cdots e_{n-1} e_n$, $i = n-1$, its one subsequence is $S_{n-1} = e_1 e_2 \cdots e_{n-3} e_{n-2} e_n$ while the ith item is removed from the sequence S_n. If the sequence S_n substitute for the sequence S_{n-1} safely with reference to the transaction database SDB, then the sequence S_n can be pruned safely.

According to the definition of sequential generator and the reference [7], the sequential generator checking theorem is as the following.

Theorem 2 (sequential generator pattern checking): a sequence $S_n = e_1 e_2 \cdots e_{n-1} e_n$ is a sequential generator pattern, if and only if there is no a subsequence that meets the under conditions simultaneously: (1) $1 \le i \le n$, $S_n^{(i)} \subset S_n$, and （2） $Sup|_{SDB}(S_n^{(i)}) = Sup|_{SDB}(S_n)$.

3. Algorithm Description

Based on analysis of the propertities of the sequence generator patterns, the description of the algorithme FSGP is given here.

Table 1. Algorithm FSGP

The Algorithm: sequential generator generated: $SeqGenMiner(SDB|_{S_p}, S_p, \min\_\sup, GS)$
Input: Sequential Database: $SDB|_{S_p}$, the prefix sequence: S_p , the minimum support: $\min\_\sup$
Output: the result set of the sequential generator patterns: GS
Procedure:
1 begin
2 $FS = \phi$;
3 $GS = \phi$;
4 $FS = ValidFreSeqMiner(SDB|_{S_p}, S_p, \min\_\sup, FS)$;
5 $GS = GeneratorCheck(SDB, FS)$;
6 return GS ;
7 End

In algorithm 1, the set of valid frequent sequential patterns and the result set of sequential generators are set to null (line 2 to line 3). the candidate set of sequential generators, that is the set of the valid frequent sequential patterns is obtained by calling the procedure $ValidFreSeqMiner(SDB|_{S_p}, S_p, \min\_\sup, FS)$ (line 4) which can use the Safe Pruning in Theorem 1 to prune most of the sequences that should be pruned . The Safe Pruning in the algorithm FSGP is better than the pruning method in the algorithm including the forward pruning and the backward pruning which consume very much time and space cost since there are n projected databases for subsequences of the sequence S_p. Each valid frequent sequential pattern is checked by the sequential generator checking theorem from the set of the valid frequent sequential patterns, then the non-generators are removed, and the result set of the sequential generators is generated $GS = GeneratorCheck(SDB, FS)$ (line 5) by the Theorem 2 which can check the candidate set of the sequential generators by a mechanism of fast sequential generator checking implementation.

4. Experiments

We performed extensive experiments to evaluate the performance of FSGP we presented. And we use FEAT as the compared algorithm to generate the sequential generators. All experiments were done on a laptop computer with a Intel Core i7-740QM @ 1.73GHz CPU, a 2GB*4 @ 1066MHz memory, running

on Win7Pro x64 operating system. The algorithm FSGP and the algorithm FEAT are implemented in C and compiled using the Embarcadero RAD Studio XE 2010.

Datasets: we use the real dataset ProgramTrace in our performance study. ProgramTrace is a dataset for program trace. It contains 10 sequences, 105 items. The average length of sequences is 488, the maximum length of sequences is 989.

The runing time comparison of the algorithm FSGP and the algorithm FEAT. The relative support is 80/100, 90/100, 100/100 respectively. The unit of running time is second.

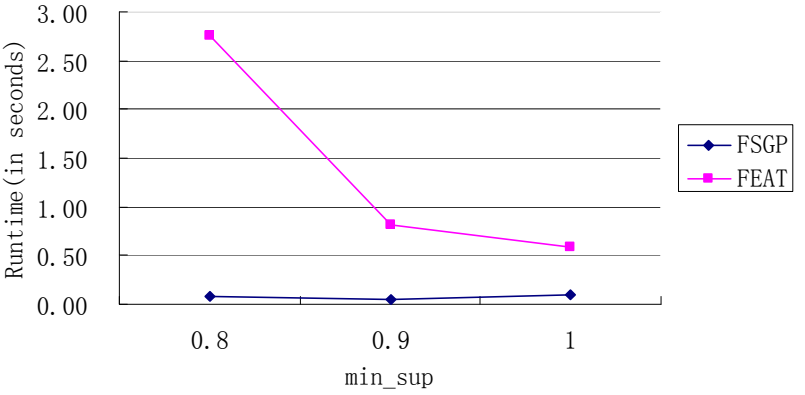

Fig. 1. The running time on the ProgramTrace dataset

Fig 1 shows that the running time on the ProgramTrace dataset by the algorithm FSGP and the algorithm FEAT about the relative support of 80/100, 90/100, 100/100 respectively. The algorithm FSGP is superior to the algorithm FEAT on the running time performance. When the support is 0.7, the algorithm FEAT collapsed.

5. Conclusions

When a dataset is described, the sequential generators are preferable to the sequential closed patterns in terms of the minimum description length principle. The time cost of the existing algorithms for mining sequential generator patterns is great. A novel algorithm named FSGP for mining sequential generator patterns is proposed with the safe pruning strategy consuming a little time cost and the mechanism of sequential generators checking fast. A lot of experiments on the performance study of the algorithm are conducted on the real datasets The experimental results show that the algorithm FSGP has better performance than the existing algorithms for mining sequential generator patterns.

Acknowledgements

The work was supported by National Key Basic Research Program (NKBRP) 973 Program (No. 2005CB321902); and Ministry of Science and Technology Major Support Project (No. 2011BAH14B04),

and National Natural Science Foundation of China (No. 61003016), and the independent project founded by the State Key Laboratory of Software Development Environment in Beihang University (No. SKLSDE-2011ZX-09).

References

[1] R. Agrawal, R. Srikant. Mining Sequential patterns. *Proceedings of the Eleventh International Conference on Data Engineering(ICDE 1995)*. Taipei, Taiwan, IEEE Computer Society Press,1995. pp. 3-14.

[2] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong 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*, 2004, 16(10), pp. 1-17.

[3] Congnan Luo and Soon M. Chung. A scalable algorithm for mining maximal frequent sequences using a sample. *Knowledge and Information Systems*. Volume 15, Number 2, 2007, pp. 149-179.

[4] X. Yan, J. Han, R. Afshar. CloSpan: Mining closed sequential patterns in large datasets. *In: Proc of the 3th SIAM International Conference on Data Mining (SDM 2003)*. San Francisco, CA, USA: SIAM Press, 2003, pp. 166-177.

[5] J. Wang, J. Han, C. Li. Frequent closed sequence mining without candidate maintenance. *IEEE Transaction Knowledge and Data Engineering*, 2007, 19(8), pp. 1042-1056.

[6] Lei Chang, Tengjiao Wang, Dongqing Yang, Hua Luan, Shiwei Tang: Efficient algorithms for incremental maintenance of closed sequential patterns in large databases. *Data Knowl. Eng*. 2009, 68(1), pp. 68-106.

[7] Chuancong Gao, Jianyong Wang, Yukai He, Lizhu Zhou. Efficient Mining of Frequent Sequence Generators. *WWW 2008*, April 21-25, 2008. Beijing, China.

[8] David Lo, Siau-Cheng Khoo, Jinyan Li: Mining and Ranking Generators of Sequential Patterns. *2008 SIAM Conference on Data Mining (SDM 2008)*: April, 2008, Atlanta, Georgia, USA, pp. 553-564.