Expert Systems with Applications 38 (2011) 15143–15150

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Temporal data mining with up-to-date pattern trees ☆

Chun-Wei Lin [a], Tzung-Pei Hong [a,b,*]

[a] Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan, ROC
[b] Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung 804, Taiwan, ROC

ARTICLE INFO

Keywords:
Data mining
Temporal data mining
Up-to-date pattern
UDP-tree
UDP-growth

ABSTRACT

Mining interesting and useful frequent patterns from large databases attracts much attention in recent years. Among the mining approaches, finding temporal patterns and regularities is very important due to its practicality. In the past, Hong et al. proposed the up-to-date patterns, which were frequent within their up-to-date lifetime. Formally, an up-to-date pattern is a pair with the itemset and its valid corresponding lifetime in which the user-defined minimum support threshold must be satisfied. They also proposed an Apriori-like approach to find the up-to-date patterns. This paper thus proposes the up-to-date pattern tree (UDP tree) to keep the up-to-date 1-patterns in a tree structure for reducing database scan. It is similar to the FP-tree structure but more complex due to the requirement of up-to-date patterns. The UDP-growth mining approach is also designed to find the up-to-date patterns from the UDP tree. The experimental results show that the proposed approach has a better performance than the level-wise mining algorithm.

1. Introduction

Knowledge discovery in databases (KDD) is to identify efficient and helpful information from large databases and provide automated analysis and solutions. It has attracted a significant amount of research. The approaches may be classified as working on transaction databases, temporal databases, relational databases, multimedia databases, and so on. In particular, finding association rules from transaction databases is most commonly seen in data mining (Agrawal, Imielinksi, & Swami, 1993a, 1993b; Agrawal & Srikant, 1994; Agrawal, Srikant, & Vu, 1997; Chen, Han, & Yu, 1996; Cheung, Han, Ng, & Wong, 1996; Mannila, Toivonen, & Verkamo, 1994; Srikant & Agrawal, 1995).

In the past, many algorithms for mining association rules from transactions were proposed, most of which were based on the Apriori algorithm (Agrawal et al., 1993a). It was proposed to discover the correlation relationships among items or itemsets in transactional databases and has been applied in many areas. When the percentage of transactions containing a candidate itemset is greater than or equal to a pre-defined minimum support threshold, the itemset is considered as a frequent itemset. That is, some correlation relationships exist among the items it includes. The number of derived frequent itemsets is greatly influenced by the pre-defined minimum support threshold.

Han et al. thus proposed the Frequent-Pattern-tree (FP-tree) structure for efficiently deriving the frequent itemsets without candidate generation (Han, Pei, & Yin, 2000). Only the frequent items were kept and composed the tree structure, which was thus condensed. A recursive mining procedure called FP-growth was executed to derive frequent patterns from the FP tree constructed (Han et al., 2000). Han et al. showed the approach had a better performance than Apriori-like approaches.

Recently, temporal data mining has been considered as an important topic attracting many researchers. Analyzing temporal data and discovering temporal patterns are the main concerns in temporal data mining. It typically shows time-related correlations of itemsets from transactions. For instance, the sales of ice cream in summer and of mittens in winter should be higher than those in the other seasons. Some approaches for finding seasonal behaviors of specific items or itemsets where thus proposed (Roddick & Spiliopoulou, 2002). In addition to finding seasonal behaviors, there are many other kinds of knowledge in temporal data mining (Ale & Rossi, 2000; Chen, Petrounias, & Heathfield, 1998; Li & Deogun, 2005; Li, Ning, Wang, & Jajodia, 2003; Lee, Lee, Kim, Hwang, & Ryu, 2002; Ozden, Ramaswamy, & Silberschatz, 1998; Verma et al., 2005).

In the past, Hong et al. proposed a concept of up-to-date patterns, which were frequent within their up-to-date lifetime (Hong, Wu, & Wang, 2009). Formally, an up-to-date pattern is a pair with the itemset and its valid corresponding lifetime, presented as ($\{itemset\}, \langle lifetime\rangle$). The end value of the lifetime is the current time and no other lifetime for the itemset may last longer than

☆ This is a modified and expanded version of the paper "Mining Up-to-Date Knowledge Based on Tree Structures", presented in The International Conference of Soft Computing and Pattern Recognition, Malaysia, 2009.

* Corresponding author at: Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan, ROC.

E-mail addresses: jerrylin@ieee.org (C.-W. Lin), tphong@nuk.edu.tw (T.-P. Hong).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.05.090

it. Note that an itemset not frequent for the entire database may be a frequent up-to-date pattern since its items seldom occurring early may constantly occur lately. Hong et al.. also proposed an algorithm to derive up-to-date patterns from transactions in a level-wise process (Hong et al., 2009).

In this paper, we attempt to derive the up-to-date patterns without the Apriori-like generation of candidates. An up-to-date pattern tree (UDP tree) is first designed to keep the derived frequent up-to-date 1-patterns. It is similar to the FP-tree structure except that the corresponding transaction identifications (TIDs) are also kept. The up-to-date 1-patterns with their frequency and their valid lifetime are retained in the Header_Table as well. An UDP-growth mining approach is then proposed to derive the up-to-date patterns from the UDP tree. Experimental results also show that the proposed approach for mining up-to-date patterns has a better performance than the Apriori-like up-to-date algorithm (Hong et al., 2009) in the execution time and the number of generated candidates.

The remainder of this paper is organized as follows. Related works are reviewed in Section 2. The proposed UDP-tree construction algorithm and an example are described in Section 3. The UDP-growth mining algorithm and an example are stated in Section 4. Experimental results for showing the performance of the proposed algorithms are provided in Section 5. Conclusions are finally given in Section 6.

2. Review of related works

In this section, some related researches are briefly reviewed. They are the frequent pattern tree (FP tree) algorithm and the up-to-date patterns.

2.1. The frequent pattern tree

Data mining involves applying specific algorithms to extract patterns or rules from data sets in a particular representation. One common type of data mining is to derive association rules from transaction data, such that the presence of certain items in a transaction will imply the presence of some other items.

Han et al. proposed the Frequent-Pattern-tree structure (FP-tree) for efficiently mining association rules without generation of candidate itemsets (Han et al., 2000). The FP-tree mining algorithm consists of two phases. The first phase constructs the FP-tree from a database, and the second phase derives frequent patterns from the FP-tree constructed. The FP-tree is used to compress a database into a tree structure with only frequent items. After the FP-tree is constructed from a database, a mining procedure called FP-growth is executed to find all the frequent itemsets (Han et al., 2000). FP-growth does not need to generate candidate itemsets as the Apriori-like approach, but derives candidate frequent patterns directly from the FP-tree. It is a recursive process, handling the frequent items one by one and bottom-up. A conditional FP-tree is generated from each frequent item, and the frequent itemsets with the current processed item can be recursively derived.

2.2. Up-to-data patterns

Hong et al. proposed the concept of up-to-date patterns which concerned the most recent items with an unfixed length of window size (Hong et al., 2009). The concept of up-to-date patterns is shown in Fig. 1, where an up-to-date itemset is a frequent itemset with a valid lifetime, in which the end point is the current time. Its start time will be found to make the lifetime as long as possible.

The valid lifetime for an up-to-date pattern I must satisfy the following formula:

$$n - First(I) + 1 \leqslant \frac{S(I)}{s},$$

where n is the number of transactions in the database, $First(I)$ is the first transaction identification (TID) of I in the valid lifetime, $S(I)$ is the count of I in the valid lifetime, and s is a user-defined minimum support threshold. An example is given below to show the above concept. Assume there is a log database as shown in Table 1. It contains 10 transactions and 6 items, denoted a to f.

Assume the user-defined minimum support threshold is set at 50%. Let us take item c as an example. Its count is 3 and the minimum count is $0.5 * 10(=5)$. The item c is thus not a frequent item in the whole database. However, it is an up-to-date pattern in the valid lifetime of $\langle 5, 10 \rangle$. Note that the first transaction ID with c in the valid lifetime is 7, but the lifetime begins at 5. There are totally six transactions in the interval and three of them include item c.

Hong et al. then proposed an algorithm to find all the up-to-date patterns from a given log database. It first translated the log database into an item-oriented bit-map representation to speed up the execution in the later mining process and then extracted frequent itemsets with the longest valid lifetime from the past to the current time. Hong et al.'s approach could mine more useful frequent itemsets than the conventional ones. Especially when the minimum support threshold was high, the proposed approach had much more frequent itemsets than the traditional ones, which derived very rare rules. The up-to-date patterns could be derived with a larger minimum support value, which thus provided more important information to decision makers.

3. The proposed tree construction algorithm

The proposed constructing algorithm for building an up-to-date pattern tree (UDP-tree) from a log database is described in this section. The notation used in the proposed algorithm is first described below.

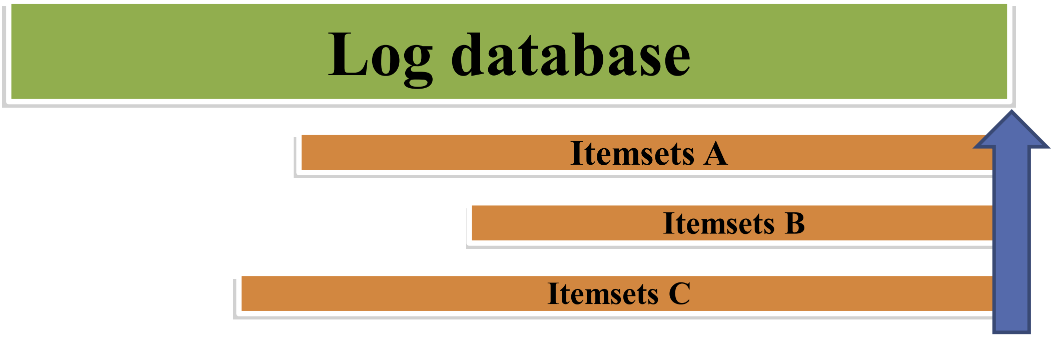

Fig. 1. The concept of up-to-date knowledge.