Expert Systems with Applications 41 (2014) 694–708

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Sliding window based weighted maximal frequent pattern mining over data streams

Gangin Lee[a], Unil Yun[a,*], Keun Ho Ryu[b]

[a] Department of Computer Engineering, Sejong University, Seoul, South Korea
[b] Department of Computer Science, Chungbuk National University, South Korea

ARTICLE INFO

Keywords:
Data mining
Data stream
Sliding window
Weighted maximal frequent pattern mining

ABSTRACT

As data have been accumulated more quickly in recent years, corresponding databases have also become huger, and thus, general frequent pattern mining methods have been faced with limitations that do not appropriately respond to the massive data. To overcome this problem, data mining researchers have studied methods which can conduct more efficient and immediate mining tasks by scanning databases only once. Thereafter, the sliding window model, which can perform mining operations focusing on recently accumulated parts over data streams, was proposed, and a variety of mining approaches related to this have been suggested. However, it is hard to mine all of the frequent patterns in the data stream environment since generated patterns are remarkably increased as data streams are continuously extended. Thus, methods for efficiently compressing generated patterns are needed in order to solve that problem. In addition, since not only support conditions but also weight constraints expressing items' importance are one of the important factors in the pattern mining, we need to consider them in mining process. Motivated by these issues, we propose a novel algorithm, weighted maximal frequent pattern mining over data streams based on sliding window model (WMFP-SW) to obtain weighted maximal frequent patterns reflecting recent information over data streams. Performance experiments report that MWFP-SW outperforms previous algorithms in terms of runtime, memory usage, and scalability.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

One of the data mining areas, frequent pattern mining has been actively studied together with various approaches and widely applied in numerous fields such as industry and business as well as computer science. As well-known fundamental frequent pattern mining algorithms, there are Apriori (Agrawal & Srikant, 1994) based on Breadth First Search and FP-growth (Han, Pei, Yin, & Mao, 2004) on the basis of Depth First Search. On the basis of those basic algorithms, a variety of pattern mining algorithms have been proposed, such as frequent pattern mining without the minimum support threshold specified by users (Chuang, Huang, & Chen, 2008; Li, 2009; Zhang & Zhang, 2011), sequential frequent pattern mining (Chang, Wang, Yang, Luan, & Tang, 2009; Muzammal & Raman, 2011; Yun, Ryu, & Yoon, 2011). Furthermore, frequent pattern mining has been utilized in extensive applications such as medical and bio data analysis (Sallaberry, Pecheur, Bringay, roche, & Teisseire, 2011; Xiong, He, & Zhu, 2010), stock market and protein networks (Sim, Li, Gopalkrishnan, & Liu, 2009), network environment (Fang, Deng, & Ma, 2009; Lin, Hsieh, & Tseng, 2010), traffic data analysis (Liu, Zheng, Chawla, Yuan, & Xing, 2011), analysis of web-click streams (Li, 2008; Li, Lee, & Shan, 2006), and so on. Frequent pattern mining can be applied not only in static databases like the above methods but also in data streams. Data streams mean that transaction data are added constantly, and thus, they have continuous and unlimited features. Note that data stream mining has to satisfy the following requirements (Farzanyar, Kangavari, & Cercone, 2012). (1) Each data element needed for data stream analysis has to be examined only once. (2) Although data streams become constantly large as data elements are continuously added, memory usage for mining operations should be limited to an acceptable and constant range. (3) All of the entered data elements have to be processed as soon as possible. (4) Results of data stream analysis should be available instantly as well as their quality should also be acceptable whenever users want the results. However, the previous frequent pattern mining methods do not satisfy these requirements since they have to conduct two or more database scans to mine frequent patterns. Therefore, to overcome these problems, mining approaches with only one scan (Tanbeer, Ahmed, Jeong, & Lee, 2009a, 2009b) have been suggested. Although these data stream mining methods can extract frequent patterns over data streams effectively, there are still the following issues. In data streams, data elements are constantly added and their sizes are continuously increased according to

* Corresponding author. Tel.: +822 34082902.
E-mail addresses: ganginlee@sju.ac.kr (G. Lee), yunei@sejong.ac.kr (U. Yun), khryu@chungbuk.ac.kr (K.H. Ryu).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.07.094

accumulation of transaction data. Therefore, frequent patterns generated over data streams also become large, which means spending a lot of time mining the patterns, and thereby it can violate one of the requirements for the data stream mining, immediate processing. In order to solve the problem, closed frequent pattern (CFP) and maximal frequent pattern (MFP) notations Burdick, Calimlim, Flannick, Gehrke, & Yiu, 2005; Chen, Bie, & Xu, 2011; Farzanyar et al., 2012; Grahne & Zhu, 2005; Gouda & Zaki, 2005; Huang, Xiong, Wu, Deng, & Zhang, 2007; Li, 2009; Luo & Chung, 2008, 2012; Priya, Vadivel, & Thakur, 2012; Selvan & Nataraj, 2010; Shiozaki, Ozaki, & Ohkawa, 2006; Thomas, Valluri, & Karlapalem, 2006; Yang, Li, Zhang, & Hu, 2007; Yun, Shin, Ryu, & Yoon, 2012; Zeng, Pei, Wang, & Li, 2009, which can represent general frequent patterns as more compact forms, can be utilized. The MFP notation guarantees more efficient pattern compressibility than that of the CFP notation although slight pattern losses can occur when MFPs are again converted into the general ones. Consequently, if the MFP method with outstanding compressibility is applied into the data stream mining, we can find valid patterns over data streams more efficiently due to its advantage. As data have been accumulated in data streams continuously, importance of certain data entered a long time ago can decline or they may be no longer needed, while that of recently added data can be relatively high. To apply these characteristics in the mining process, a variety of window model-based mining approaches (Ahmed, Tanbeer, Jeong, & Lee, 2009; Chen, Shu, Xia, & Deng, 2012; Deypir, Sadreddini, & Hashemi, 2012; Farzanyar et al., 2012; Li, 2011; Mozafari, Thakkar, & Zaniolo, 2008; Shie, Yu, & Tseng, 2012; Tanbeer et al., 2009b; Zhang & Zhang, 2011) have been proposed, and damped window, landmark window, and sliding window techniques can be selectively applied according to characteristics of data streams. Especially since the sliding window-based mining approaches perform mining operations with only the most recent data among accumulated data streams, we can obtain recent high-quality results by using them. Data streams (or stream databases) are composed of numerous items, where each item represents objects in the real world. For example, in a retail market data stream, items reflect information regarding products, and in a data stream for traffic accidents, each item becomes accident information, where importance assigned to each item is actually different. Thus, we can obtain high-quality mining results reflecting not only items' frequency (or support) but also their importance (or weight) by applying the weight factor into the data stream mining. In this paper, we propose a novel algorithm satisfying the aforementioned issues, called weighted maximal frequent pattern mining over data streams based on sliding window model (WMFP-SW). To our knowledge, it is the first approach for mining weighted maximal frequent patterns (WMFPs) over sliding window model-based data streams. Through the proposed algorithm, we can always extract mining results regarding the latest data over data streams, and can gain the resulting patterns more quickly through the MFP technique and weight conditions. The main contributions of this work are summarized as follows.

1. We introduce a novel algorithm, WMFP-SW which can efficiently mine WMFPs with only one scan over sliding window-based data stream environment and a tree structure, WMFP-SW-tree used for the WMFP mining work. We also describe another tree structure, WMFP-tree managing WMFP information and performing subset-checking tasks effectively and an array structure, WMFP-SW-array for improving efficiency of mining operations. We help understand mining processes of the proposed algorithm by providing various examples.
2. Pruning strategies for reducing needless mining operations efficiently are described. Since WMFP-SW considers not only patterns' supports but also their weights when it decides whether extracted patterns are valid or not, the corresponding pruning range becomes lager than that of general frequent pattern mining. In addition, elements except for the latest ones are excluded in the mining procedure by the sliding window model, and thereby WMFP-SW conducts mining operations with faster runtime and less memory usage. We also provide a strategy which can prune unnecessary operations causing meaningless pattern generation in single paths.
3. To evaluate performance of the proposed algorithm, we compare ours with previous state-of-the-art algorithms, and various real and synthetic datasets applying weight conditions are used in performance experiments. These experimental results show that WMFP-SW presents more outstanding performance compared to the previous ones.

The remainder of this paper is organized as follows. Related work for this paper is introduced in Section 2, and thereafter, we describe details of the proposed algorithm, data structures, and pruning techniques in Section 3. Results of performance evaluation for ours and previous algorithms are presented in Section 4, and finally we conclude this paper in Section 5.

2. Related work

As an early frequent pattern mining algorithm, Apriori (Agrawal & Srikant, 1994) finds frequent patterns over static databases. The algorithm performs mining operations in Breadth First Search (BFS) manner and has to generate numerous candidate patterns in the process of actual frequent patterns. Moreover, to obtain complete results of frequent patterns, the algorithm should scan databases repeatedly, and especially in the worst case, the scanning task has to be performed as many as the number of items of the longest transaction in a database. Thereafter, FP-Growth algorithm (Han et al., 2004) based on Depth First Search (DFS) was proposed in order to overcome that problem, and most of the numerous algorithms suggested so far are on the basis of the framework and techniques of FP-growth. The algorithm can more efficiently conduct mining work with two fixed database scans and does not generate candidate patterns in comparison to Apriori.

2.1. Sliding window-based frequent pattern mining over data streams

Although mining methods based on FP-Growth have an effect on static databases, they are not suitable for data streams accumulating data continuously. Since these methods perform more than two database scans, they do not deal with data streams instantly. Moreover, since they construct trees with items remained after infrequent items are deleted, they have to discard previously generated trees and build new trees again if new transaction data are added into data streams. In data streams, although a certain item is currently infrequent, it can become frequent one according to addition of new transaction data. However, those two scan-based methods must read databases from the first again since they already eliminated infrequent items in the previous step. To solve this, mining methods suitable for data streams (Ahmed, Tanbeer, Jeong, Lee, & Choi, 2012; Chen & Wang, 2010; Tanbeer et al., 2009a) have been proposed, and they can perform mining tasks with only one database scan, thereby responding to changes of data streams immediately. After that, sliding window-based frequent pattern mining approaches (Ahmed et al., 2009; Chen et al., 2012; Deypir et al., 2012; Farzanyar et al., 2012; Li, 2011; Mozafari et al., 2008; Shie et al., 2012; Tanbeer et al., 2009b; Zhang & Zhang, 2011) have been proposed, which can mine frequent patterns considering the latest transaction data of large data streams. Especially in those paper (Tanbeer et al., 2009a, 2009b), an efficient

Download English Version:

https://daneshyari.com/en/article/382579

Download Persian Version:

https://daneshyari.com/article/382579

Daneshyari.com