Future Generation Computer Systems 59 (2016) 1–20

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Sliding window based weighted erasable stream pattern mining for stream data applications

Unil Yun *, Gangin Lee

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

HIGHLIGHTS

- We propose sliding window based stream pattern mining algorithm that finds weighted erasable patterns.
- We devise strategies for pruning techniques guaranteeing efficiency of the proposed algorithm.
- We suggest performance improving stream data mining techniques for the stream data applications.
- We provide extensive, comprehensive performance evaluation results.

ARTICLE INFO

Article history:
Received 15 June 2015
Received in revised form
5 October 2015
Accepted 18 December 2015
Available online 8 January 2016

Keywords:
Weighted erasable pattern mining
Sliding window
Weight condition
Data stream
Data mining

ABSTRACT

As one of the variations in frequent pattern mining, erasable pattern mining discovers patterns with benefits lower than or equal to a user-specified threshold from a product database. Although traditional erasable pattern mining algorithms can perform their own mining operations on static mining environments, they are not suitable for dealing with dynamic data stream environments. In such dynamic data streams, algorithms have to process them immediately with only one database scan in order to consider characteristics of data stream mining. However, previous tree-based erasable pattern mining methods have difficulty in processing dynamic data streams because they need two or more database scans to construct their own tree structures. In addition, they do not also consider specific information of each item within a product database, but they need to conduct mining operations considering such additional information of the items in order to find more useful erasable pattern results. For this reason, in this paper, we propose a weighted erasable pattern mining algorithm suitable for sliding window-based data stream environments. The algorithm employs tree and list data structures for more efficient mining processes and solves the problems of previous erasable pattern mining approaches by using a sliding window-based stream processing technique and an item weight-based pattern pruning method. We compare performance of the proposed algorithm to state-of-the-art tree-based approaches with respect to various real and synthetic datasets. Experimental results show that our method is more efficient and scalable than the competitors in terms of runtime, memory, and pattern generation.

1. Introduction

Frequent pattern mining [1,2], one of the data mining areas, is a method for discovering useful knowledge of pattern forms from large databases, where each pattern result signifies a set of items related to one another in the databases. Using frequent pattern mining tools, users can obtain a set of frequent patterns with frequency or support values higher than or equal to user-specified threshold. Since *Apriori* [3] and *FP-Growth* [4,5], a variety of works related to frequent pattern mining have actively been studied to provide better performance or more useful information to users. Moreover, frequent pattern mining methods [6,7] have been applied into various user environments such as top-k pattern mining [8], utility pattern mining [9–11], representative pattern mining [12–14,7], network traffic visualizing [15], and incremental pattern mining [16]. As another variation, frequent pattern mining has also been applied in dynamic data streams as well as static databases. Since traditional frequent pattern mining methods perform two or more database scans to analyze a given database, they have limitations of dealing with dynamic data streams that require immediate processing. To deal with such data streams in frequent pattern mining, various concepts and models have been proposed [17,18]. Sliding window-based pattern mining [19,20]

* Corresponding author.
E-mail addresses: yunei@sejong.ac.kr (U. Yun), ganginlee@sju.ac.kr (G. Lee).

http://dx.doi.org/10.1016/j.future.2015.12.012
0167-739X/

is one of the representative models for processing data streams. This model conducts mining operations employing data structures of sliding window forms to handle dynamic data streams. In particular, it can be used effectively in data stream environments that consider the latest information within bounded memory space.

Meanwhile, erasable pattern mining [21,22] is a variation of frequent pattern mining and finds a set of patterns that can be deleted from a product database on the basis of a user-given threshold. However, traditional erasable pattern mining approaches also have the aforementioned problems in data stream environments. Especially, since the latest tree-based methods [23, 21] perform two or more database scans in order to store information of product databases into their own tree structures, it is hard for the traditional methods to discover erasable patterns considering the characteristics of continuous, dynamic data streams. In addition, they also have limitations that cannot reflect product information in the real world to their mining processes because they do not consider additional information of components such as their prices or importance. In real world applications, components or items composing products have different weights according to their quality, size, price, etc.; therefore, we need to employ such characteristics of components in the erasable pattern mining procedure. Motivated by the aforementioned issues, in this paper, we propose a Weighted Erasable Pattern mining algorithm on Sliding window-based data streams (called *WEPS*). The main contributions of the proposed algorithm are as follows.

1. We devise a new algorithm, *WEPS*, for extracting weighted erasable patterns by considering sliding window-based data stream environments. The algorithm improves efficiency of the mining operations by employing a new tree structure suitable for weighted erasable pattern mining and the sliding window model, and a new list structure including essential information for the mining process. In addition, empirical examples are provided to help clear understanding of the proposed method.
2. We propose pattern pruning techniques based on constraints of component weights. Through the proposed technique, we not only reduce the number of unnecessary mining operations in *WEPS* but also improve its efficiency. Moreover, the algorithm can discover more useful pattern results compared to the previous approaches.
3. We provide extensive experimental results and their analyses of the proposed algorithm. Various real and synthetic datasets are used in our tests, and state-of-the-art tree-based erasable pattern mining methods are compared to our *WEPS* in terms of runtime, memory, pattern generation, and scalability. The results show that the proposed algorithm is more outstanding than the previous methods.

The remainder of this paper is as follows. In Section 2, previous studies related to our method are introduced, and in Section 3, details of the proposed algorithm are described with empirical examples. In Section 4, we show results of the performance evaluation of our algorithm and previous ones, and in Section 5, we finally conclude this paper.

2. Related work

2.1. Frequent pattern mining based on the sliding window model

Sliding window-based methods [24,25] have been proposed to solve dynamic data stream problems of traditional frequent pattern mining. Unlike traditional approaches that require two or more database scans for mining patterns, the methods construct their own data structures within a single database scan on the basis of sliding window structures and perform pattern mining operations. As one of the sliding window-based frequent pattern mining methods, *pWin* [26] solved the problems of previous algorithms consuming heavy memory by considering characteristics of data stream environments where available memory is limited and states of data are continually changed. In addition, the algorithm used a prefix tree structure for efficient tree searches. Another sliding window-based algorithm using *SWP-tree* [20] employed a time decay model to discover interesting patterns from historical transactions. Sliding window-based mining has been also utilized in other mining areas such as maximal frequent pattern mining and top-k frequent pattern mining. *WMFP-SW* [27] is an algorithm for mining maximal frequent patterns over sliding window-based data streams. The algorithm considers the latest data stream information and weight conditions of items in the process of weighted maximal frequent pattern mining. In addition, the sliding window model can be applied in closed pattern mining [28]. However, the above approaches only focus on the traditional pattern mining framework but do not deal with erasable pattern mining. Meanwhile, the proposed method can consider the erasable pattern mining framework together with the weight conditions of items and sliding window-based data streams.

2.2. Erasable pattern mining based on tree structure

Erasable pattern mining is a method that discovers all of the erasable patterns from a given product database according to a maximum gain threshold given by the user. In particular, unlike traditional frequent pattern mining approaches, erasable pattern miners find patterns with gain values smaller than or equal to a given threshold. *META* [29] is a representative erasable pattern mining algorithm based on a Breadth-First Search (BFS) manner. *META* performs *Apriori*-like mining operations to mine erasable patterns; therefore, it also operates in a generate-and-test manner, which causes excessive candidate pattern creations. *MERIT* [30] is another approach for discovering such patterns on the basis of tree data structures. The algorithm was proposed to solve the efficiency problem of *META*, and it effectively improved the performance by using its own data structures, *WPPC-tree* and *NC-Set*. However, *MERIT* has several problems as follows. The first one is the error occurring when the algorithm uses its equivalent class-based pattern expanding technique. Because of this error, the algorithm may cause a fatal pattern loss problem during the mining process [31,23]. The second problem is that it requires two database scans to mine patterns and does not consider the different importance of each item in product databases. *MERIT+* [23] is a method solving the pattern loss problem of *MERIT* by excluding this defective technique. *dMERIT+* [23] is another approach based on *MERIT+*. The algorithm additionally employs a hash table and an advanced version of *NC-set*, *dNC-set*, in order to improve mining efficiency. With more sophisticated considerations for the ancestor–descendant relations, the algorithm constructs *dNC-set* by removing duplicated information that can occur in *NC-set*. The hash table of the algorithm is used to map post-order information to a table form. In addition, there are other erasable pattern mining algorithms based on list data structures, *VME* [32] and *MEI* [31]. *VME* employs its own data structure, *PID_set*, to mine erasable patterns in a different way from the tree-based methods. *MEI* also follows a similar way with *VME*, but it additionally considers the difference of indexes by using its data structure, *dPID_set*. In spite of the various efforts to improve erasable pattern mining techniques, there is no algorithm for mining erasable patterns over sliding window-based data streams before the proposed algorithm, *WEPS*. In addition, our method is a more advanced approach that can also deal with different importance of items in data streams.