Engineering Applications of Artificial Intelligence 72 (2018) 111–123

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Efficient incremental high utility pattern mining based on pre-large concept

Judae Lee [a], Unil Yun [a,*], Gangin Lee [a], Eunchul Yoon [b,*]

[a] Department of Computer Engineering, Sejong University, Seoul, Republic of Korea
[b] Department of Electronics Engineering, Konkuk University, Seoul, Republic of Korea

ARTICLE INFO

Keywords:
Data mining
High utility patterns
Incremental mining
Utility mining
Pre-large

ABSTRACT

High utility pattern mining has been actively researched in recent years, because it treats real world databases better than traditional pattern mining approaches. Retail data of markets and web access information data are representative examples of the real world data. However, fundamental high utility pattern mining methods aiming static data are not proper for dynamic data environments. The pre-large concept based methods have efficiency compared to static approaches when dealing with dynamic data. There are several methods dealing with dynamic data based on the pre-large concept, but they have drawbacks that they have to scan original data again and generate many candidate patterns. These two drawbacks are the main issues of performance degradation. To handle these problems, in this paper, we suggest an efficient approach of pre-large concept based incremental utility pattern mining. The proposed method adopts a more proper data structure to mine high utility patterns in incremental environments. The state-of-the-art method performs a database scan operation many times, which is not suitable for incremental environments. However, our method needs only one scan, which is more suitable to process dynamic data compared to the state-of-the-art method. In addition, with the proposed data structure, high utility patterns can be mined in dynamic environments more efficiently than the former method. Experimental results on real datasets and synthetic datasets show that the proposed method has better performance than the former method.

1. Introduction

Pattern mining is one of data mining techniques. The pattern mining area (Ahmed et al., 2012; Feng et al., 2013; Lee et al., 2016b; Yun and Lee, 2016b) has interests in finding useful pattern information in databases. Frequent pattern mining (Lee et al., 2016a, 2015a), which has an important role in the pattern mining area, expresses item information of databases as a binary form, so it has a limitation that it cannot consider real data's non-binary character. For example, retail databases of markets include prices and quantity of items, which are not binary data. To handle this limitation, utility pattern mining (Ahmed et al., 2012; Kim and Yun, 2016; Lin et al., 2016; Ryang et al., 2016) was proposed. Utility pattern mining uses database's non-binary data like prices and quantity. In addition, various fields related to utility pattern mining such as Top-K (Duong et al., 2016; Ryang and Yun, 2015; Tseng et al., 2016) have been researched. To satisfy the anti-monotone property, which is used importantly in frequent pattern mining, is hard in high utility pattern mining area because the extensions of patterns can increase their utilities unexpectedly. Utility pattern mining, to maintain the anti-monotone property, adopts the transaction weighted utilization (TWU) concept (Liu et al., 2005), and various studies based on the TWU concept have been conducted.

A great part of high utility pattern mining studies have been focused on the static database. However, databases utilized in the real world are dynamic, so existing methods are not proper to process real world databases. The reason is that the methods for static databases gain efficiency through initial pruning, but, in dynamic circumstances that new data are inserted continuously, the initial pruning processes can yield pattern losses. However, the mining process without initial pruning demands a lot of time and lots of memory. To deal with dynamic databases, various methods were proposed. Stream pattern mining methods (Lu et al., 2014; Manike and Om, 2015; ZiHayat and An, 2014) and incremental pattern mining methods (Hyo et al., 2016; Wang and Huang, 2016; Yun and Lee, 2016a; Zheng and Li, 2015), have been researched. The pre-large concept (Lin et al., 2015b, 2014, 2015a) was also proposed to deal with dynamic database. The pre-large concept employs two thresholds. One is used for mining high utility patterns, and the other is used for a preparation of new data insertion. The methods based on the pre-large concept classify patterns of incremental data into 9 cases. They require database re-scanning in only two cases.

[*] Corresponding authors.
E-mail addresses: jdlee@sju.ac.kr (J. Lee), yunei@sejong.ac.kr (U. Yun), ganginlee@sju.ac.kr (G. Lee), ecyoon@konkuk.ac.kr (E. Yoon).

https://doi.org/10.1016/j.engappai.2018.03.020
Received 6 February 2017; Received in revised form 31 August 2017; Accepted 23 March 2018

Therefore, the methods have better performance than methods which always rescan original data. Even though such characteristics of the pre-large concept, the previous methods with the pre-large concept requires a number of database scans. Therefore, they have performance limitations and are inappropriate for incremental environments.

Motivated by the above problems, in this paper, we propose new data structures and algorithms which adopt the pre-large concept and show better performance than the former state-of-the-are method. This paper has contributions as follows: (1) New data structures are suggested for storing and maintaining data which can be increased, (2) Techniques for storing, maintaining, restructuring and mining data and patterns with the data structure are proposed, and (3) Various experiments are conducted to show our method has better performance than the former ones on real and synthetic datasets. The rest part of this paper is organized with the following contents. In Section 2, influencing studies are explained which are related to the topic of this paper. In Section 3, our suggested data structure and storing, maintaining and mining algorithms are discussed in details. In Section 4, various experiments are conducted on real and synthetic datasets with their analysis results. In the final section, we conclude this paper and summarize contents of this paper.

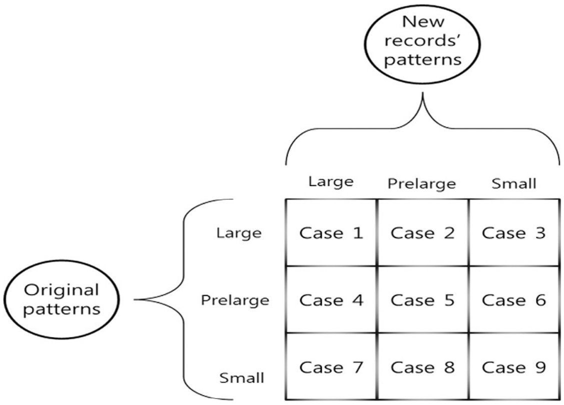
2. Related work

2.1. Frequent pattern mining

Frequent pattern mining is a representative approach in the pattern mining (Chen and Mei, 2014; Guo and Gao, 2016; Lee et al., 2015b) concept. With frequent pattern mining methods, every pattern having supports larger than or equal to a user defines minimum support threshold called *minsup* can be found. In the frequent pattern mining, the downward closure property (Agrawal and Srikant, 1994) is essential for the efficient performance. The property means if pattern X is frequent, than all subsets of X except for an empty set are must frequent. Apriori (Agrawal and Srikant, 1994) is based on the level-wise approach based frequent pattern mining method, so the method performs a number of database scans and generates an enormous numbers of candidate patterns. This is the main problem of the Apriori algorithm, which declines performance of Apriori. To handle this problem, the FP-Growth (Han et al., 2000) algorithm was proposed. It adopts a tree based data structure. FP-Growth builds a data structure called FP-Tree with only two database scans and mines frequent patterns by using a divide and conquer way. However, frequent pattern mining does not consider relative importance of items and expresses appearance of items in a binary form.

2.2. High utility pattern mining

Utility pattern mining utilizes non-binary data. Therefore, it is hard to satisfy the downward closure property in utility pattern mining. To overcome this problem, transaction weighted utilization (TWU) is employed in the Two-Phase (Liu et al., 2005) method, which is based on Apriori. TWU indicates a pattern's overestimated utility value. The anti-monotone property based on TWU concept is called the transaction weighted downward closure (TWDC) property. This means that a pattern which has the TWU value lower than a certain threshold never generates supersets with utilities greater than or equal to the threshold. After the Two-Phase algorithm was proposed, various pattern mining methods (Sahoo et al., 2016; Tseng et al., 2015; Yun et al., 2016) were also proposed for mining high utility patterns efficiently. Recently, for mining patterns more efficiently than TWU based methods, list structure based methods (Lin et al., 2012; Liu et al., 2016, 2012; Liu and Qu, 2012) without candidate generation have been studied.



	Large	Prelarge	Small
Large	Case 1	Case 2	Case 3
Prelarge	Case 4	Case 5	Case 6
Small	Case 7	Case 8	Case 9

Fig. 1. Nine cases occurring from inserting new transactions.

2.3. Pre-large concept based frequent pattern mining

Existing pattern mining methods for static databases do cannot process dynamic databases used in the real world efficiently. Therefore, several incremental pattern mining methods have been proposed. There are Apriori-like incremental methods (Lin et al., 2014), and FP-Growth-like methods (Ahmed et al., 2009; Yun and Ryang, 2015). The pre-large concept is one of the concepts for mining patterns in dynamic circumstances. The disadvantage of existing methods is that the methods have to rescan original data sustainably. In this regard, the pre-large concept improves the performance of the algorithm by reducing the number of re-scans.

The pre-large concept uses cases as shown in Fig. 1 based on characteristics of data. When new data are inserted, some cases require no re-scanning operation, and few cases need re-scanning operation. Hence, methods based on the pre-large concept require a smaller number of re-scanning processes compared to existing methods.

The pre-large concept uses two thresholds. The one is called *upper*, and the other one is called *lower*. A pattern with a utility larger than or equal to *upper* is defined as *large*, and a pattern which has a utility smaller than *upper* and larger than or equal to *lower* is defined as *pre-large*, and a pattern which has a utility smaller than *lower* is defined as *small*.

3. Mining high utility patterns using tree structure with pre-large concept

In this section, a novel method for mining high utility patterns from incremental databases is proposed. Our method adopts the pre-large concept and uses the novel tree-based data structure. Furthermore, we propose algorithms for maintaining data, restructuring the data structure, and mining patterns based on our proposed data structure. After the preliminary introduction, the methods for the construction of the data structure and its maintenance are explained. Thereafter, we describe our incremental high utility pattern mining method using the pre-large concept.

3.1. Preliminaries

In utility pattern mining (Kim and Yun, 2016; Lan et al., 2014; Ryang and Yun, 2016; Sahoo et al., 2016), a database D consists of n non-binary transactions $\{T_1, T_2, \ldots, T_n\}$, containing a set of m distinct items $I = \{i_1, i_2, \ldots, i_m\}$. Each transaction T_d ($T_d \in D$ and $1 \leq d \leq n$) consisting of subsets of I is identified by their own TIDs. Each item $i_p (1 \leq p \leq m)$ has relative importance called an external utility. The external utility is notated as $eu(i_p)$. Furthermore, i_p has a quantity in a transaction T_d called the internal utility, and the internal utility is notated as $iu(i_p, T_d)$.