Knowledge-Based Systems 55 (2014) 49–65

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Mining maximal frequent patterns by considering weight conditions over data streams

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

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

ARTICLE INFO

Article history:
Received 28 March 2013
Received in revised form 26 September 2013
Accepted 6 October 2013
Available online 23 October 2013

Keywords:
Data stream
Data mining
Maximal frequent pattern mining
Weight condition
Knowledge discovery

ABSTRACT

Frequent pattern mining over data streams is currently one of the most interesting fields in data mining. Current databases have needed more immediate processes since enormous amounts of data are being accumulated and updated in real time. However, existing traditional approaches have not been entirely suitable for a data stream environment since they operate with more than two database scans. Moreover, frequent pattern mining over data streams mostly generates an enormous number of frequent patterns, thereby causing a significant amount of overheads. In addition, as weight conditions are very useful factors in reflecting importance for each object in the real world, it is necessary to apply them to the mining process in order to obtain more practical, meaningful patterns. To consider and solve these problems, we propose a novel method for mining Weighted Maximal Frequent Patterns (WMFPs) over data streams, called MWS (Maximal frequent pattern mining with Weight conditions over data Streams). MWS guarantees efficient mining performance in the data stream environment by scanning stream databases only once, and prevents overheads of pattern extractions with an abbreviated notation: a maximal frequent pattern form instead of the general one. Furthermore, MWS contributes to enhanced reliability of the mining results by applying weight conditions to each element of the data streams. Extensive experiments report that MWS has outstanding performance in comparison to previous algorithms.

1. Introduction

Data mining means finding potentially useful and hidden information from large databases, and frequent pattern mining, one of the data mining fields, plays an important role in extracting meaningful information, i.e. frequent patterns from databases. Apriori [1] and FP-growth [15], which are regarded as fundamental frequent pattern mining methods, have become important criteria in numerous frequent pattern mining studies and applications. Researchers in this field have proposed various methods [6,13,16,37,40,43,46] for mining frequent patterns more efficiently and quickly, and advanced studies are being developed continuously. In addition, there are extensive and various approaches such as mining sequential frequent patterns [7,8,28], top-k frequent patterns without any threshold [10,25,40,46], frequent patterns over data streams [2,4,7–9,17,36], and weighted frequent patterns [3,39,41,44]. The frequent pattern mining is also utilized in a variety of applications such as geographic pattern mining [5], network environment [11,23], web click stream analysis [18,19], traffic data analysis [24], bio and medical data analysis [32,42], and stock market and protein networks [35]. Furthermore, it can be used in graph databases [9,38,47] as well as transactional databases, streaming databases, and sequential databases. Frequent pattern mining over data streams [2–4,7–9,17,20,37,46] is one of the issues receiving the most attention in the data mining research field. It performs a series of mining operations to extract valid information from dynamic databases that are constantly updated from data streams. This method needs more constraints than frequent pattern mining from static databases, and therefore, dealing with its operations is a difficult task due to many considerations. In particular, since mining frequent patterns from data streams requires fast and immediate processing with respect to databases, it is not suitable for data streams to apply FP-growth-based algorithms with 2 database scans as well as Apriori-based ones with more than 2 scans. Therefore, to satisfy the requirements of data stream mining, we need to mine frequent patterns by scanning databases only once. Nevertheless, it is hard to extract all frequent patterns whenever there is any mining request because data streams are constantly changing and become large as new elements are added

* Corresponding author. Tel.: +82 234082902.
E-mail addresses: yunei@sejong.ac.kr (U. Yun), ganginlee@sju.ac.kr (G. Lee), khryu@cbnu.ac.kr (K.H. Ryu).

0950-7051/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2013.10.011

continuously. Moreover, the lower a given minimum support threshold becomes, the more exponentially the number of resulting frequent patterns increase. Accordingly, their computation overheads are also sharply increased. To overcome this problem, two types of abbreviated notations can be used. They are called CFP (Closed Frequent Pattern) [7,13,20,40] and MFP (Maximal Frequent Pattern) [6,8,13,14,16,17,26,27,30,33,44,45]. Between them, applying MFP to data stream mining helps find valid patterns over data streams more efficiently due to the MFP's high compression ratio. However, the studies mentioned above consider only the support conditions but do not distinguish the weights of items. Every object in the real world has a unique weight since their importance is different from one another. Thus, these values are one of the important factors reflecting the information of the real world, and we can obtain more useful results by considering them. In addition, since weights can also be used as a strong pruning condition, we can remove patterns that become invalid by applying the weights, and therefore, this leads to enhancement of mining performance. Thus, in this paper, we propose a novel method, called MWS (Maximal frequent pattern mining with Weight conditions over data Streams), and we also suggest MWS-tree used for mining WMFPs, WMFP-tree for storing WMFP information and conducting subset checking operations, and WMFP-array which can improve mining efficiency by reducing tree scans. This method is the first approach that mines maximal frequent patterns that considers the weight conditions for each element in patterns over data streams. Our main motivation is to mine more important patterns more efficiently over data streams. In the data stream environment, data are accumulated continuously. Then, the number of frequent patterns which can be generated from this environment is exponentially increased depending on the degree of data accumulation, where the important considerations are that mining these numerous patterns causes enormous operational overheads and that it is hard to conduct pattern analyses with respect to the numerous patterns. As mentioned earlier, the weight of items or patterns is a fairly important factor that can express their actual importance. Therefore, it is essential to select patterns with a high importance among the numerous ones, which makes it possible to reduce the operational overheads and conduct a pattern analysis more easily by preventing the generation of less important patterns. However, there is still the following problem. Although we can reduce the number of generated patterns and selectively mine important ones due to the weight constraint, numerous patterns are still mined because of the features of the data stream. Thus, there is a need to mine a smaller number of patterns that can represent all the patterns. In this regard, the maximal frequent pattern mining technique completely fulfills this requirement. In summary, applying the weight constraint and extracting representative patterns allow us to achieve the main goal of this paper: mining more important patterns more efficiently over data streams. The proposed algorithm, MWS, also satisfies the "build one mine many" property since this can use a previously constructed global MWS-tree many times regardless of the threshold's change, unless the tree is restructured due to new data entered into data streams. Major contributions of this paper are as follows.

1. The MWS algorithm, which can extract WMFPs over data streams by scanning databases only once, is proposed, and MWS conducts mining operations by using a proposed tree structure, called MWS-tree. Moreover, to perform subset-checking operations effectively, we use a special tree structure, WMFP-tree storing valid WMFPs, and to reduce the number of conditional MWS-tree scans and advance mining efficiency, a two-dimensional array structure, a WMFP-array is also suggested and applied to our MWS method. A series of steps for mining WMFPs through MWS are explained with several examples in this paper in detail.
2. Various mining strategies are proposed. First, MWS removes meaningless patterns through their weight considerations, and thereby, the method contributes to reducing both runtime and memory usage. Second, a strategy by the WMFP-array is suggested, where we can construct conditional MWS-trees and mine WMFPs faster due to this strategy. The last strategy is applied in trees with a single path form. We can find valid patterns from single paths more effectively through this strategy.
3. In order to demonstrate the effectiveness of the proposed method, MWS, we compare ours to state-of-the-art algorithms with respect to a variety of real and synthetic datasets. Experimental results report that MWS outperforms the compared algorithms in terms of runtime, memory usage, and scalability.

The remainder of this paper is organized in the following sequence. In Section 2, we introduce the background related to this paper, and in Section 3, details of MWS and proposed techniques and strategies are described. In Section 4, Extensive experimental results are presented to demonstrate the performance of MWS, and we finally conclude this paper in Section 5.

2. Background

2.1. Related work

Starting from the Apriori method [1] with a level-wise search, frequent pattern mining has been studied actively, and various approaches expanding Apriori have been proposed. However, this method has to perform multiple database scans to find frequent patterns. Especially in the worst case scenario, the method must scan a database by the number of items belonging to a transaction with the longest length. Thereafter, to overcome this problem, FP-growth [15] was proposed. It can mine frequent patterns by scanning a database twice, where the method uses its special tree structure called FP-tree and DFS (Depth First Search) technique to reduce the number of database scans. Accordingly, most of the approaches that have been proposed recently are based on the FP-growth.

2.1.1. Frequent pattern mining over data streams

The above methods with multiple scans are not suitable for data streams in which data are constantly added and changed since they cannot immediately respond to the changes. Let us assume that any mining algorithm conducts multiple scans like FP-growth over data streams. Then, the algorithm reads a streaming database once to confirm support of items in the database and reread it to mine patterns. However, if new transactions are added into a streaming database or there are some changes in the database in the middle of mining process, this multi-scan method causes a serious problem. When the method scans transactions from data streams, the result of its first scan may be different from the next scan because transactions can be added continuously over data streams. In this case, this approach cannot guarantee the exact intended mining results. Otherwise, adding transactions from data streams has to be delayed until the current mining process is completed to prevent the algorithm from producing the wrong results. To solve the problem, researchers have conducted studies which can mine patterns with only one database scan, and a variety of approaches have been published, such as [4,7–9,20,37,40,46]. In [36], Tanbeer et al. proposed a frequent pattern mining algorithm with one scan, where they introduced a new restructuring technique,