iTM: An Efficient Algorithm for Frequent Pattern Mining in the Incremental Database without Rescanning

Bi-Ru Dai and Pai-Yu Lin

Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, ROC
brdai@csie.ntust.edu.tw, m9615082@mail.ntust.edu.tw

Abstract. Frequent pattern mining plays an important role in the data mining community since it is usually a fundamental step in various mining tasks. However, maintenance of frequent patterns is very expensive in the incremental database. In addition, the status of a pattern changes with time. In other words, a frequent pattern is possible to become infrequent, and vice versa. In order to exactly find all frequent patterns, most algorithms have to scan the original database completely whenever an update occurs. In this paper, we propose a new algorithm iTM, stands for incremental Transaction Mapping algorithm for incremental frequent pattern mining without rescanning the whole database. It transfers the transaction dataset to the vertical representation such that the incremental dataset can be integrated to the original database easily. As demonstrated in our experiments, the proposed method is very efficient and suitable for mining frequent patterns in the incremental database.

Keywords: Frequent Patterns, Incremental Database, Incremental Mining.

1 Introduction

The frequent pattern plays a key role in many data mining tasks, such as association rules [1,4,10], sequential patterns [11], classification [2,3], and clustering [4]. The frequency of a pattern is computed by counting its occurrence in the transaction database. If the frequency exceeds a user defined threshold, the pattern is considered to be important in the transaction database. However, the cost of finding all frequent patterns is expensive, because patterns are generated from combinations of items, and the number of combinations is enormous. Therefore, mining frequent patterns efficiently becomes a difficult challenge and attracts lots of research attention.

While many efficient algorithms are proposed in recent years, the amount of data also has a huge growth. The size of database evolves from megabytes to terabytes in recent two decades. Furthermore, hundreds of megabytes of new datasets are accumulated to the database every day. Even though many efficient algorithms have been proposed, mining frequent patterns iteratively without past frequent patterns is still very expensive. Therefore, developers change their focus to incremental frequent patterns mining [6,7,8,9,11]. Cheung et al. develop an algorithm, named FUP [6] which is the first incremental mining algorithm. It reuses information from old frequent patterns to save the computation cost. Mining frequent patterns in the
incremental database is more complicated than the mining of the static transaction database. Some frequent patterns are possible to become useless after more data records are collected. Infrequent patterns encounter the similar situation. These phenomena bring about new challenges, such as counting the frequency of a pattern in the updated database.

In order to find frequent patterns from the incremental transaction database, most incremental approaches have to rescan the original database for counting frequencies of new frequent patterns, and then it becomes the bottleneck in this type of approaches. Hence, earlier incremental approaches focus on reducing the number of rescanning the original database when an update occurs. Some developers also try to solve the trade-off in accuracy and efficiency [25]. The approach allows incorrect result in frequent patterns, but guarantees the error bound. It effective increases the efficiency for mining frequent patterns in some situation, if user does not request the accuracy.

One of the algorithms had been proposed for incremental mining without rescanning the original database is the CanTree algorithm [12]. CanTree can be easily maintained without adjusting, merging, and splitting tree nodes when the transaction database is updated. However, it incurs another issue that balances performance between maintenance of supports of patterns and efficiently mining frequent patterns. Unfortunately, they are exclusive in most situations or methods.

In this paper, we propose a refined algorithm of incremental frequent pattern mining base on TM algorithm [5]. It is called incremental Transaction Mapping, abbreviated as iTM. This method maintains frequent patterns in the incremental database without rescanning the original database completely. We also focus on achieving a better compression ratio on the representation of the whole database. In addition, iTM has other benefit that scanning the database only once for constructing its compressed data structure while an update occurs and the data structure can be reused in interactive mining. This is an important feature of our method: building once and mining multiple times. Experiment presents that iTM has the ability of mining frequent patterns in the incremental transaction database efficiently. In addition, it demonstrates that the representation of the transaction database can effectively assist in frequent patterns mining.

The rest of this paper is organized as follows. Section 2 describes some related works about incremental mining. The problem definition will be given in section 3. Then we discuss our algorithm iTM in section 4. Section 5 presents performance evaluation of our method. Conclusion will be given in Section 6.

2 Related Work

In the past, many developers had reported the issue of incremental data mining. The focus of this issue is on maintaining frequent patterns when the database is updated. The first algorithm is FUP which is proposed by Cheung, D.W. et al. [6]. The idea of FUP is to store the supports of all frequent patterns in the previous mining process. Then, this information is used to check status of frequent patterns when an update occurs. The FUP iteratively discovers new frequent patterns in Apriori-like manner from the updated database. Even though, FUP stores the supports of frequent patterns,