Information Sciences 399 (2017) 81–97

Contents lists available at ScienceDirect

Information Sciences

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

A lattice-based approach for mining high utility association rules

Thang Mai[a], Bay Vo[b,c], Loan T.T. Nguyen[d,e,*]

[a] Institute of Research and Development, Duy Tan University, Da Nang, Vietnam
[b] Faculty of Information Technology, Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam
[c] College of Electronics and Information Engineering, Sejong University, Seoul, Republic of Korea
[d] Division of Knowledge and System Engineering for ICT, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[e] Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam

ARTICLE INFO

Article history:
Received 25 August 2016
Revised 25 February 2017
Accepted 28 February 2017
Available online 2 March 2017

Keywords:
Data mining
High utility itemsets
High utility association rules
Lattice

ABSTRACT

Most businesses focus on the profits. For example, supermarkets often analyze sale activities to investigate which products bring the most revenue, as well as find out customer trends based on their carts. To achieve this, a number of studies have examined high utility itemsets (HUI). Traditional association rule mining algorithms only generate a set of highly frequent rules, but these rules do not provide useful answers for what the high utility association rules are. Therefore, Sahoo et al. (2015) proposed an approach to generate utility-based non-redundant high utility association rules and a method for reconstructing all high utility association rules. This approach includes three phases: (1) mining high utility closed itemsets (HUCI) and generators; (2) generating high utility generic basic (HGB) association rules; and (3) mining all high utility association rules based on HGB. The third phase of this approach consumes more time when the HGB list is large and each rule in HGB has many items in both antecedent and consequent. To overcome this limitation, in this paper, we propose an algorithm for mining high utility association rules using a lattice. Our approach has two phases: (1) building a high utility itemsets lattice (HUIL) from a set of high utility itemsets; and (2) mining all high utility association rules (HARs) from the HUIL. The experimental results show that mining HARs using HUIL is more efficient than mining HARs from HGB (which is generated from HUCI and generators) in terms of runtime and memory usage.

1. Introduction

Data mining techniques are to explore the interesting and useful information hidden in databases, where the meaning of interest and usefulness depends on the problem formulation and application domain. Association rule mining plays an important role in many decision systems, and association rules help to determine the relationships among objects in the database.

Traditional association rule mining [1], based on the support-confidence framework, provides an objective measure of the rules that are of interest to users. In this approach, all items are given the same importance by considering the presence

[*] Corresponding author.
E-mail addresses: maihoangthang@duytan.edu.vn, mhthang.it@gmail.com (T. Mai), bayvodinh@gmail.com (B. Vo), nguyenthithuyloan@tdt.edu.vn (L.T.T. Nguyen).

http://dx.doi.org/10.1016/j.ins.2017.02.058
0020-0255/© 2017 Elsevier Inc. All rights reserved.

of items in a transaction, and the utility of items is not observed. Many investigations have been completed on in high utility itemsets mining of (HUI) [6–10,17–19,24–26,28], in which, users specify the utility for all items in the database and an itemset having utility no lower than a minimum utility threshold (*min-util*) is called a high utility itemset (HUI).

The problem of HUIM is widely recognized as more difficult than the problem of frequent itemset mining (FIM) [2]. In FIM, the downward-closure property states that the support of an itemset is anti-monotonic, and hence the supersets of an infrequent itemset are infrequent and the subsets of a frequent itemset are frequent. This property is very powerful with regard to pruning the search space. In HUIM, the utility of an itemset is neither monotonic nor anti-monotonic, and thus a high utility itemset may have a superset or subset with lower, equal or higher utility. Therefore, techniques to prune the search space developed in FIM cannot be directly applied in HUIM, and so many recent algorithms have focused on mining HUI, especially on candidate elimination [8–10,18,19,28]. However, the algorithms for generating association rules from HUI have received little attention.

Sahoo et al. [16] proposed an approach for mining high utility association rules from HUI. The main contributions of this approach are mining non-redundant rules (high utility generic basic association rules – HGB), and mining rules from HGB by considering the current returned rules list to see if there are any rules that can be formed from its antecedent and subset of consequents. However, this approach consumes much time and memory usage, due to its costs to check: (1) whether temporary generated itemsets, which are combined from the antecedent and subset of consequent in current returned rules, are high utility itemsets; and (2) whether the newly generated rules satisfy the threshold or whether those rules are duplicated.

In this paper, we propose an approach to mine high utility association rules using a high utility itemsets lattice. The main contributions of this paper are as follows:

- We propose an algorithm for building HUIL from mined HUIs.
- We propose an algorithm to generate all high utility association rules from HUIL.
- We carry out experiments to show the efficiency of the proposed method, especially with regard to the reusability of HUIL.

The rest of this paper is organized as follows. Section 2 outlines some current works related to mining HUI, the generation of high utility association rules from HUI, and mining association rules from lattices. Section 3 presents some basic definitions and the problem statement. Section 4 describes on our proposed algorithm to build a HUIL from a HUI list. The algorithm for mining all high association rules is presented in Section 5. Section 6 then discusses our experimental results and evaluates the performance of the algorithm with regard to both runtime and memory usage. Conclusions and future works are shown in Section 7.

2. Related works

2.1. High utility itemset mining

High utility itemset mining (HUIM) has been known in the last years to solve the problems that items can appear more than once in each transaction, and each item has a weight (e.g. profit or utility). Several studies on HUIM have been carried out. Liu et al. [9] proposed a two-phase algorithm with the concepts of Transaction Utility (TU) and Transaction Weighted Utility (TWU) to prune the search space of high utility itemsets. Because the TWUs of itemsets satisfy the downward-closure property, we can modify any frequent itemset mining algorithm to mine HUIs. Therefore, the authors modified the Apriori algorithm to mine HUIs. Although this algorithm can reduce the search space of utility mining, it still has performance issues for the following reasons: (1) a high number of candidates are generated with Apriori approach; and (2) the TWU of an itemset is often much higher than its utility.

To address the issue of a large number of candidates being generated, Tseng et al. [18] proposed UP-Growth algorithm. This algorithm includes two steps: (1) construct the UP-Tree; and (2) identify high utility itemsets using a new term potential high utility itemsets (PHUI). In the first stage, the algorithm scans all transactions to accumulate the (TWU) of each item. In the second step of scanning the database, items whose TWU values are less than the specified *min-util* threshold will be removed from each transaction. Four strategies are applied in this algorithm: (1) Discarding Global Unpromising Items (DGU) to eliminate the low utility items and their utilities from transaction utilities; (2) Discarding Global node utilities (DG) to reduce overestimated utilities; (3) Discarding Local Unpromising Items (DLU) to remove the utilities of low utility items from the path utilities of the path; and (4) Discarding Local Node Utilities (DLN) to discard item utilities of descendant nodes during the local UP-Tree construction. Tseng et al. [19] improved the UP-Growth algorithm and proposed UP-Growth+ to reduce the overestimated utilities.

Liu and Qu [10] proposed HUI-Miner to discover high utility itemsets with a list data structure, called a utility list. It firstly creates an initial utility list for itemsets of the length 1 for promising items. The HUI-Miner then recursively constructs a utility list for each itemset of the length k using a pair of utility lists for itemsets of the length $k-1$. For mining high utility itemsets, each utility list for an itemset keeps the information of TIDs for all of the transactions containing the itemset, the utility values of the itemset in the transactions, and the sum of the utilities of remaining items that can be included in super itemsets of the itemset in the transactions.