Knowledge-Based Systems 84 (2015) 67–77

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Mining summarization of high utility itemsets

Xiong Zhang, Zhi-Hong Deng *

Key Laboratory of Machine Perception (Ministry of Education), School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China

ARTICLE INFO

Article history:
Received 8 November 2014
Received in revised form 1 April 2015
Accepted 1 April 2015
Available online 6 April 2015

Keywords:
Data mining
High utility itemsets
Utility mining
Summarization

ABSTRACT

Mining interesting itemsets from transaction databases has attracted a lot of research interests for decades. In recent years, high utility itemset (HUI) has emerged as a hot topic in this field. In real applications, the bottleneck of HUI mining is not at the efficiency but at the interpretability, due to the huge number of itemsets generated by the mining process. Because the downward closure property of itemsets no longer holds for HUIs, the compression or summarization methods for frequent itemsets are not available. With this in mind, considering coverage and diversity, we introduce a novel well-founded approach, called SUIT-miner, for succinctly summarizing HUIs with a small collection of itemsets. First, we define the condition under which an itemset can cover another itemset. Then, a greedy algorithm is presented to find the least itemsets to cover all of HUIs, in order to ensure diversity. For enhancing the efficiency, the greedy algorithm employs some pruning strategies. To evaluate the performance of SUIT-miner, we conduct extensive experiments on real datasets. The experimental results show that SUIT-miner is effective and efficient.

1. Introduction

Mining interesting itemsets from transaction databases has attracted a lot of research interest for decades. Frequent itemsets mining [1,6–8,11,25] is a fundamental research topic in data mining. However, the importance of different items is always not the same for users, which cannot be represented by frequency. Hence, frequent itemsets mining may discover itemsets which have a large amount of frequencies and low utility. To address this issue, utility itemsets mining [38] was proposed as an emerging topic in itemsets mining.

In utility itemsets mining, each item has a unit profit and can appear more than once in a transaction. Like frequent itemsets, itemsets with utilities not less than a user-specified minimum utility threshold are generally valuable and interesting, and they are called "high utility itemsets" (HUIs). High utility itemsets mining has a wide range of applications such as click stream analysis [2,13,27,30,39], cross-marketing in retail stores [9,16,22,31,32,37], mobile commerce environment planning [26] and biomedical applications [40]. Recent studies on HUI mining have seen significant performance improvements on efficiency, such us [3,17–21,29,41,42].

However, the major challenge of HUI mining is not at the efficiency but at the interpretability: unwieldy number of HUIs makes the patterns themselves difficult to explore, thus hampering the individual and global analysis of discovered patterns. If the threshold is not proper, HUI mining algorithm may output millions of itemsets or more, which cannot be useful for customer. The choice of the threshold also greatly influences the performance of the algorithms. A large number of high utility itemsets causes the mining algorithms to become inefficient or even run out of memory, because the more high utility itemsets the algorithms generate, the more resources they consume.

The same problem also occurs in frequent pattern mining. And as a result researchers have turned to various strategies to summarize the patterns the user is asked to examine. They solve this problem generally through two method: investigating the use of closed itemsets, maximal itemsets and non-derivable itemsets to make lossless compression [5,10,23,24,28,36] or summarizing itemset patterns using probabilistic models [12,33,35]. Nevertheless, because the *downward closure* property [1] of itemsets no longer holds for HUIs, such compression methods are not available for utility mining. And the iterative approaches based on probabilistic models are not practical in the days of big data, because they always need to scan the database many times or run generation process for model. So, an efficient and proper novel algorithm for mining the summarization of HUIs (SUITs) is in urgent need. To best of our knowledge, no such problem has been presented in the field of data mining.

We should at first make sure what kind of itemsets can be seen as the summarization of some HUIs in a transaction database. For

* Corresponding author. Tel.: +86 10 62755592.
E-mail address: zhdeng@cis.pku.edu.cn (Z.-H. Deng).

http://dx.doi.org/10.1016/j.knosys.2015.04.004
0950-7051/

exploring the solution to the problem, consider supermarket basket analysis. Beer and sausage is perfect match for many people. They can be enjoyed with some vegetable or staple food. So beer-sausage is always an important component or backbone of many HUIs in supermarket transaction databases, such us {beer, sausage, cheese} or {beer, sausage, bread}. And market strategies about this pair will get much profit with less work. If the prices of beer and sausage are reduced, bread and cheese will be sold more with the pair too. This kind of itemset is what we want: they are subsets of many HUIs and they have relative high utility. SUITs should be representative and own the most important information about the HUIs. SUITs also should be diversity and we will reach the aim of reducing result itemset number by them.

To mine the summarization of HUIs, we introduce a novel well-founded approach, called SUIT-miner, for succinctly summarizing HUIs with a small collection of itemsets. We at first define the condition under which an itemset can cover another itemset. After we have tried different definitions of "cover", we choose a brief and intuitive one. When itemset X is subset of itemset Y and the utility of X is higher than that of Y multiplied by a user-specified parameter λ, we say X cover Y. Based on the observation above, if an itemset can cover many other ones, it should be included by the summarization set. Then, in order to ensure diversity of the result, the algorithm finds the least itemsets to cover all of HUIs. It is an intractable task, because the number of candidates to be summarization and the number of HUIs waiting to be covered are both really great. To fulfill this task, we present a greedy algorithm incorporating some strategies for achieving high efficiency. To evaluate the performance of SUIT-miner with different pruning strategies, we conduct extensive experiments on real datasets. The experimental results show that SUIT-miner is efficient.

In this paper, we propose a novel approach to effectively and efficiently summarize high utility itemsets. Specifically, our contributions are as follows.

1. This paper is the first one suggests the problem of summarizing itemsets in utility database. The definition of summarization is intuitive and brief, so SUIT is available in many applications.
2. To solve the problem, we present a novel algorithm, called SUIT-miner, for mining summarization of HUIs. The algorithm is a greedy approach based on many effective strategies. The experiments show our algorithm is efficient and can be used in big databases.
3. The results of our experiments show the summarization of HUIs have many valuable features. The number of representative itemsets in the summarization is fairly small than HUIs, the summary itemsets have shorter length and they are diversity which we cannot get using utility as the only measure. So, it is convenience for user to scan the results and plan strategies.

The remainder of this paper is organized as follows. In Section 2, we introduce the background for utility mining and the statement of problem we suggest. The related works are introduced in Section 3. Sections 4 and 5 presents the proposed methods. The implementation of our algorithms is shown in Section 6. Experiments are shown in Section 7 and conclusion is given in Section 8.

2. Background

2.1. Utility mining

This section introduces the preliminaries related to utility mining, and then defines the problem statement of SUITs mining. We adopt the notations used in [20]. For more details about mining high utility itemsets, readers can refer to [20].

Let $I = \{i_1, i_2, i_3, \ldots, i_n\}$ be a set of items and DB be a database composed of a utility table and a transaction table. Each item in I has a utility value in the utility table. Each transaction T in the transaction table has a unique identifier (tid) and is a subset of I, in which each item is associated with a count value. An itemset is a subset of I and is called a k-itemset if it contains k items.

Definition 1. The external utility of item i, denoted as $\mathrm{eu}(i)$, is the utility value of i in the utility table of DB.

Definition 2. The internal utility of item i in transaction T, denoted as $\mathrm{iu}(i, T)$, is the count value associated with i in T in the transaction table of DB.

Definition 3. The utility of item i in transaction T, denoted as $\mathrm{u}(i, T)$, is the product of $\mathrm{iu}(i, T)$ and $\mathrm{eu}(i)$, where $\mathrm{u}(i, T) = \mathrm{iu}(i, T) \times \mathrm{eu}(i)$.

Definition 4. The utility of itemset X in transaction T, denoted as $\mathrm{u}(X, T)$, is the sum of the utilities of all the items in X in T in which X is contained, where $\mathrm{u}(X, T) = \Sigma_{i \in X \wedge X \subseteq T}\ \mathrm{u}(i, T)$.

Definition 5. The utility of itemset X, denoted as $\mathrm{u}(X)$, is the sum of the utilities of X in all the transactions containing X in DB, where $\mathrm{u}(X) = \Sigma_{T \in DB \wedge X \subseteq T}\ \mathrm{u}(X, T)$.

For better understanding of the above concepts, let's examine an example database shown by Table 1. For instance, u({beer, sausage}, T1) = u(beer, T1) + u(sausage, T1) = 4 × 5 + 3 × 6 = 38, and u({beer, sausage}) = u({beer, sausage}, T1) + u({beer, sausage}, T4) + u({beer, sausage}, T5) = 38 + 20 + 24 = 82.

An itemset X is an HUI if $\mathrm{u}(X)$ is not less than a user-specified minimum utility threshold denoted as *minutil*, or the product of a *minutil* and the total utility of a mined database if the *minutil* is a percentage. Given a database and a *minutil*, the high utility itemsets mining problem is to discover from the database all the itemsets whose utilities are not less than the *minutil*.

It should be noticed that the downward closure property of itemsets no longer holds for high utility itemsets (HUIs). For example, assume a transaction database has only one transaction, {a, 1; b, 1}, where eu(a) = 2 and eu(b) = 3. And if minutil is 3, then for u({a}) = 2, u({b}) = 3 and u({a, b}) = 5, {b} and {a, b} are high utility itemsets and {a} is not. That indicates that the downward closure property becomes invalid for high utility itemsets, which makes high utility itemset mining is much more challenging than frequent itemset mininig.

2.2. Problem statement

Now, we discuss the definition of condition under which an itemset can cover another itemset. At first, if itemset Y covers

Table 1
The example database.

Item	beer	sausage	bread	cheese	egg	milk	greens
(a) Utility table							
Utility	4	6	5	10	2	3	3
Tid	Transaction					Count	
(b) Transaction table							
T1	{beer, sausage}					{5, 3}	
T2	{bread, cheese, egg, greens, milk}					{4, 1, 2, 1, 2}	
T3	{bread, milk}					{3, 3}	
T4	{beer, sausage, greens, cheese}					{2, 2, 1, 2}	
T5	{beer, sausage, bread, cheese}					{3, 2, 4, 1}	
T6	{cheese, greens}					{1, 2}	
T7	{sausage, bread, greens}					{2, 2, 2}	