A MINING STRATEGY FOR MULTIPLE HIGH UTILITY ITEMSET

Prof.S.Krishnamoorthi,Professor,Department of Computer Science and Engineering
Bharathiyar college of Engineering and Technology,Karaikal
Ms.R.Suganyaa,II year PG Student
Bharathiyar college of Engineering and Technology,Karaikal

ABSTRACT

High utility item sets (HUIs) mining is an emerging topic in data mining, which refers to discovering all itemsets having a utility meeting a user-specified minimum utility threshold min_util. Mining high utility itemsets from a transactional database refers to the discovery of itemsets with high utility like profits. The process of finding an appropriate High utility itemset based on threshold is a tedious process for users. If threshold is set too low, too many HUIs will be generated, which may cause the mining process to be very inefficient. On the other hand, if threshold is set too high, it is likely that no HUIs will be found. In this paper, the above issues overcome by proposing a new framework for Extended-top-k based on Frequent Itemset Mining(FIM) and High-Utility Itemset Mining(HUIM), where k is the desired number of HUIs to be mined. Two types of efficient algorithms named TKUI (mining Top-K Utility item sets) and TKOI (mining Top-K utility item sets in one phase) are proposed for mining such item sets without the need to set threshold.

Keywords— *Frequent itemset mining High utility itemsets, Top-K Utility item sets, Top-K utility item sets in One phase*

1. INTRODUCTION

During the last ten years, Data mining, also known as knowledge discovery in databases has established its position as a prominent and important research area. The goal of data mining is to extract higher-level hidden information from an abundance of raw data. Data mining has been used in various data domains. Data mining can be regarded as an algorithmic process that takes data as input and yields patterns, such as classification rules, itemsets, association rules, or summaries, as output. abundance of raw data. Data mining has been used in various data domains. Data mining can be regarded as an algorithmic process that takes data as input and yields patterns, such as classification rules, itemsets, association rules, or summaries, as output.

Utility Mining is an extension of Frequent Itemset mining, which discovers itemsets that occur frequently. In many real-life applications, high-utility itemsets consist of rare items. Rare itemsets provide useful information in different decision-making domains such as business transactions, medical, security, fraudulent transactions, retail communities. For example, in a supermarket, customers purchase microwave ovens or frying pans rarely as compared to bread, washing powder, soap. But the former transactions yield more profit for the supermarket. Similarly, the high-profit rare itemsets are found to be very useful in many application areas. For example, in medical application, the rare combination of symptoms can provide useful insights for doctors .A retail business may be interested in identifying its most valuable customers i.e. who contribute a major fraction of overall company profit. Several researches about itemset utility mining were proposed.

In this paper, we address all of the above challenges by proposing a novel framework for Extended-top-k based on Frequent Itemset Mining and high-utility itemset mining, where k is the desired number of HUIs to be mined. Major contributions of this work are summarized as follows:

First, two efficient algorithms named TKUI (mining Top-K Utility item sets) and TKOI (mining Top-K utility item sets in One phase) are proposed for mining the complete set of top-k HUIs in databases without the need to specify the min_util threshold.

The TKUI algorithm adopts a compact tree-based structure named UP-Tree to maintain the information of transactions and utilities of item sets. TKUI inherits useful properties from the TWU model and consists of two phases.

2. SCOPE OF THE PROJECT

Now a day, in any real application, the size of the data set easily goes to hundreds of Mbytes or Gbytes. One of the most challenging data mining tasks is the mining of high utility itemsets efficiently. Identification of the itemsets with high utilities is called as Utility Mining. Frequent itemsets are the itemsets that occur frequently in the transaction data set. The utility can be measured in terms of cost, profit or other expressions of user preferences. In order to tackle this challenge, we propose Extended Top-K algorithm to efficiently mine high utility itemsets. Frequent Itemset Mining is to identify all the frequent itemsets in a transaction dataset. For example, a computer system may be more profitable than a telephone in terms of profit.

The utility of an itemset is defined as the external utility multiplied by the internal utility. An itemset is called a high utility itemset, if its utility is no less than a user- specified threshold; otherwise, the itemset is called a low utility itemset. Mining high utility itemsets from databases is an important task which is essential to a wide range of applications such as website click streaming analysis, cross-marketing in retail stores, business promotion in chain hypermarkets and even biomedical applications.

3. LITERATURE SURVEY

3.1 Fast Algorithms for Mining Association Rules

We consider the problem of discovering association rules between items in a large database of sales transactions. We present two new algorithms for solving this problem that are fundamentally die rent from the known algorithms. Empirical evaluation shows that these algorithms outperform the known algorithms by factors ranging from three for small problems to more than an order of magnitude for large problems. We also show how the best features of the two proposed algorithms can be combined into a hybrid algorithm, called AprioriHybrid. Scale-up experiments show that Apriori Hybrid scales linearly with the number of transactions. AprioriHybrid also has excellent scale-up properties with respect to the transaction size and the number of items in the database.

3.2 Efficient tree structures for high-utility pattern mining in incremental databases

High utility pattern (HUP) mining is one of the most important research issues in data mining due to its ability to consider the non binary frequency values of items in transactions and different profit values for every item. On the other hand, incremental and interactive data mining provide the ability to use previous data structures and mining results in order to reduce unnecessary calculations when a database is updated, or when the minimum threshold is changed. In this paper, we propose three novel tree structures to efficiently perform incremental and interactive HUP mining. The first tree structure, Incremental HUP Lexicographic Tree ({\rm IHUP}_{{\rm {L}}}-Tree), is arranged according to an item's lexicographic order. It can capture the incremental data without any restructuring operation. The second tree structure is the IHUP Transaction Frequency Tree ({\rm IHUP} _{{\rm {TF}}}-Tree), which obtains a compact size by arranging items according to their transaction frequency (descending order). To reduce the mining time, the third tree, IHUP-Transaction-Weighted Utilization Tree ({\rm IHUP} _ {{\rm {TWU}}}-Tree) is designed based on the TWU value of items in descending order. Extensive performance analyses show that our tree structures are very efficient and scalable for incremental and interactive HUP mining.

3.3 Mining high-utility itemsets

Traditional association rule mining algorithms only generate a large number of highly frequent rules, but these rules do not provide useful answers for what the high utility rules are. In this work, we develop a novel idea of top-K objective-directed data mining, which focuses on mining the top-K high utility closed patterns that directly support a given business objective. To association mining, we add the concept of utility to capture highly desirable statistical patterns and present a level-wise item-set mining algorithm. With both positive and negative utilities, the anti-monotone pruning strategy in Apriori algorithm no longer holds. In response, we develop a new pruning strategy based on utilities that allow pruning of low utility itemsets to be done by means of a weaker but anti-monotonic condition. Our experimental results show that our algorithm does not require a user specified minimum utility and hence is effective in practice.

3.4 Mining Frequent Patterns without Candidate Generation

Mining frequent patterns in transaction databases, time-series databases, and many other kinds of databases has been studied popularly in data mining research. Most of the previous studies adopt an Apriori-like candidate set generation-and-test approach. However, candidate set generation is still costly, especially when there exist a large number of patterns and/or long patterns. In this study, we propose a novel frequent-pattern tree (FP-tree) structure, which is an extended prefix-tree structure for storing compressed, crucial information about frequent patterns, and develop an efficient FP-tree, based mining method, FP-growth, for mining the complete set of frequent patterns by pattern fragment growth.

Efficiency of mining is achieved with three techniques: (1) a large database is compressed into a condensed, smaller data structure, FP-tree which avoids costly, repeated database scans, (2) our FP-tree-based mining adopts a pattern-fragment growth method to avoid the costly generation of a large number of candidate sets, and (3) a partitioning-based, divide-and-conquer method is used to decompose the mining task into a set of smaller tasks for mining confined patterns in conditional databases, which dramatically reduces the search space. Our performance study shows that the FP-growth method is efficient and scalable for mining both long and short frequent patterns, and is about an order of magnitude faster than the Apriori algorithm and also faster than some recently reported new frequent-pattern mining methods.

4. PROPOSED WORK

To tackle this problem, the concept of discover the itemsets with the highest utilities without setting the thresholds, a promising solution is to redefine the task of mining HUIs as mining top-k high utility itemsets (top-k HUIs) and FIM. The idea is to let the users specify k, i.e., the number of desired itemsets with frequently accessed, instead of specifying the minimum utility threshold. The concept of transaction-weighted utilization (TWU) model was introduced to facilitate the performance of the mining task.

4.1 Top-k High Utility Pattern Mining

The task of top-k high utility pattern mining was introduced by Chan et al. [4]. But the definition of high utility itemset used in their study is different from the one used in this work. Chan et al.'s study has considered utilities of various items, but quantitative values of items in transac-tions were not taken into consideration. In [6], we have defined the task of top-k high utility itemset mining by considering both quantities and profits of items. This work has inspired a few studies for mining top-k high utility patterns. Zihayat and An [9] have proposed an efficient algorithm T-HUDS for mining top-k HUIs over data streams. Yin et al. [8] have proposed a new framework for mining top-k high utility sequential patterns. Recently, Ryang and Yun extended [10] to propose the REPT algorithm [11] with four strategies PUD, RIU, RSD and SEP for top-k HUI mining. In REPT, besides the parameter k, users need to set another parameter N to control the effective-ness of RSD [11]. However, it is not easy for users to choose an appropriate N value and the choice of N greatly influences the performance of REPT.

4.2 The TKOI Algorithm

The second algorithm that we propose is TKOI (mining Top-k utility itemsets in one phase). It can discover top-k HUIs in only one phase. It utilizes the basic search procedure of HUI-Miner and its utility-list structure. Whenever an itemset is generated by TKOI, its utility is calculated by its utility-list without scanning the original database. We first describe a basic version of TKOI named TKOI Base and then the advanced version, which includes several strategies to increase its efficiency.

4.3 UP-Tree Structure

Then, we briefly introduce the UP-Tree structure. For more details about it, readers are referred to [12]. Each node N of a UP-Tree have five entries: N.name is the item name of N; .count is the support count of N; N.nu is the node utility of N; N.parent indicates the parent node of N; N.hlink is a node link which may point to a node having the same item name as N.name. The Header table is a structure employed to facilitate the traversal of the UP-Tree.

High-utility item sets can be generated from UP-Tree efficiently with only two scans of original databases. It proposed for facilitating the mining process of UP-Growth+ by maintaining only essential information in UP-Tree. The transactions are inserted in the UP-Tree. Then, Top-k High Utility Pattern Mining are used in the architecture, High utility item set used in their study is different from the one used in this work. We have defined the task of top-k high utility item set mining by considering both quantities and profits of items.

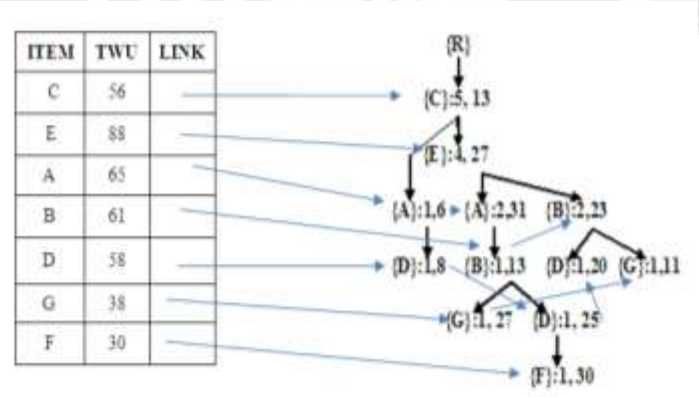
Example: Let consider Table 1 be an example database containing five transactions. Each row in Table 1 represents a transaction, where each letter represents an item and has a purchase quantity (i.e., internal utility). The unit profit (i.e., external utility) of each item is shown in Table 2. If *abs_min_util* = 30, the complete set of HUIs in Table 1 is {{BD}:30, {ACE}:31, {BCD}:34, {BCE}:31, {BDE}:36, {BCDE}:40, {ABCDEF}:30}, where the number beside each itemset is its absolute utility.

Table 1: Representation of Transaction

TID	TRANSACTION	TRANSACTION UTILITY(TU)
T1	(A,1)(C,1)(D,1)	8
T2	(A,2)(C,6)(E,2)(G,5)	27
T3	(A,1)(B,2)(C,1)(D,6)(E,1)(F,5)	30
T4	(B,4)(C,3)(D,3)(E,1)	20
T5	(B,2)(C,2)(E,1)(G,2)	11

Table 2: Unit profit

ITEM	A	B	C	D	E	F	G
UNIT PROFIT	5	2	1	2	3	1	1



ITEM	TWU	LINK
C	56	
E	88	
A	65	
B	61	
D	58	
G	38	
F	30	

Fig-1: A UP-Tree After inserting all the transaction in Table 1

4.4 The TKUI Algorithm

In this subsection, we propose four strategies to effectively raise min_utilBorder during different stages of the mining process. The four strategies are incorporated in TKUI Base to form the advanced TKUI algorithm.

4.4.1 Pre-evaluation Step

Though TKUI Base provides a way to mine top-k HUIs, min_util Border is set to 0 before the construction of the UP-Tree. This results in the construction of a full UP-Tree in memory, which degrades the performance of the mining task. If min_util Border could be raised before the construction of the UP-Tree and prune more unpromising items [12] in transactions, the number of nodes maintained in memory could be reduced and the mining algorithm could achieve better performance. Based on this idea, we propose a strategy named PE (Pre-evaluation Step) to raise min_utilBorder during the first scan of the database.

```
ALGORITHM: TKUBase
Input: (1) A database D; (2) The number of desired HUIs k;
Output: (1) The complete set of PKHUIs C;
Set min_utilBorder ← 0; TopK-MIU-List ← Ø ;
C← Ø ;
Construct a UP-Tree by scanning D twice;
//Apply a UP-Growth search procedure to generate PKHUIs;
For each PKHUI generated with estimated utility ESTU(X) do
{
If (ESTU(X) ≥ min_utilBorder and MAU(X) ≥ min_utilBorder)
{
Output X and min
{
ESTU(X), MAU(X); C ← C □ X; If(MIU(X) ≥ min_utilBorder )
{
//Raise min_utilBorder by the strategy MC;
min_utilBorder ← MC(MIU(X), TopK-MIU-Lit); }
}
}
```

4.4.2 Raising the Threshold by Node Utilities

We also propose a strategy called NU (raising the threshold by Node Utilities), which is applied during the construction of the UP-Tree.

4.4.3 Raising the threshold by MIU values of Descendents

The third strategy that we propose is called MD (raising the threshold by MIU values of Descendents). It is applied after the construction of the UP-Tree and before the generation of PKHUIs.

4.4.4 Raising the Threshold during Phase II

The fourth proposed strategy is called SE (raising the threshold by Sorting and calculating Exact utility of candidates), which is applied during the phase II of TKUI.

5. CONCLUSION

In this paper, we have studied the problem of top-k high utility item sets mining, where k is the desired number of high utility item sets to be mined. Two efficient algorithms TKUI (mining Top-K Utility item sets) and TKOI (mining Top-K utility item sets in One phase) are proposed for mining such item sets without setting minimum utility thresholds. TKUI is the first two-phase algorithm for mining top-k high utility item sets, which incorporates five strategies PE, NU, MD, MC and SE to effectively raise the border minimum utility thresholds and further prune the search space. On the other hand, TKOI is the first one-phase algorithm developed for top-k HUI mining, which integrates the novel strategies RUC, RUZ and EPB to greatly improve its performance. Empirical evaluations on different types of real and synthetic datasets show that the proposed algorithms have good scalability on large datasets and the performance of the proposed algorithms is close to the optimal case of the state-of-the-art two-phase and one-phase utility mining algorithms. Although we have proposed a new framework for Extended top-k HUI mining and FIM, it has not yet been incorporated with other utility mining tasks to discover different types of top-k high utility patterns such as top-k high utility episodes, top-k closed+ high utility item sets, top-k high utility web access pat-terns and top-k mobile high utility sequential patterns. These leave wide rooms for exploration as future work.

6. REFERENCES

[1] R. Agrawal and R. Srikant, ―Fast Algorithms for Mining Association Rules,‖ in Proc. of Int'l Conf. on Very Large Data Bases, pp. 487-499, 1994.

[2] C. Ahmed, S. Tanbeer, B. Jeong and Y. Lee, ―Efficient Tree Structures for High-utility Pattern Mining in Incremental Databases,‖ IEEE Transactions on Knowledge and Data Engineering, Vol. 21(12), pp. 1708-1721, 2009.

[3] K. Chuang, J. Huang and M. Chen, ―Mining Top-K Frequent Patterns in the Pres-ence of the Memory Constraint,‖ The VLDB Journal, Vol. 17, pp. 1321-1344, 2008.

[4] R. Chan, Q. Yang and Y. Shen, ―Mining High-utility Itemsets,‖ in Proc. of IEEE Int'l Conf. on Data Mining, pp. 19-26, 2003.

[5] P. Fournier-Viger, V. S Tseng, ―Mining Top-K Sequential Rules,‖ in Proc. of Int’l Conf. on Advanced Data Mining and Applications, pp. 180-194, 2011.

[6] P. Fournier-Viger, C. Wu, V. S. Tseng, ―Mining Top-K Association Rules,‖ in Proc. of Int’l Conf. on Canadian conference on Advances in Artificial Intelligence, pp. 61–73, 2012.

[7] P. Fournier-Viger, C. Wu, V. S. Tseng, ―Novel Concise Representations of High Utility Itemsets Using Generator Patterns," in Proc. of Int’l. Conf. on Advanced Data Mining and Applications and Lecture Notes in Computer Science, Vol. 8933, pp. 30-43, 2014.

[8] Yin, Z. Zheng, L. Cao, Y. Song and W. Wei, ―Mining Top-K High Utility Sequen-tial Patterns,‖ in Proc. of IEEE Int'l Conf. on Data Mining, pp. 1259-1264, 2013.

[9] M. Zihayat and A. An, ―Mining Top-K High Utility Itemsets over Data Streams,‖ Information Sciences, Vol. 285 (20), pp. 138–161, 2014.

[10] C. Wu, B. Shie, V. S. Tseng and P. S. Yu, ―Mining Top-K High Utility Itemsets,‖ in Proc. of the ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, pp. 78–86, 2012.

[11]H. Ryang and U. Yun, ―Top-K High Utility Pattern Mining with Effective Thre-shold Raising Strategies,‖ Knowledge-Based Systems, Vol. 76, pp. 109-126, 2015

[12]V. S. Tseng, C. Wu, B. Shie, and P. S. Yu, ―UP-Growth: An Efficient Algorithm for High Utility Itemset Mining,‖ in Proc. of the ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, pp. 253–262, 2010.