Expert Systems with Applications 38 (2011) 11024–11031

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Using a projection-based approach to mine frequent inter-transaction patterns

Chun-Sheng Wang [a], Kuo-Chung Chu [b,*]

[a] Department of Information Management, Jinwen University of Science and Technology, No. 99, An-Chung Road, Hsin-Tien, Dist. New Taipei City, Taiwan, ROC
[b] Department of Information Management, National Taipei University of Nursing and Health Sciences, No. 365, Min-Te Road, Taipei, Taiwan, ROC

ARTICLE INFO

Keywords:
Data mining
Frequent pattern
Inter-transaction pattern
Projected database

ABSTRACT

In this paper, we propose an algorithm called PITP-Miner that utilizes a projection based approach to mine frequent inter-transaction patterns efficiently. The algorithm only searches for local frequent items in a projected database that stores potential local inter-transaction items and partitions the database into a set of smaller databases recursively. In addition, two pruning strategies are designed to further condense the partitioned databases and thus accelerate the algorithm. Our experiment results demonstrate that the proposed PITP-Miner algorithm outperforms the ITP-Miner and FITI algorithms in most cases.

1. Introduction

Mining association rules from large databases have received considerable attention in recent years and numerous studies have been conducted (Agrawal, Imielinski, & Swami, 1993; Han & Kamber, 2006; Lee, Lin, & Wang, 2006; Lee, Hong, Ko, Tsao, & Lin, 2007). Traditional association rule mining methods are intra-transactional in nature, so they only consider associations that occur within the same transaction. In contrast, an inter-transaction association rule can represent the associations of items within a transaction, as well as the associations of items across different transactions along a certain dimension (Lu, Feng, & Han, 2000). For example, an intra-transaction association rule might state: "When the prices of Intel and IBM go up, 85% of the time, the price of SUN will increase on the same day". However, an inter-transaction can extend such a rule to: "When the prices of Intel and IBM go up, 85% of the time, the price of SUN will increase two days later".

Lu, Han, and Feng (1998) first used inter-transaction association rules to predict stock market movements, and Feng, Dillon, and Liu (2001) applied such rules to meteorological data study. Subsequently, Li, Feng, and Wong (2005) extended inter-transaction association rules to a more general form of association rules, called generalized multidimensional inter-transaction association rules. Several algorithms have been proposed for mining frequent inter-transaction patterns from large databases. For example, Lu et al. (2000) introduced the E/EH-Apriori (Extended/Extended Hash-based Apriori) algorithm, in which E-Apriori uses the Apriori property to discover frequent inter-transaction patterns and EH-Apriori uses a hashing technique to prune the number of candidates of length-2. Feng, Yu, Lu, and Han (2002) described a constraint-based inter-transaction association rules mining approach that uses several optimization techniques to enhance the EH-Apriori algorithm under rule templates. The following year, Tung, Lu, Han, and Feng (2003) proposed the FITI (First-Intra-Then-Inter) algorithm, which is implemented in two phases. First, the Apriori algorithm (Agrawal & Srikant, 1994) is executed to discover frequent intra-transaction patterns, which are then used to generate frequent inter-transaction patterns in the second phase. More recently, Lee and Wang (2007) presented an algorithm called ITP-Miner, which uses vertical data structure dat-lists and an ITP-tree to mine all frequent inter-transaction patterns in a depth-first search manner. It has been shown that ITP-Miner outperforms previous inter-transaction mining algorithms.

The task of mining all frequent inter-transaction patterns in very large databases is quite challenging because the transaction boundaries are broken, and the search space increases exponentially when some of the mining parameters change. In this paper, we propose a projection-based approach called the PITP-Miner algorithm for efficient mining of frequent inter-transaction patterns in a large transaction database. The approach is based on a divide-and-conquer, pattern-growth principle, which means that the algorithm searches along a structure called a PITP-tree in a depth-first search manner. In a PITP-tree, each node represents a frequent inter-transaction pattern and an associated projected database that stores local potential inter-transaction items. In the algorithm, a projected database is recursively projected into a set of smaller projected sub-databases, after which frequent inter-transaction patterns are grown in each sub-database by searching only local frequent fragments. In addition, we introduce two pruning strategies, called *ancestor node pruning* and *hash table pruning*, to further condense the projected sub-databases and thus accelerate the algorithm. A performance study shows that, in most cases, the proposed PITP-Miner algorithm outperforms the ITP-Miner and FITI algorithms.

* Corresponding author.
E-mail addresses: kcchu8992@gmail.com, kcchu@ntunhs.edu.tw (K.-C. Chu).

0957-4174/$ - see front matter © 2011 Published by Elsevier Ltd.
doi:10.1016/j.eswa.2011.02.146

Table 1
A transaction database.

Transaction	Dat	Itemset	Megatransaction (*maxspan* = 1)			
t_1	0	(a, c)	$(a^0, c^0, a^1, c^1, d^1)$			
t_2	1	(a, c, d)		(a^0, c^0, d^0, c^1)		
t_3	2	(c)			$(c^0, a^1, b^1, c^1, d^1)$	
t_4	3	(a, b, c, d)				(a^0, b^0, c^0, d^0)

The contribution of this paper is threefold. First, we propose an effective projection-based algorithm for mining frequent inter-transaction patterns efficiently. Second, we design two pruning strategies to reduce the search space and thus speed up the algorithm. Third, we demonstrate via experiments that the proposed algorithm outperforms the ITP-Miner and FITI algorithm in most cases.

The remainder of this paper is organized as follows. Section 2 defines the frequent inter-transaction pattern mining problem and introduces some notations. In Section 3, we present the PITP-Miner algorithm. Section 4 describes the experiments and their performance results. Finally, we present our conclusions and indicate some future research directions in Section 5.

2. Problem description

In traditional association mining models, each transaction in a database contains a set of items. Although transactions may occur in different contexts, such as time and location, traditional models ignore this contextual information because the patterns are intra-transactional in nature. However, if we are interested in inter-transaction patterns across multiple transactions, the context in which a transaction occurs is important.

We define a model for inter-transaction pattern mining as follows. An itemset $I = (i_1, i_2, \ldots, i_n)$ is a set of distinct items. When there is only one item in an itemset, the parentheses can be omitted; that is, (i) can be written as i. Items in an itemset are listed in alphabetical order, i.e., we write (a, b, d) instead of (d, a, b). A transaction database $D = \{t_1, t_2, \ldots, t_{|D|}\}$, where $|D|$ is the number of transactions in D and t_i $(1 \leqslant i \leqslant |D|)$ is a transaction of the form $(Dat, Itemset)$. *Dat* is the domain attribute of t_i that describes the contextual information, such as the time stamp or space location associated with t_i. As *Dat* is an ordinal, it can be divided into intervals of equal length. For example, time is a domain attribute that can be divided into days, weeks, etc. These intervals can be represented by non-negative integers 0, 1, 2, and so on. To demonstrate our proposed inter-transaction pattern mining algorithm, we use a transaction database containing four transactions, as shown in Table 1.

An inter-transaction context can be defined based on the domain attribute of a transaction database. Let i_1 and i_2 be the domain attributes of transactions t_1 and t_2, respectively. If we take i_1 as the reference point, the span between t_1 and t_2 is defined as $i_2 - i_1$. The itemset $s_2 = (u_1, u_2, \ldots, u_n)$ at domain attribute i_2 with respect to i_1 is called an extended itemset (e-itemset for short) and denoted as $s_2^{i_2-i_1} = (u_1^{i_2-i_1}, u_2^{i_2-i_1}, \ldots, u_n^{i_2-i_1})$. For example, in Table 1, the e-itemset of the second transaction with respect to the first transaction is $(a^{1-0}, c^{1-0}, d^{1-0}) = (a^1, c^1, d^1)$. Suppose we have an e-itemset $(u_1^i, u_2^i, \ldots, u_n^i)$ in which u_j is an item $(1 \leqslant j \leqslant n)$ and i is the span. We define u_j associated with i as an extended item (e-item for short) denoted by u_j^i. For example, the extended itemset (a^1, c^1, d^1) contains three extended items: a^1, c^1, and d^1.

Given a list of k consecutive transactions $z_1 = (i_1, s_1)$, $z_2 = (i_2, s_2), \ldots, z_k = (i_k, s_k)$ in a transaction database, $w = s_1^0 \cup s_2^{i_2-i_1} \cup \cdots \cup s_k^{i_k-i_1}$ is called a megatransaction, where $k \geqslant 1$. In a megatransaction, the span of the last transaction must be less than or equal to *maxspan* (i.e., $i_k - i_1 \leqslant maxspan$ in w), where *maxspan* is a user-specified threshold. If we set *maxspan* = 1, the transaction database in Table 1 contains four megatransactions: $(a^0, c^0, a^1, c^1, d^1)$, (a^0, c^0, d^0, c^1), $(c^0, a^1, b^1, c^1, d^1)$, and (a^0, b^0, c^0, d^0).

Let $\alpha = s_1^{w_1}, s_2^{w_2}, \ldots, s_m^{w_m}$ be a list of e-itemsets, where $w_1 < w_2 < \cdots < w_m$, and $w_m - w_1$ is not greater than *maxspan*. We can normalize α as $\beta = s_1^0 \cup s_2^{w_2-w_1} \cup \cdots \cup s_m^{w_m-w_1}$ and call β a *pattern*. The number of e-items in a pattern is called the *length* of the pattern, and a pattern of length k is called a k-pattern. Assume there are two patterns, $\alpha = s_1^{w_1} \cup s_2^{w_2} \cup \cdots \cup s_n^{w_n}$ and $\beta = s_1'^{v_1} \cup s_2'^{v_2} \cup \cdots \cup s_m'^{v_m}$, where $w_1 = v_1 = 0$ and $n \leqslant m$. Then, we say α is a subpattern of β (or β is a supperpattern of α) if we find $s_1^{w_1} \subseteq s_{j_1}'^{v_{j_1}}, s_2^{w_2} \subseteq s_{j_2}'^{v_{j_2}}, \ldots, s_n^{w_n} \subset s_{j_n}'^{v_{j_n}}$, such that $1 = j_1 < j_2 < \cdots < j_n \leqslant m$ and $w_1 = v_{j_1}, w_2 = v_{j_2}, \ldots, w_n = v_{j_n}$. We can also say that $\alpha \subset \beta$ or β *contains* α. For example, both (d^0, a^3, c^3) and (a^0, d^0, a^1, d^1) are subpatterns of $(a^0, d^0, a^1, b^1, d^1, a^3, c^3)$.

In a transaction database D, let α be a pattern, and T_α be a set of megatransactions in D, where each megatransaction in T_α contains α. The support of α, $sup(\alpha)$, is defined as $|T_\alpha|$. If $sup(\alpha)$ is not less than the user-specified minimum support threshold *minsup*, α is called a frequent pattern. An *inter-transaction association rule* is written in the form of $\alpha \to \beta$, where both α and $\alpha \cup \beta$ are frequent patterns; $\alpha \cap \beta = \varnothing$, $conf(\alpha \to \beta)$ is not less than the user-specified minimum confidence; and the confidence of the rule $conf(\alpha \to \beta)$ is defined as $sup(\alpha \cup \beta)/sup(\alpha)$. The problem of inter-transaction pattern mining involves finding all frequent inter-transaction patterns in a transaction database with respect to the user-specified *minsup* and *maxspan* thresholds.

Let us consider the example shown in Table 1. Assume that *maxspan* = 1, *minsup* = 2 and $X = (c^0, a^1)$. Since the megatransaction formed by the first and second transactions, $(a^0, c^0, a^1, c^1, d^1)$, contains X, $sup(X)$ is set to 1. In addition, the megatransaction formed by the third and fourth transactions, $(c^0, a^1, b^1, c^1, d^1)$, contains X. Thus, $sup(X)$ is incremented by 1, that is, $sup(X) = 2$. Consequently, X is a frequent pattern. Let another pattern $Y = (c^0, a^1, c^1)$, then, we can find $sup(Y) = 2$, so $conf((c^0, a^1) \to (c^0)) = 2/2 = 100\%$.

Table 2 shows the complete set of frequent inter-transaction patterns mined from the transaction database in Table 1 with *maxspan* = 1 and *minsup* = 2. The number after each pattern represents the pattern's support. The total number of frequent inter-transaction patterns is 16.

3. The proposed method

3.1. A projection-based approach

In this section, we introduce a new projection-based approach for mining frequent inter-transaction patterns in a transaction database. First, we define the concept of projected databases.

Table 2
The patterns mined from Table 1 with *maxspan* = 1 and *minsup* = 2.

Length	Frequent inter-transaction patterns
1	(a^0):3, (c^0):4, (d^0):2
2	(a^0, c^0):2, (a^0, d^0):2, (a^0, c^1):2, (c^0, d^0):2, (c^0, a^1):2, (c^0, c^1):3, (c^0, d^1):2
3	(a^0, c^0, d^0):2, (a^0, c^0, c^1):2, (c^0, a^1, c^1):2, (c^0, a^1, d^1):2, (c^0, c^1, d^1):2
4	(c^0, a^1, c^1, d^1):2