Knowledge-Based Systems 76 (2015) 109–126

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Top-*k* high utility pattern mining with effective threshold raising strategies

Heungmo Ryang, Unil Yun *

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

ARTICLE INFO

Article history:
Received 21 May 2014
Received in revised form 21 October 2014
Accepted 7 December 2014
Available online 30 December 2014

Keywords:
High utility patterns
Raising a minimum utility threshold
Top-*k* mining
Utility mining
Pattern mining

ABSTRACT

In pattern mining, users generally set a minimum threshold to find useful patterns from databases. As a result, patterns with higher values than the user-given threshold are discovered. However, it is hard for the users to determine an appropriate minimum threshold. The reason for this is that they cannot predict the exact number of patterns mined by the threshold and control the mining result precisely, which can lead to performance degradation. To address this issue, top-*k* mining has been proposed for discovering patterns from ones with the highest value to ones with the *k*th highest value with setting the desired number of patterns, *k*. Top-*k* utility mining has emerged to consider characteristics of real-world databases such as relative importance of items and item quantities with the advantages of top-*k* mining. Although a relevant algorithm has been suggested in recent years, it generates a huge number of candidate patterns, which results in an enormous amount of execution time. In this paper, we propose an efficient algorithm for mining top-*k* high utility patterns with highly decreased candidates. For this purpose, we develop three strategies that can reduce the search space by raising a minimum threshold effectively in the construction of a global tree, where they utilize exact and pre-evaluated utilities of itemsets. Moreover, we suggest a strategy to identify actual top-*k* high utility patterns from candidates with the exact and pre-calculated utilities. Comprehensive experimental results on both real and synthetic datasets show that our algorithm with the strategies outperforms state-of-the-art methods.

1. Introduction

Data mining finds useful information hidden in large databases. One of the data mining techniques, pattern mining discovers meaningful information as pattern forms composed of items. In pattern mining, users generally set a minimum threshold to extract crucial patterns from the databases. As a result, they obtain a set of patterns such that their values are not lower than the threshold. Accordingly, the size of the mining result depends on the user-specified threshold. However, it is not easy for the users to determine an appropriate minimum threshold in real-world applications. The reason for this is that if the value is assigned too high, no useful pattern may be found. Otherwise, if it is set too low, an enormous number of pattern results may be extracted, which degrades mining performance due to the large search space. It signifies that the users cannot predict the exact number of patterns mined by the threshold and control the mining result precisely.

To address this issue, top-*k* mining has been proposed [3,8,23,32]. Instead of minimum threshold settings, it allows users to set the desired number of patterns, *k*, and discovers patterns from ones with the highest value to ones with the *k*th highest value. In top-*k* frequent pattern mining [3,5,11,26], a set of top-*k* frequent patterns is extracted from binary databases of which the items are represented as a binary form in transactions and treated with the same importance. In this framework, the *anti-monotone property* (also known as *downward closure property*) [1] is used to reduce the search space, which makes a significant contribution to improving efficiency of the mining process. The property means that if any pattern is infrequent, all of its possible super patterns are also infrequent, where they include not only all of the items in the pattern but also at least one other item. Items in real-world applications such as retail market data analysis, meanwhile, have their own importance such as profits. Moreover, multiple copies of an item can be sold within a transaction. That is, top-*k* frequent pattern mining cannot consider the above characteristics of real-world databases. Although this problem can be solved with the concept of utility mining [12,14,17,33], it does not satisfy the anti-monotone property. Therefore, it is hard to directly apply into top-*k* utility pattern mining the techniques of top-*k* frequent pattern mining relying upon the anti-monotonicity. For this purpose, overestimation methods [19,29,30] can be employed. A relevant

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

http://dx.doi.org/10.1016/j.knosys.2014.12.010
0950-7051/

algorithm [35] has been suggested in recent years for top-*k* high utility pattern mining with one of the methods [29,30]. However, it generates a huge number of candidate patterns, which lead to performance degradation. In the framework of high utility pattern mining, decreasing extracted candidate patterns is a significant issue since the more candidates an algorithm produces, the greater its execution time becomes [30,41,2]. Consequently, the main challenge of top-*k* high utility pattern mining is how to raise a minimum utility threshold effectively under the overestimation model in order to reduce the candidates for performance improvement.

Top-*k* high utility pattern mining can play a significant role in real-world applications with non-binary databases such as web click stream analysis, mobile commerce environment planning [27], cross-marketing in retail stores, and biological gene database analysis. In retail market data analysis, especially, there is a need to analyze an enormous amount of sales data generated from all branches every day so as to establish sales strategies related to company benefits such as inventory preparation, product arrangement, and promotion. Furthermore, it is necessary to perform a rapid analysis with respect to huge sales databases of the branches during non-opening hours for their smooth running according to plan. Hence, to satisfy this requirement, mining performance in terms of execution time is significant.

In this paper, motivated from the above, we propose an algorithm, called *Raising threshold with Exact and Pre-calculated utilities for Top-k high utility pattern mining* (*REPT*), with strategies that can effectively increase a minimum utility threshold in top-*k* high utility pattern mining, through which we can mine top-*k* high utility patterns efficiently from non-binary databases with item importance by reducing the search space. Major contributions of this paper are summarized as follows:

1. We develop three strategies to raise a minimum utility threshold effectively in the construction of a global tree structure. They utilize exact and pre-evaluated utilities of itemsets with the length of 1 or 2.
2. We suggest a strategy that reduces the search space in the identification process of actual top-*k* high utility patterns from candidates by sorting them and increasing the threshold with the exact and pre-calculated utilities.
3. We also propose an efficient algorithm for top-*k* high utility pattern mining with highly decreased candidates based on the four strategies as well as two other techniques [35].
4. Comprehensive experiments on both real and synthetic datasets are conducted to evaluate performance of the proposed algorithm compared to state-of-the-art ones. Experimental results show that our algorithm outperforms the state-of-the-art methods.

The remainder of this paper is organized as follows. In Section 2, we introduce related work. In Section 3, we illustrate the proposed algorithm with the strategies in detail. In Section 4, we show and analyze experimental results for performance evaluation. Finally, conclusions are given in Section 5.

2. Related work

2.1. Frequent pattern mining

Frequent pattern mining [13,31,39,4] is one of the fundamental researches in data mining, which mines patterns with no smaller supports than a given minimum support threshold. Numerous relevant studies have been conducted [7,22,25,40] including two well-known representative algorithms: Apriori [1] and FP-Growth [9]. Apriori applies a level-wise candidate generation-and-test approach, and thereby it has problems of scanning databases multiple times and extracting a large number of candidates. These problems are the main causes of performance degradation in Apriori-based algorithms. FP-Growth is based on a divide-and-conquer approach with a tree structure, called FP-Tree, to achieve better performance than the Apriori-based ones.

2.2. High utility pattern mining

Two-Phase [19], which is an Apriori-based algorithm, suggested an overestimation method based on *Transaction Weighted Utilization* (*TWU*). In this model, all the super patterns of any pattern with lower TWU than a given minimum threshold also have lower values, and vice versa. Namely, it satisfies the downward closure property. After that, IIDS [17] was proposed to solve the weakness of the Two-Phase algorithm that generates excessively many candidate patterns. However, it is also based on the Apriori method, and thus has to scan databases many times in general and use the level-wise approach. Then, IHUP [2] has been suggested as a solution to the problem, which is based on the FP-Growth method with the TWU model. Two-Phase, IIDS, and IHUP algorithms find all *High Transaction Weighted Utilization Patterns* (*HTWUPs*) with TWUs higher than or equal to a threshold value from a given database in the first step (*Phase I*), and then identify real high utility patterns from the found HTWUPs in the last step (*Phase II*). There is still a problem that they generate too many candidate patterns due to the overestimation method. UP-Growth [29,30] solved the above issue with the following strategies for reducing overestimated utilities of the TWU model: *Discarding Global Unpromising items* (*DGU*), *Decreasing Global Node utilities* (*DGN*), *Discarding Local Unpromising items* (*DLU*), and *Decreasing Local Node utilities* (*DLN*). The algorithm mines high utility patterns with the above four strategies and its own tree structure, called UP-Tree. This data structure is constructed within two database scans, and the UP-Growth algorithm generates *Potential High Utility Patterns* (*PHUPs*) from the tree. Thereafter, it identifies high utility patterns by computing real utility values for the PHUPs through one additional database scan. Besides, UP-Growth+ [29] was suggested with *Discarding local unpromising items and their estimated Node Utilities* (*DNU*) and *Decreasing local Node utilities for local UP-Tree by estimated utilities of descendant Nodes* (*DNN*) instead of the DLU and DLN strategies used by the UP-Growth algorithm in order to more decrease overestimated utilities in local UP-Trees. To improve performance of these tree-based approaches with the overestimation models, MU-Growth [41] proposed two pruning techniques for producing a fewer number of candidate patterns than those of the previous ones. In addition, it utilizes a tree-based data structure, named MIQ-Tree, which includes item quantity values to keep information of the current database and set of high utility patterns. In addition, algorithms with list data structures [20,21] have also been proposed. In the mining process of these list-based methods, list data structures are used as main components in contrast to the previous tree-based algorithms [2,29,30,41]. On the other hand, there have been many studies with other concepts such as privacy pattern mining [38], closed pattern mining [36], maximal pattern mining [28,16], episode pattern mining [34], and incremental mining [15].

2.3. Top-K high utility pattern mining

Top-K Utility itemset mining (*TKU*) [35], which is an algorithm based on UP-Growth [29,30], was proposed for top-*k* high utility pattern mining. Its mining mechanism is as follows. In Phase I, TKU initially assigns a minimum utility threshold as zero and scans a given database once. In this stage, it calculates lower bounds of