COCOA: Compressed Continuity Analysis for Temporal Databases

Kuo-Yu Huang, Chia-Hui Chang, and Kuo-Zui Lin

Department of Computer Science and Information Engineering,
National Central University, Chung-Li, Taiwan 320
{want,kuozui}@db.csie.ncu.edu.tw, chia@csie.ncu.edu.tw

Abstract. A continuity is a kind of inter-transaction association which describes the relationships among different transactions. Since it breaks the boundaries of transactions, the number of potential itemsets and the number of rules will increase drastically. In this paper we consider the problem of discovering frequent compressed continuity patterns, which have the same power as mining the complete set of frequent continuity patterns. We devised a three-phase algorithm, COCOA, for frequent compressed continuity mining.

1 Introduction

Patterns mining plays an important role in data mining areas. Most of the previous studies, such as frequent itemsets and sequential patterns are on mining associations among items within the same transaction. This kind of the patterns describe how often items occur together. In spite the above frequent patterns reflect some relationship among the items, it is not a suitable and definite rule for trend prediction. Therefore, we need a pattern that shows the temporal relationships between items in different transactions. In order to distinguish these two kinds of associations, we call the former task as **intra-transaction association mining**, the latter task as **inter-transaction association mining**. There are three kinds of inter-transaction patterns defined recently, including frequent episodes [2], periodic patterns[4] and inter-transaction associations [3].

An episode is defined to be a collection of events in a specific window interval that occur relatively close to each other in a given partial order. Take Fig. 1 as an example, there are six matches of episode $\langle AC, BD \rangle$, from E_1 to E_6 , in temporal database TD . Unlike episodes, a periodic pattern considered not only the order of events but also the exact positions of events. To form periodicity, a list of k disjoint matches is required to form a contiguous subsequence with k satisfying some predefined minimum repetition threshold. For example, pattern $(AC, *, BD)$ is a periodic pattern that matches P_1 , P_2 , and P_3 , three contiguous and disjoint matches. The notion of inter-transaction association mining is firstly used in [3] by Tung et al. In order to distinguish this kind of inter-transaction patterns from episodes and periodic patterns, we call them as “**continuity**” as suggested in [1]. For example, pattern $[AC, *, BD]$ is a continuity with four matches P_1 , P_2 , P_3 , and P_4 in Fig. 1.

Fig. 1. An example of temporal database TD .

Since inter-transaction associations break the boundaries of transactions, the number of potential itemsets and the number of rules will increase drastically. This reduces not only efficiency but also effectiveness since users have to sift through a large number of mined rules to find useful ones. In this paper, we proposed the mining of frequent **compressed** continuities, which has the same power as mining the complete set of frequent continuities. A compressed continuity is composed of only closed itemsets and the don't-care characters. We devised a three-phase algorithm, COCOA (Compressed Continuity Analysis), for mining frequent compressed continuities from temporal databases with a utilization of both horizontal and vertical formats.

Therefore, the problem is formulated as follows: given a minimum support level $minsup$ and a maximum window bound level $maxwin$, our task is to mine all frequent compressed continuities from temporal databases with support greater than $minsup$ and window bound less than $maxwin$.

2 COCOA: Compressed Continuities Analysis

This section describes our frequent compressed continuities mining algorithm phase by phase. The algorithm uses both horizontal and vertical database formats. The COCOA algorithm consists of following three phases:

- Phase I: Mining Frequent Closed Itemsets. Since the COCOA algorithm uses a vertical database format, frequent closed itemsets are mined using a vertical mining algorithm, CHARM[5]. For example, there are four closed frequent itemsets in Fig. 1: $\{AC\}$, $\{C\}$, $\{BD\}$, $\{B\}$.
- Phase II: Pattern Encoding and Database Recovery. For each frequent closed itemset in TD , we assign a unique number to denote them (see Fig. 2(b)). Next, based on the time lists of the encoded patterns, we construct a recovered horizontal database, RD , as shown in Fig. 3.
- Phase III: Mining Frequent Compressed Continuities. Starting from 1-continuity pattern, we use depth-first enumeration to discover all frequent compressed continuity patterns. For each event (frequent closed itemset) in RD , we calculate its projected window list (PWL , see Fig. 2) from the event's timelist and find all frequent events in its PWL through recovered database RD . We then output the frequent continuity formed by current pattern and the frequent events in the PWL . For each extension pattern, the process is applied recursively to find all frequent continuities until the PWL becomes empty or the window of a continuity is greater than the $maxwin$. Finally, the frequent compressed continuities should be decoded from its primal symbol sets.

Id	itemset	Time List	Projected Window List
[1]	{C}	1, 2, 4, 7, 8, 11, 14, 15	2, 3, 5, 8, 9, 12, 15, 16
[2]	{D}	3, 5, 6, 9, 12, 13, 16	4, 6, 7, 10, 13, 14
[3]	{A, C}	1, 4, 7, 8, 11, 14	2, 5, 8, 9, 12, 15
[4]	{B, D}	3, 6, 9, 12, 16	4, 7, 10, 13

Fig. 2. Frequent closed itemsets for Fig. 1

Time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Encoded	[1]	[1]	[2]	[1]	[2]	[2]	[1]	[1]	[2]		[1]	[2]	[2]	[1]	[1]	[2]
Pattern	[3]		[4]	[3]		[4]	[3]	[3]	[4]		[3]	[4]		[3]		[4]

Fig. 3. The recovered horizontal database for Fig. 1

3 Conclusion and Future Work

The performance study shows that the algorithm proposed is both efficient and scalable, and is about an order of magnitude faster than FITI. Readers may consult the authors for the complete paper. For future work, maintaining closed patterns can further reduce the number of redundant patterns and save additional efforts for unnecessary mining. Therefore, more efforts need to be involved in the inter-transaction association mining.

Acknowledgement

This work is sponsored by NSC, Taiwan under grant NSC92-2213-E-008-028.

References

1. K. Y. Huang, C. H. Chang, and Kuo-Zui Lin. Prowl: An efficient frequent continuity mining algorithm on event sequences. In *Proceedings of 6th International Conference on Data Warehousing and Knowledge Discovery (DaWak'04)*, 2004. To Appear.
2. H. Mannila, H. Toivonen, and A. I. Verkamo. Discovering frequent episodes in event sequences. *Data Mining and Knowledge Discovery (DMKD)*, 1(3):259–289, 1997.
3. A. K. H. Tung, H. Lu, J. Han, and L. Feng. Efficient mining of intertransaction association rules. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 15(1):43–56, 2003.
4. J. Yang, W. Wang, and P.S. Yu. Mining asynchronous periodic patterns in time series data. *IEEE Transaction on Knowledge and Data Engineering (TKDE)*, 15(3):613–628, 2003.
5. M.J. Zaki and C.J. Hsiao. Charm: An efficient algorithm for closed itemset mining. In *Proc. of 2nd SIAM International Conference on Data Mining (SIAM 02)*, 2002.