Pre-evaluation Strategy on Algorithms for Mining Top – k High Utility Item Sets

M. V. Mali
Department of Computer Science and Engineering, V.V.P. Institute of Engineering and Technology,
Solapur, Maharashtra, India -413004

H. B. Torvi
Assistant Professor, Department of Computer Science and Engineering, V.V.P. Institute of Engineering and Technology, Solapur, Maharashtra, India -413004

ABSTRACT

A rising trend in data mining is a High utility item sets (HUIs) mining. It aims to find all item sets which have an utility which meets a client determined least utility edge min_util. But , for clients, it is an issue to set a min_util efficiently. So, it is not proper procedure for clients to find a least utility edge by experimentation. An excessive number of HUIs will be produced, in the case that min_util is set very low. Due to this the mining procedure may result wasteful. It is also possible that no HUIs be found, if min_util is set very high. So for addressing the above issues, we redefine the problem of high utility item sets (HUIs) mining by top-k high utility item sets (top-k HUI) mining. Here, desired number of HUIs to be mined is k. Two different algorithms which are named as TKU and TKO (mining Top-K Utility item sets in two stages , mining Top-K utility item sets in one stage, respectively) are proposed for mining the item sets without setting the value of min_util. We apply pre-evaluation strategy to algorithms to improve the performance.

General Terms

Data mining ,High utility mining

Keywords

Utility mining, high utility item set mining, top-k high utility item set mining, frequent item set, transactional database.

1. INTRODUCTION

A fundamental research topic in data mining is Frequent item set mining (FIM). But, a large amount of frequent but low-value item sets are discovered in the traditional FIM and lose the information on valuable item sets which may have low selling frequencies. Hence, it can't satisfy the requirement of users who desire to discover item sets with high utilities such as high profits. An item set is said to be frequent if its support is no less than a given minimum support threshold. Hundreds of studies have been conducted on this topic. However, an important limitation of FIM is its assumptions that all items have the same importance to the user (e.g. unit profit or weight) and that items may not appear(quantity) more than once in each transaction.

These assumptions often do not hold in real life. For example, in transaction databases, items may have different unit profits, and items in transactions may be associated with different purchase quantities. Besides, in real-life applications, retailers may be more interested in finding item sets that yield a high profit rather than discovering frequent item sets.

Utility mining is nothing but the discovery of item sets with utilities higher than a user-specified minimum utility threshold. It has a wide range of applications including e-commerce. But a difficult problem here is to set an appropriate minimum utility threshold. Many (too much) high utility item sets will be generated if in case, the minimum threshold is set too low. Also it may take a long time to compute. On the other hand setting the minimum threshold too high may result in too few results. Setting appropriate minimum utility threshold by trial and error is not very efficient. To precisely control the output size and discover the item sets 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 item sets (top-k HUIs). The idea is to let the users specify k, i.e., the number of desired item sets, instead of specifying the minimum utility threshold. Setting k is more comfortable than setting the threshold because k represents the number of item sets that the users want to find.

An efficient algorithm named TKU (Top-K Utility item sets mining) and TKO (mining Top-K utility item sets in one stage) are proposed for mining such item sets without the need to set *min*_util.

2. LITERATURE REVIEW

In this section, studies related to high utility item set mining, top-k frequent pattern mining and top – k high utility pattern mining are briefly reviewed.

2.1 High Utility Itemset Mining

High utility item set mining[3] has received lots of attention in recent years. Also many efficient algorithms have been proposed. Some of them are HUI-Miner, Two-Phase, IIDS, IHUP, UPGrowth and d2HUP. These algorithms can be generally categorized into two types: two phase and one-phase algorithms.

The main characteristic of two-phase algorithms is that they consist of two phases.

Phase one generates a set of candidates that are potential high utility item sets. And the second phase calculates the exact utility of each candidate found in the first phase to identify high utility item sets.

Unlike this, the discovery of high utility item sets using only one phase and the production of no candidates are the main characteristics of one-phase algorithms. d2HUP and HUI-Miner are one-phase algorithms. Even if above studies may perform well in some applications, they are not developed for top-k high utility itemset mining. Also it suffers from the problem of setting appropriate thresholds.

2.2 Top-k Frequent Pattern Mining

Many studies have been proposed to mine different kinds of top-k patterns[8], such as top-k frequent item sets, top-k frequent closed item sets, top-k closed sequential patterns, top-k association rules, top-k sequential rules, top-k correlation patterns.

2.3 Top – k High Utility Pattern Mining

Chan et al. introduced the task of top-k high utility pattern mining[11] . But the definition of high utility itemset used in this project and in their study is different. Chan et al.'s study has considered utilities of various items, but they have not considered the quantitative values of items in transactions. In this project, we are defining the task of top-k high utility itemset mining by considering both quantities and profits of items.

2.4 Top – k High Utility Itemset Mining

We try to solve above issues by redefining the problem of high utility item sets (HUIs) mining by top-k high utility item sets (top-k HUI) mining[1]. Here k is the desired number of HUIs to be mined. For this , two algorithms are proposed for mining such item sets without the need to set min_util.

The TKU Algorithm:

The baseline approach TKU is an extension of UPGrowth , a tree-based algorithm for mining HUIs. TKU adopts the UP-Tree structure of UP-Growth to maintain the information of transactions and top-k HUIs. TKU is executed in three steps:

(1) Constructing the UP-Tree.

(2) Generating potential top-k high utility itemsets (PKHUIs) from the UP-Tree.

(3) Identifying top-k HUIs from the set of PKHUIs.

The TKO Algorithm:

The TKO (mining Top-k utility itemsets in One phase) 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 TKO, its utility is calculated by its utility-list without scanning the original database.

3. PROPOSED SYSTEM

In proposed system, for better performance of both algorithms TKU and TKO, we will apply Pre-evaluation strategy to them.

Pre-Evaluation Strategy of Min_Util :

Though TKU provides a way to mine top-k HUIs, min_util 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. The number of nodes maintained in memory could be reduced if min_util could be raised before the construction of the UP-Tree and more unpromising items in transactions sets could be pruned more unpromising items in transactions, and the mining algorithm could achieve better performance. Based on this idea, we propose a strategy named PE (Pre-evaluation Step) to raise min_util during the first scan of the database.

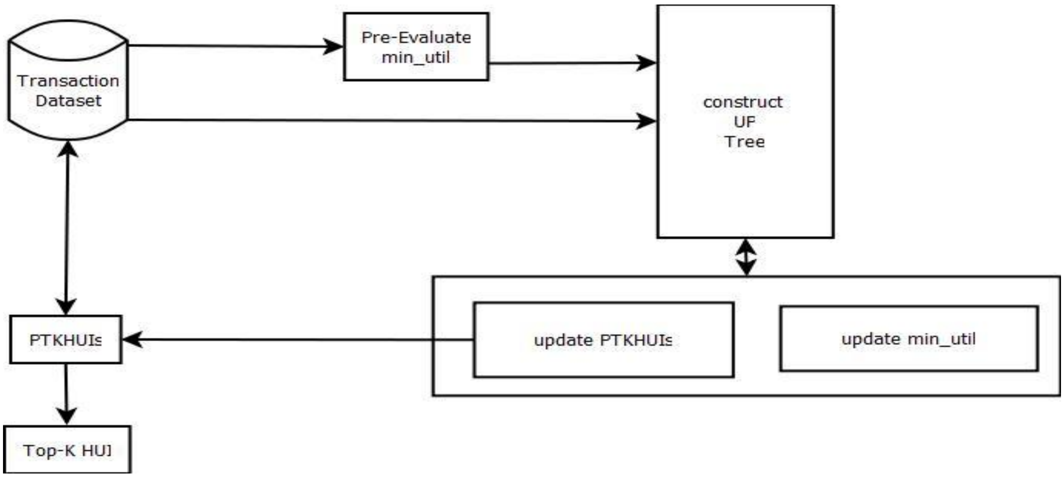

Fig 1: System design for TKU with Pre-Evaluation

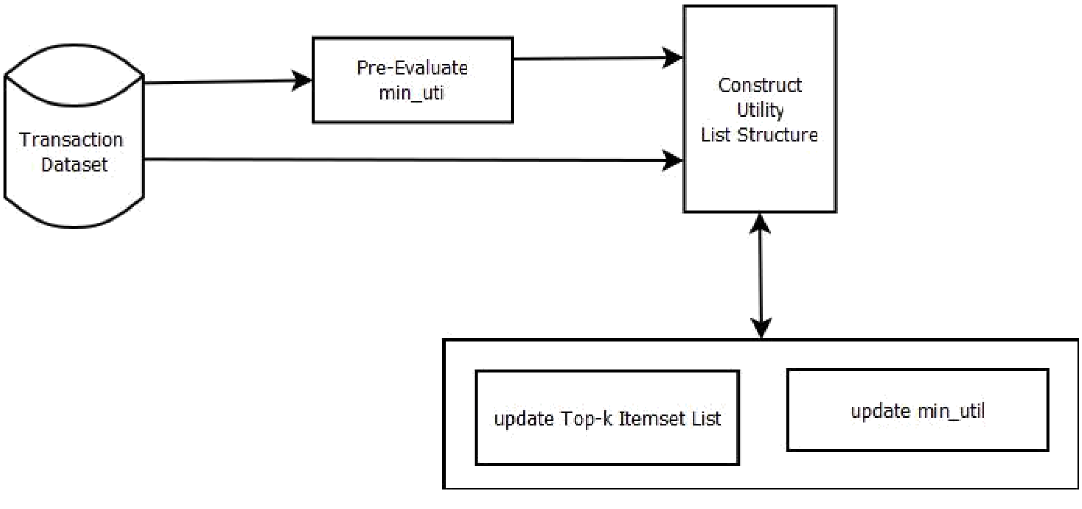

Fig 2: System design for TKO with Pre-Evaluation

4. METHODOLOGY

In the proposed system, in advance, we are calculating the min_util before the algorithms named TKU (mining Top-K Utility item sets in two stages) and TKO (mining Top-K utility item sets in one stage) starts working on transactions . Calculation of the min_util is done by the Pre-evaluation matrix method which considers the transactions, items and their utility. According to this strategy, min_util could be raised before the construction of the UP-Tree and prune more unpromising items in transactions, the number of nodes maintained in memory could be reduced and the mining algorithm could achieve better performance , while finding top k HUIs.

5. EXPERIMENTAL RESULTS

Following tables shows the time required for finding top-k HUIs by TKU and TKO with and without pre-evaluation of min_util. Two different datasets are used.

5.1 Supermarket Dataset

a)TKU

Table 1. Supermarket Dataset TKU

K	min_util Zero	min_util nonzero	% Improvement in Performance	Min_util
1	2356	1885	19.99	2103
10	2401	1777	25.98	2159
20	2525	1818	28.00	2203
30	2547	1885	25.99	2249
40	3015	2300	23.71	2305
50	3079	2433	20.98	2315
60	3120	2496	20.00	2326
70	3157	2337	25.97	2359
80	3162	2372	24.98	2378
90	3429	2504	26.97	2415

b)TKO

Table 2. Supermarket Dataset TKO

K	min_util Zero	min_util nonzero	% Improvement in Performance	Min_util
1	2215	1716	22.52	2103
10	2233	1635	26.78	2159
20	2324	1673	27.98	2203
30	2344	1772	24.40	2249
40	2800	2115	24.46	2305
50	2895	2300	20.55	2315
60	2871	2272	20.86	2326
70	2968	2127	28.33	2359
80	3004	2159	28.12	2378
90	3155	2304	26.97	2415

5.2 Stationary Shop Dataset

a)TKU

Table 3. Stationary Datasett TKU

K	min_util Zero	min_util nonzero	% Improvement in Performance	Min_util
1	2348	1855	20.99	1903
10	2401	1777	25.98	1959
20	2525	1818	28.00	2003
30	2547	1885	25.99	2015
40	3015	2300	23.71	2205
50	3079	2433	20.98	2295
60	3120	2496	20.00	2298
70	3157	2337	25.97	2359
80	3162	2372	24.98	2389
90	3429	2504	26.97	2412

b)TKO

Table 4. Stationary Datasett TKO

K	min_util Zero	min_util nonzero	% Improvement in Performance	Min_util
1	2195	1759	19.86	1903
10	2233	1635	26.78	1959
20	2324	1673	28.01	2003
30	2344	1772	24.40	2015
40	2800	2115	24.46	2205
50	2895	2300	20.55	2295
60	2871	2272	20.86	2298
70	2968	2127	28.33	2359
80	3004	2159	28.12	2389
90	3155	2304	26.97	2412

6. CONCLUSION

In the proposed system, an attempt is made to improve the performance of algorithms named TKU (mining Top-K Utility item sets in two stages) and TKO (mining Top-K utility item sets in one stage) by the use of pre-evaluation strategy. According to this strategy, min_util is calculated and raised before the construction of the UP-Tree and pruned more unpromising items in transactions. Also, the number of nodes maintained in memory is reduced and the mining algorithm achieves better performance. In future, for better performance, while calculating min_util, instead of considering all items, , the items with the less profit can be neglected.

7. REFERENCES

[1] V. Kavitha, B. G. Geetha, “Review on high utility itemset mining algorithms” IEEE Trans. Knowl. Data Eng., vol. 21, no. 12, pp. 1708–1721, October 2016.

[2] Song Wei, Liu Yu, Li Jinhong, "Mining high utility itemsets by dynamically pruning the tree structure", *Applied intelligence*, vol. 40, no. 1, pp. 29-43, 2014.

[3] Vincent S. Tseng et al., "Efficient Algorithms for Mining the Concise and Lossless Representation of High Utility Itemsets", *Knowledge and Data Engineering IEEE Transactions on*, vol. 27, no. 3, pp. 726-739, 2015.

[4] C. Ahmed, S. Tanbeer, B. Jeong, and Y. Lee, “Efficient tree structures for high-utility pattern mining in incremental databases,” IEEE Trans. Knowl. Data Eng., vol. 21, no. 12, pp. 1708–1721, Dec. 2009.

[5] K. Chuang, J. Huang, and M. Chen, “Mining top-k frequent patterns in the presence of the memory constraint,” VLDB J., vol. 17, pp. 1321–1344, 2008.

[6] R. Chan, Q. Yang, and Y. Shen, “Mining high-utility itemsets,” in Proc. IEEE Int. Conf. Data Mining, 2003, pp. 19–26.

[7] M. Liu and J. Qu, “Mining high utility itemsets without candidate generation,” in Proc.ACM Int. Conf. Inf. Knowl. Manag., 2012, pp. 55–64.

[8] P. Fournier-Viger, C.Wu, and V. S. Tseng, “Mining top-k association rules,” in Proc. Int. Conf. Can. Conf. Adv. Artif. Intell., 2012, pp. 61–73.

[9] P. Fournier-Viger, C. Wu, and V. S. Tseng, “Novel concise representations of high utility itemsets using generator patterns,” in Proc. Int. Conf. Adv. Data Mining Appl. Lecture Notes Comput. Sci., 2014, vol. 8933, pp. 30–43.

[10] J. Han, J. Pei, and Y. Yin, “Mining frequent patterns without candidate generation,” in Proc. ACM SIGMOD Int. Conf. Manag. Data, 2000, pp. 1–12.