Expert Systems With Applications 57 (2016) 214–231

Contents lists available at ScienceDirect

Expert Systems With Applications

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

High utility pattern mining over data streams with sliding window technique

Heungmo Ryang, Unil Yun*

Department of Computer Engineering, Sejong University, Seoul, Republic of Korea

ARTICLE INFO

Article history:
Received 3 March 2015
Revised 29 February 2016
Accepted 1 March 2016
Available online 28 March 2016

Keywords:
Data mining
High utility pattern
Sliding window
Stream mining
Utility mining

ABSTRACT

Processing changeable data streams in real time is one of the most important issues in the data mining field due to its broad applications such as retail market analysis, wireless sensor networks, and stock market prediction. In addition, it is an interesting and challenging problem to deal with the stream data since not only the data have unbounded, continuous, and high speed characteristics but also their environments have limited resources. High utility pattern mining, meanwhile, is one of the essential research topics in pattern mining to overcome major drawbacks of the traditional framework for frequent pattern mining that takes only binary databases and identical item importance into consideration. This approach conducts mining processes by reflecting characteristics of real world databases, non-binary quantities and relative importance of items. Although relevant algorithms were proposed for finding high utility patterns in stream environments, they suffer from a level-wise candidate generation-and-test and a large number of candidates by their overestimation techniques. As a result, they consume a huge amount of execution time, which is a significant performance issue since a rapid process is necessary in stream data analysis. In this paper, we propose an algorithm for mining high utility patterns from resource-limited environments through efficient processing of data streams in order to solve the problems of the overestimation-based methods. To improve mining performance with fewer candidates and search space than the previous ones, we develop two techniques for reducing overestimated utilities. Moreover, we suggest a tree-based data structure to maintain information of stream data and high utility patterns. The proposed tree is restructured by our updating method with decreased overestimation utilities to keep up-to-date stream information whenever the current window slides. Our approach also has an important effect on expert and intelligent systems in that it can provide users with more meaningful information than traditional analysis methods by reflecting the characteristics of real world non-binary databases in stream environments and emphasizing on recent data. Comprehensive experimental results show that our algorithm outperforms the existing sliding window-based one in terms of runtime efficiency and scalability.

1. Introduction

Frequent pattern mining (Lin, Liao, Chang, & Lin, 2014; Pyun, Yun, & Ryu, 2014; Tseng, 2013; Tseng, 2012) is a fundamental research topic in pattern mining. It generates all frequent patterns with no smaller supports (or frequency) than a given minimum support threshold. In this framework, the anti-monotone property (or downward closure property) (Agrawal & Srikant, 1994) is used to prune search space effectively. The property means that any super pattern of an infrequent one cannot be frequent. In addition, it is an essential criterion for efficient pattern mining. Meanwhile, this approach has been applied to difference kinds of databases such as sequential databases (Chang, 2011; Fournier-Viger, Faghihi, Nkambou, & Nguifo, 2012; Guil & Marín, 2012; Huang, 2013; Pham, Luo, Hong, & Vo, 2014) and incremental databases (Yun & Lee, 2016a), and there are also various types of methods according to characteristics of patterns such as erasable pattern mining (Lee, Yun, & Ryang, 2015b; Yun & Lee, 2016b), periodic pattern mining (Yang, Hong, Lan, & Chen, 2014), and uncertain pattern mining (Lee, Yun, & Ryang, 2015a). Although frequent pattern mining has played an important role in the data mining field, it cannot consider both relative importance of items and non-binary transactions. Weighted (Ahmed, Tanbeer, Jeong, Lee, & Choi, 2012a; Lee et al., 2015b; Vo, Coenen, & Le, 2013; Vo, Hong, & Le, 2012; Lee, Yun, Ryang, & Kim, 2016) and share (Barber & Hamilton, 2000; Barber & Hamilton, 2003) frequent pattern mining were proposed to address this

* Corresponding author.
E-mail addresses: ryang@sju.ac.kr (H. Ryang), yunei@sejong.ac.kr (U. Yun).

http://dx.doi.org/10.1016/j.eswa.2016.03.001
0957-4174/© 2016 Elsevier Ltd. All rights reserved.

issue. In each framework, patterns with high weight or quantity values can be extracted even if they occur infrequently. In real world applications such as market analysis, however, each item not only has a relative importance but also is represented with a non-binary value in a transaction. For instance, in market databases, an item has its own different price or profit and can be sold multiple copies in a transaction. Hence, to satisfy this real world scenario, both item significance and non-binary quantity have to be reflected at the same time.

In view of this, utility mining (Liu, Liao, & Choudhary, 2005; Tseng, Shie, Wu, & Yu, 2013; Tseng, Wu, Shie, & Yu, 2010; Yun, Ryang, & Ryu, 2014; Ryang, Yun, & Ryu, 2016; Yin, Zheng, & Cao, 2012; Lan, Hong, & Tseng, 2014) has emerged as an essential research topic. In this framework, items have two types of utilities: (1) external utility such as price or profit and (2) non-binary item quantity in a transaction, i.e., internal utility. Given a pattern, its utility is defined as the sum of the products of external and internal utilities of items in the pattern. A pattern is a high utility pattern when its utility is no smaller than a user-specified minimum utility threshold; otherwise, it is a low utility one. High utility pattern mining is a series of processes to find a set of high utility patterns, and this reflects real world market data (Ahmed, Tanbeer, Jeong, & Lee, 2009). Accordingly, it can play an important role in market analysis. Furthermore, it can be used not only in other application areas (e.g., cross-marketing in retail stores and mobile commerce environment planning (Shie, Hsiao, Tseng, & Yu, 2011)) but also with other pattern mining concepts (e.g., maximal pattern mining (Lin, Tu, & Hsueh, 2012), closed pattern mining (Wu, Fournier-Viger, Yu, & Tseng, 2011), episode pattern mining (Wu, Lin, Yu, & Tseng, 2013), rare pattern mining (Ryang, Yun, & Ryu, 2014; Kim & Yun, 2016), and top-*k* pattern mining (Wu, Shie, Tseng, & Yu, 2012; Tseng, Wu, Fournier-Viger, & Yu, 2016)).

In recent years, many applications have been generating stream data such as online click streams, sensor data from wireless sensor networks, transactions of retail markets, large sets of web pages, and telephone call records. These data show distinct characteristics such as very fast arrival rate and unbounded length. Moreover, it is difficult to backtrack to previously arrived information. Due to the unbounded length, it is intractable to handle the entire data in main memory using previous methods for static transactional databases. For this reason, mining useful patterns over continuous data streams has been researched (Chen, Shu, Xia, & Deng, 2012; Chen & Mei, 2014; Chu, Tseng, & Liang, 2008; Li & Lee, 2009; Tsai, 2010) as a significant topic for wide application areas such as retail market analysis, wireless sensor networks, web usage mining, network traffic analysis, and stock market prediction. There are three major stream processing models (Li et al., 2009): (1) damped window model, (2) landmark window model, and (3) sliding window model, where a window is a set of continuous transactions treated as a unit in data streams. In the sliding window model (Chen et al., 2012; Tsai, 2010), only recent data in a fixed size window are employed to discover meaningful patterns over data streams. This model is widely used for stream mining because of its ability to emphasize recent data and require bounded memory resources. Sliding window-based high utility pattern mining (Ahmed, Tanbeer, Jeong, & Choi, 2012b; Li, Huang, & Lee, 2011; Li, Huang, Chen, Liu, & Lee, 2008) has been proposed to consider the characteristics of real-world databases over stream data in resource-limited environments. To satisfy the downward closure property, previous sliding window-based studies utilize an overestimation concept (Liu et al., 2005). However, they often lead to a generation of a huge number of candidate patterns due to the use of overestimated utilities, and as a result, their mining performance is degraded. The reason for this is that more execution time is needed to handle the greater number of candidate patterns (Tseng et al., 2013; Yun et al., 2014). In order to perform as fast as possible, therefore, the number of extracted candidates has to be decreased, through which we can improve mining performance and deal with stream data efficiently. Accordingly, this study aims to develop an algorithm to improve performance of high utility pattern mining over non-binary data streams, not static databases (Tseng et al., 2013; Tseng et al., 2010; Yun et al., 2014; Ryang et al., 2016), by reducing overestimated utilities and search space effectively. In stream mining, furthermore, an appropriate method for updating a data structure is necessary to handle stream data efficiently based on the sliding window model since the current window slides whenever it is full and recent data arrive by eliminating the oldest information and reflecting the recent ones.

In this paper, motivated from the above, we propose an efficient sliding window-based algorithm for mining high utility patterns over data streams to improve mining performance of the previous overestimation approach-based ones. That is, a major objective of this paper is to facilitate efficient high utility pattern mining in stream environments where a rapid process is necessary by decreasing overestimated utilities in tree data structures constructed through a single-scan with an appropriate updating method. Main contributions of this paper for this purpose are summarized as follows: (1) devising a novel sliding window-based data structure, (2) developing both an updating method for a global data structure constructed by a single-scan and a technique to reduce overestimated utilities in the construction and updating processes, (3) proposing an algorithm for mining high utility patterns efficiently from data streams by decreasing overestimated utilities in local data structures with the other technique, and (4) providing various experimental results for performance evaluation between the proposed algorithm and the previous one.

The remainder of this paper is organized as follows. We describe the related work in Section 2. Section 3 illustrates our sliding window-based tree structure and mining algorithm with techniques for reducing overestimated utilities in detail. Moreover, an updating method for the tree to handle stream data on the basis of a sliding window is explained in this section. We show and analyze experimental results for performance evaluation in Section 4. Finally, Section 5 concludes our contributions.

2. Related work

2.1. Frequent pattern mining

Apriori (Agrawal et al., 1994) is the initial solution for frequent pattern mining based on the anti-monotone property. This algorithm has two major drawbacks causing performance degradation, a level-wise candidate generation-and-test approach and a series of multiple database scans. FP-Growth (Han, Pei, & Yin, 2000) was then suggested to improve mining performance of Apriori-based methods with a pattern growth approach. Various relevant algorithms were afterward proposed such as MFS_doubleCons (Duong, Truong, & Vo, 2014) that satisfies two opposite types of constraints, anti-monotone and monotone constraints, and IMIT (Pasquier, Sanhes, Flouvat, & Selmaoui-Folcher, 2016) that mines frequent patterns in a form of a collection of attributed trees. Recently, PrePost+ (Deng & Lv, 2015) has been developed to achieve faster runtime speed than the existing methods for frequent pattern mining. Meanwhile, infrequent pattern mining (Hemalatha, Vaidehi, & Lakshmi, 2015) was also suggested for some domains such as outlier detection. This traditional framework is weak for discovering useful information on expert and intelligent systems since it only considers a frequency factor with binary databases.