Available online at www.sciencedirect.com

ScienceDirect

Information Sciences 177 (2007) 3453–3476

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

www.elsevier.com/locate/ins

An efficient algorithm for mining frequent inter-transaction patterns

Anthony J.T. Lee *, Chun-Sheng Wang

Department of Information Management, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei 10617, Taiwan, ROC

Received 3 May 2006; received in revised form 1 March 2007; accepted 10 March 2007

Abstract

In this paper, we propose an efficient method for mining all frequent inter-transaction patterns. The method consists of two phases. First, we devise two data structures: a dat-list, which stores the item information used to find frequent inter-transaction patterns; and an ITP-tree, which stores the discovered frequent inter-transaction patterns. In the second phase, we apply an algorithm, called ITP-Miner (Inter-Transaction Patterns Miner), to mine all frequent inter-transaction patterns. By using the ITP-tree, the algorithm requires only one database scan and can localize joining, pruning, and support counting to a small number of dat-lists. The experiment results show that the ITP-Miner algorithm outperforms the FITI (First Intra Then Inter) algorithm by one order of magnitude.

Keywords: Association rules; Data mining; Inter-transaction patterns

1. Introduction

Association rule mining is an important problem in the data mining field [1,2,4,5,8,10–13,15,16,20,22,25,27]. Traditional association analysis is intra-transactional because it focuses on association relationships among itemsets within a transaction. For example, an intra-transaction association rule might be: "If the stock prices of Microsoft and IBM go up, the price of Intel is likely to go up on the same day." However, intra-transactional approaches cannot capture a rule like: "If the stock prices of Microsoft and IBM go up, the price of Intel is likely to go up two days later." Inter-transaction association rule mining [6,7,14,17,18,23,24] extends the association rules to describe association relationships among itemsets across several transactions.

Many algorithms for mining inter-transaction association rules have been proposed. Lu et al. [17] and Feng et al. [6] applied inter-transaction association rule mining algorithms to the prediction of trends in meteorological and stock market data. Lu et al. [17,18] proposed the EH-Apriori algorithm, which uses the Apriori

* Corresponding author.
E-mail addresses: jtlee@ntu.edu.tw, d91725001@ntu.edu.tw (A.J.T. Lee).

0020-0255/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2007.03.007

algorithm to discover frequent inter-transaction itemsets. To enhance their algorithm's efficiency, the authors used a hashing technique to reduce the number of candidate itemsets of length two. Feng et al. [7] used templates to reduce the number of rules. More recently, Tung et al. [23,24] developed an algorithm, called FITI (First Intra Then Inter), which discovers frequent intra-transaction itemsets and uses them to generate frequent inter-transaction itemsets.

All the algorithms for mining inter-transaction association rules developed thus far have been based on Apriori-like breadth-first search (BFS) approaches that search for frequent itemsets level by level. At each level, a database must be scanned once to determine the support for each candidate itemset. It has been shown that Apriori-like approaches [19,22] perform well in finding frequent intra-transaction itemsets when the itemsets are short. However, when mining long frequent itemsets, or using very small support thresholds, the performance of such algorithms often deteriorates dramatically. The reason is that a frequent itemset of length k implies the presence of $2^k - 2$ additional frequent sub itemsets, each of which must be examined. Moreover, since Apriori-like approaches may generate a large number of candidate patterns at each level, they are prone to memory shortage during the mining process. We observe that Apriori-like methods for finding frequent inter-transaction itemsets have the same drawbacks as those for finding frequent intra-transaction itemsets.

Therefore, in this paper, we propose an efficient method for mining a complete set of frequent inter-transaction itemsets (patterns). The method consists of two phases. First, we find all frequent items. For each frequent item found, we construct a dat-list that records the item information required for finding the frequent inter-transaction patterns. Then, we devise a data structure, called an ITP-tree, to store the discovered frequent inter-transaction patterns. In the second phase, we propose an algorithm, called ITP-Miner, to efficiently find all frequent inter-transaction patterns in a depth-first search manner. By using the ITP-tree and dat-lists to mine the frequent inter-transaction patterns, the ITP-Miner algorithm requires only one database scan and can localize joining, pruning, and support counting to a small number of dat-lists. Therefore, it is more efficient than the FITI algorithm.

The remainder of this paper is organized as follows. In Section 2, we describe the problem of mining frequent inter-transaction patterns. Section 3 introduces our proposed algorithm, ITP-Miner, for mining frequent inter-transaction patterns. We explain the basic concept of the algorithm and give an example to illustrate how it works. Section 4 describes the performance evaluation. Finally, we present the conclusions in Section 5.

2. Problem description

In this section, we introduce the notations and describe the problem of mining frequent inter-transaction patterns.

Definition 1. Let I be a set of data items, and N be a set of non-negative integers called domain attributes. A transaction database consists of a set of transactions, where a transaction is in the form of $\langle t, s\rangle, t \in N$ and $s \subseteq I$; t is called a <u>d</u>imensional <u>at</u>tribute (or dat), and s is called an itemset.

The dimensional attribute describes the properties associated with the data items, such as time and location. As it is an ordinal, it can be divided into intervals of equal length. For example, time can be divided into days, weeks, etc. These intervals can be represented by non-negative integers 0, 1, 2, and so on. An itemset is denoted by $\{u_1, u_2, \ldots, u_k\}$, where u_i is an item, $1 \leqslant i \leqslant k$; and items in an itemset are listed in alphabetical order, i.e., we write $\{a, c, d\}$ instead of $\{c, a, d\}$. Table 1 shows a transaction database containing six transactions.

Definition 2. An itemset $s = \{u_1, u_2, \ldots, u_k\}$ at the dimensional attribute t is called an *extended itemset* and denoted by $\Delta_t s = \{u_1(t), u_2(t), \ldots, u_k(t)\}$.

Before mining inter-transaction association rules, we need to find the frequent inter-transaction itemsets that span several transactions. Since an inter-transaction itemset can span many intervals, discovering all such itemsets would require a lot of resources, but a user may only be interested in rules that span a certain number of intervals. Therefore, to avoid wasting resources by mining unwanted rules, we introduce a parameter called *maxspan*. When mining for inter-transaction association rules, we only mine rules whose span is equal to or less than the *maxspan* intervals.