Information Sciences 278 (2014) 88–103

Contents lists available at ScienceDirect

Information Sciences

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

Maintenance of prelarge trees for data mining with modified records [☆]

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

[a] *Innovative Information Industry Research Center (IIIRC), School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, HIT Campus Shenzhen University Town, Xili, Shenzhen 518055, PR China*
[b] *Shenzhen Key Laboratory of Internet Information Collaboration, School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, HIT Campus Shenzhen University Town, Xili, Shenzhen 518055, PR China*
[c] *Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan, ROC*
[d] *Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung 804, Taiwan, ROC*

ARTICLE INFO

Article history:
Received 8 March 2009
Received in revised form 16 January 2011
Accepted 8 March 2014
Available online 18 March 2014

Keywords:
Data mining
FP-tree
Prelarge tree
Pre-large itemset
Record modification
Maintenance

ABSTRACT

The frequent pattern tree (FP-tree) is an efficient data structure for association-rule mining without generation of candidate itemsets. It is used to compress a database into a tree structure which stores only large items. When data are modified, it, however, needs to process all transactions in a batch way. In the past, the prelarge-tree structure was proposed to incrementally mine association rules efficiently. In this paper, we propose an algorithm to maintain this structure when records in an original database are modified. The proposed maintenance algorithm is based on the pre-large concepts, which are defined by a lower support threshold and an upper support threshold. Due to the pruning properties of pre-large concepts, the proposed approach can reduce the rescan number of an original database when records are modified. It can thus obtain good execution performance for pre-large tree maintenance, especially when each time a small number of records are modified. Although experimental results show that the proposed prelarge-tree maintenance algorithm has good performance for handling modified records, the proposed algorithm needs to maintain nodes of pre-large items in the tree structure. This is the additional overhead, which is a trade-off between execution time and tree complexity.

1. Introduction

Years of effort in data mining have produced a variety of efficient techniques. Depending on the types of databases processed, these mining approaches may be classified as working on transaction databases, temporal databases, relational databases and multimedia databases, among others. On the other hand, depending on the classes of knowledge derived, the mining approaches may be classified as finding association rules [1–3], classification rules [17,23], clustering rules [18,20] and sequential patterns [4,12], and others. Among them, finding association rules in transaction databases is most commonly seen in data mining [5–7,21,22,24,25,27].

[☆] This is a modified and expanded version of the paper "Mining with Prelarge Trees for Record Modification," presented at The Third International Conference on Innovative Computing, Information and Control, Dalian, China, 2008.

[*] Corresponding author at: Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan, ROC. Tel.: +886 75919191.

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

http://dx.doi.org/10.1016/j.ins.2014.03.023
0020-0255/© 2014 Elsevier Inc. All rights reserved.

In the past, many algorithms for mining association rules from transactions were proposed, most of which were based on the Apriori algorithm [3], which generated and tested candidate itemsets level-by-level. This, however, might cause iterative database scans and high computational costs. Han et al. thus proposed the Frequent-Pattern-tree (FP-tree) structure for efficiently mining association rules without generation of candidate itemsets [13]. They showed the approach could have better performance than the Apriori approach.

In real-world applications, a database is dynamic because that its contents may be continually updated over time and its transactions may be added into or deleted and modified from the database. Newly updated data are usually considered more interesting and weighted more important than those kept previously. Some knowledge mined from original databases may thus be no longer valid along with elapsed time. Effectively mining from dynamic databases can thus help users make timely relevant decision.

Cheung et al. thus designed a FUP concept to efficiently handle record insertion, which was referred to incremental mining approaches [8]. Some research focused on managing record deletion, which was called decremental mining [28,29]. Zhang et al. also proposed a post-maintenance process which adopted a weighting technique to highlight new data [27]. The above maintenance approaches could reduce the execution time than simply processing updated databases in a batch way.

Hong et al. proposed the pre-large concepts to incrementally mine association rules, which could reduce the number of rescanning databases [14]. It used a lower support threshold and an upper threshold to reduce the need for rescanning original databases and to save maintenance cost. Lin et al. then combined the concepts of pre-large itemsets and FP trees to design the prelarge-tree structure for effectively handling data insertion in mining [19]. In this paper, we further extend the mining problem for modified records. An efficient and effective prelarge-tree maintenance algorithm is designed for quickly updating the discovered knowledge. The proposed algorithm first calculates the count difference of each item in modified records based on pre-large concepts [16]. It then partitions items into nine cases according to whether they are large, pre-large or small in the original database and whether their item differences are positive, zero or negative. Each case is then processed in its own way. Experimental results also show that the proposed prelarge-tree maintenance algorithm can achieve good performance when records are modified.

The remainder of this paper is organized as follows. Some related works are reviewed in Section 2. The proposed prelarge-tree algorithm for handling modified records is described in Section 3. An example to illustrate the proposed algorithm is given in Section 4. Experimental results for showing the performance of the proposed algorithm are provided in Section 5, and conclusions are finally given in Section 6.

2. Review of related works

In this section, some related researches are briefly reviewed. They are mining association rules, frequent pattern tree, and prelarge tree.

2.1. Mining association rules

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 presence of certain items in a transaction will imply the presence of some other items. To achieve this purpose, Agrawal and his co-workers proposed several mining algorithms based on the concept of large itemsets to find association rules from transaction data [1–3]. They divided the mining process into two main phases. In the first phase, candidate itemsets were generated and counted by scanning the transaction data. If the count of an itemset appearing in the transactions was larger than the predefined threshold value (called the minimum support), the itemset was considered a large itemset. Itemsets containing only one item were processed first. Large itemsets containing only single items were then combined to form candidate itemsets containing two items. This process was repeated until all large itemsets had been found. In the second phase, association rules were induced from the large itemsets found in the first phase. All possible association combinations for each large itemset were formed, and those with calculated confidence values larger than a predefined threshold (called the minimum confidence) were output as association rules.

2.2. Frequent pattern tree

In the Apriori algorithm, it might cause iterative database scans and high computational costs. Han et al. thus proposed the Frequent-Pattern-tree (FP-tree) algorithm for efficiently mining association rules without generation of candidate itemsets [13]. It consisted of two phases. The first phase focused on constructing the FP-tree from the database, and the second phase focused on deriving frequent patterns from the FP-tree. In the first phase, the FP-tree [13] was used to compress a database into a tree structure storing only large items. It was condensed and complete for finding all the frequent patterns. Three steps were involved in FP-tree construction. The database was first scanned to find all items with their frequency. The items with their supports larger than a predefined minimum support were selected as large 1-itemsets (items). Next, the large items were sorted in descending frequency. At last, the database was scanned again to construct the FP-tree according