S.N.V.R.Tapaswi* et al.
(IJITR) INTERNATIONAL JOURNAL OF INNOVATIVE TECHNOLOGY AND RESEARCH
Volume No.5, Issue No.3, April – May 2017, 6097-6100.

Mining Top-K High Utility Item Sets By Using Efficient Data Structure to Improve the Performance

S.N.V.R.TAPASWI
M.Tech Student
Gudlavalleru Engineering College, Gudlavalleru, India

Dr.G.V.S.N.R.V.PRASAD
Dean AA & Professor of CSE
Gudlavalleru Engineering College, Gudlavalleru, India

Abstract: **Association rules show strong relationship between attribute-value pairs (or *items*) that occur frequently in a given data set. Association rules are commonly used to determine the purchasing patterns of customers in a store. Such analysis is implemented in many decision-making processes, such as product placement, catalogue design, and cross-marketing. The discovery of association rules is based on frequent itemset mining. These frequent itemset mining algorithms mainly suffers from generation of more number of candidate itemsets and large no of database scans. These issues are addressed by two algorithms namely TKU (mining Top-K Utility itemsets) and TKO (mining Top-K utility itemsets in one phase) which are recommended for mining K- high utility itemsets in two scans of the entire database. Though scans are reduced to two, processing time is more because of UP-Tree traversals which is the data structure used by TKU and TKO algorithms. The proposed algorithm uses B+-Tree data structure instead of UP-Tree to reduce the time. Experimental analysis clearly shows that the processing time is improved and hence limitations of existing work are overcome by proposing a methodology using B+ -Tree.**

Keywords: **Utility Mining, High Utility Itemset Mining, Top-K Pattern Mining, Top-K High Utility Itemset Mining.**

I. INTRODUCTION

Association rules [2] [3] demonstrate solid relationship between attribute-value pairs (or items) that occur frequently in given information set. These rules are commonly used to analyze the purchasing behaviors of customers in a store. Mining of association rules is based on frequent itemset mining. The main problems in association rule mining is itemsets are not mined efficiently and they do not use the concept of utility. There are many frequent itemset mining algorithms [4], one of the simplest methods is Apriori algorithm. It is an iterative approach to mine the items which are frequent where it requires multiple passes over the data and itemsets that are derived are huge in number and vague. As more no of passes are required it is called as multiphase algorithm. So main drawback which arises here is requires more no of database scans. This results in inefficiency and high complexity of algorithms ,In order to overcome this problem a new framework that discuss about two algorithms TKU(mining Top-K Utility itemsets)and TKO(mining Top-k Utility in One Phase)[4][6][9][10][11].These two algorithms that make use of efficient data structure called UP-Tree to find high utility itemsets from transactional databases. Utility itemset[11] means each item is associated with a utility (e.g. unit profit) or an occurrence count in each transaction (e.g. quantity). The utility of an itemset[7][8] represents its importance, which can be measured in terms of weight, value, quantity or other information depending on the user specification. Making use of this concept, even if some products appear infrequently, they could be found when they have high utility. In the existing methodology the information of high utility itemsets is maintained in a special data structure named UP-Tree (Utility Pattern Tree)[5][11] from which the candidate itemsets can be generated efficiently with only two scans of the database and there by reduces query exhaustion but it takes more time because of more number of iterations. This paper proposes a new methodology which uses B+-Tree instead of UP-Tree that outshine prior approaches substantially with regards to execution time, specifically when databases contain lots of extended transactions.

II. BACKGROUND AND PROBLEM DEFINITION

This section defines the problem of top-k high utility itemset mining and then introduces related studies.

Problem Definition: Let be a finite set of distinct items $I^* = \{i_1, i_2, .. i_m\}$. A transactional database $D = \{T_1, T_2, .. T_n\}$ is a set of transactions, where each transaction $T_r \in D$ is a subset of I^* and has a unique identifier r, called T_{id}. Each item $I_j \in T_r$ has a positive value $Q(I_j, T_r)$, called its internal utility in T_r. Each item $I_j \in I^*$ is associated with a positive number $P(I_j, D)$, called its external utility. An itemset $X = \{I_1, I_2, . . ., I_L\}$ is a set of L distinct items, where $I_j \in I^*$ and L is the length of X. An L-itemset is an itemset of length L.

Problem Statement: Given a transactional database D and the desired number of HUIs k, the problem of top-k high utility itemsets mining is to discover all the itemsets having a utility no less than δ^* in D.Here δ is called border minimum utility threshold.

III. RELATED WORKS

This section introduces related works about top-k high utility itemset mining[7][8][10][11], including high utility itemset mining[9][10][11], top-k frequent pattern mining[2][3][4][5][6]. In recent years, high utility itemset mining has received lots of attention and many efficient algorithms were proposed. 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. In the first phase, they generate a set of candidates that are potential high utility itemsets. In the second phase, they calculate the exact utility of each candidate found in the first phase to identify high utility itemsets. IHUP, IIDS, UP- Growth are two-phase algorithms and d^2HUP and HUI-Miner are one phase algorithms. One the contrary, the main characteristic of one-phase algorithms is that they discover high utility itemsets using only one phase and produce no candidates. d^2HUP transforms a horizontal database into a tree-based structure called CAUL and adopts a pattern-growth strategy to directly discover high utility itemsets in databases. HUI-Miner considers a database of vertical format and transforms it into utility-lists. The utility-list structure used in HUI-Miner allows directly computing the utility of generated itemsets in main memory without scanning the original database. Although the above studies may perform well in some applications, they are not developed for top-k high utility itemset mining and still suffer from the subtle problem of setting appropriate thresholds. UP-Growth [9] is one of the state-of-the-art two-phase algorithms which incorporate four effective strategies DGU, DGN, DLU and DLN for pruning candidates in the first phase. This particular algorithm overcomes the problems of above stated works.

In existing work two algorithms TKU and TKO are used out of which TKU is the two phase algorithm which incorporates UP-Tree structure. The TKU concentrates on modifying and setting the min_util parameter within the itemset coupled with TKO method which tries to deduce user specified (k) rated itemsets configurations. They may need UP-tree [4] [8] [9] construction and maintenance approach to querying and reducing database scans to two. Four strategies are proposed for efficient construction of UP-Tree [9] [10] and the processing in UP-Growth. By these strategies, the estimated utilities of candidates can be well reduced by discarding the utilities of the items which are impossible to be high utility or not involved in the search space. The proposed strategies can not only efficiently decrease the estimated utilities of the potential high utility itemsets but also effectively reduce the number of candidates. The experimental results show that UP-Growth not only reduces the number of candidates effectively but also outperforms other algorithms substantially in terms of execution time, especially when the database contains lots of long transactions. TKU is executed in three steps: (1) constructing the UP-Tree, (2) generating potential high utility itemsets (PHUIs) from the UP-Tree, and (3) identifying top-k PKHUIs from the set of PHUIs. The second algorithm that we propose is TKO. It can discover top-k HUIs in only one phase. It utilizes the basic search procedure of HUI-Miner [10] 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. Though these algorithms reduce the database scans to two to traverse the UP-Tree node, but requires several no of iterations which increases the processing time. To optimize the solution a new algorithm named **$TKO_{Modified}$ algorithm using B+ -Tree** was proposed.

IV.EXISTING METHODOLOGY

In existing methodology to mine the high utility itemsets two algorithms are used namely TKU and TKO [9] [10] [11] algorithm. These algorithms uses compact specialized data structure called UP-Tree to mine the high utility itemsets that reduces the candidate generation and no of database scans. It also uses some strategies [11] to efficiently raise the min_utility threshold which is initially set to zero.

UP-Tree Structure:

Each node **N** in UP-Tree mainly consists of mainly 5 entries:

- ✓ N.name name of the item N.
- ✓ N.nu is the utility of node N.
- ✓ N.parent indicates parent node of N.
- ✓ N.hlink is a node that points to the node having same item name
- ✓ A header table that facilitates the UP-Tree Traversal.

The Fig1 shows the UP-Tree structure

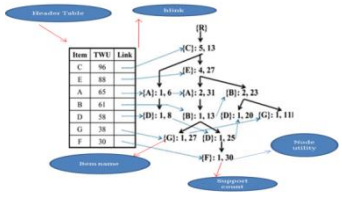

Fig1.UP-Tree Structure

Construction of UP-Tree:

In the first algorithm TKU incorporates a specialized data structure called UP-Tree. It is constructed by scanning the original database

twice. In the first scan, the transaction utility (TU) [11] of each transaction and Transaction Weighted utilization (TWU) of each item are computed. Subsequently, items are inserted into the header table in descending order of their TWUs [11]. During the second database scan, transactions are reorganized and then inserted into the UP- Tree.

The second algorithm which is implemented is TKO (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 TKO; its utility is calculated by its utility-list without scanning the original database.

By the above study of existing algorithms it is clearly states that UP-Tree structure improves the efficiency in terms of database scans as the scans are reduced to two. But the tree traversal requires more no of iterations which require more time. To optimize this, B+ tree data structure is used instead of UP-Tree. The new algorithm is designed named *$TKO_{Modified}$ algorithm using B+ -Tree* which is similar to TKU and TKO except that instead of UP-Tree this uses B+-Tree data structure.

V. PROPOSED METHODOLOGY

$TKO_{Modified}$ using B+-Tree is the one which uses B+-Tree data structure to store and retrieve the high utility itemsets. It is a tree of trees where it consists of internal and external nodes. As it is having proper structure the tree traversal is easy. Although the above algorithms helps to fasten up the process of rule mining by limiting the total number of database scans to two, it's complexity lies in the frequent UP-Tree traversal for obtaining potential high utility item sets. Tree Traversal of UP-tree structure requires many numbers of iterations. So this paper proposes a methodology which uses B-tree derivative known as B+- Tree.

Advantages of B+ trees:

- Because B+ trees don't have data associated with interior nodes, more keys can fit on a page of memory. Therefore, it will require fewer cache misses in order to access data that is on a leaf node.
- The leaf nodes of B+ trees are linked, so doing a full scan of all objects in a tree requires just one linear pass through all the leaf nodes.

  The fig2 show the structural difference between B+ trees and B trees.

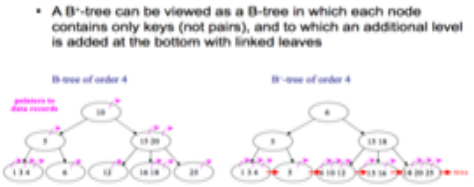

Fig2: Structure of B+-Tree

An experimental result on real and synthetic data sets confirms the performance increase in terms of time complexity of the above algorithms. By comparing the existing methodology with $TKO_{modified}$ using B+-Tree, B+-Tree is the best solution because the tree is having proper tree structure and requires less no of iterations to traverse the tree there by reducing the time complexity and improves the processing time.

VI.EXPERIMENTAL RESULTS

Here the performance is evaluated for both existing and proposed algorithms. Experiments are done on i_5 processor with 8GB RAM. The dataset used is MUSHROOM dataset.

Both UP-Tree and B+-Tree based classify the data based on the maximum no of occurrence count. UP-Tree based classification generate a statistical proof along with values and it requires only two scans.B+-Tree based classification provides the same output but the processing time is less when compared to UP-Tree because B+-tree has proper tree structure, which takes less no of iterations.

Table1: Processing time of the systems

System	*Processing time(sec)*
TKO using UP-Tree Based	3.14
$TKO_{Modified}$ using B+-Tree	0.88

Table1 shows that processing time of proposed TKO Modified using B+-Tree is less when compared to existing methodology.

VII. CONCLUSION

In this paper, we have studied the problem of top-k high utility itemsets mining, where k is the desired number of high utility itemsets to be mined. Two efficient algorithms TKU (mining Top-K Utility itemsets) and TKO (mining Top-K utility itemsets in One phase) are proposed for mining such itemsets without setting minimum utility thresholds. TKU is the first two-phase algorithm for mining top-k high utility itemsets, 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, TKO 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. Though the

scans are reduced by the above strategies but because of adoption of the UP-Tree in the above methodology they require more traversal time. To overcome a new data structure called B+-Tree is implemented using the same methodology and it was named as TKO $_{Modified}$ using B+-Tree.

As the processing time is reduced by using TKO $_{Modified}$ using B+-Tree when compared to existing so the proposed methodology is the best solution to derive the Top-K high utility itemsets and also best in terms of processing time

VIII. REFERENCES

[1] P. Fournier-Viger, C. Wu, and V. S. Tseng, "Novel concise representations of high utility itemsets using generator patterns," inProc. Int. Conf. Adv. Data Mining Appl. Lecture Notes Comput.

[2] 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.

[3] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in Proc. Int. Conf. Very Large Data Bases, 1994, pp. 487–499.

[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. 17081721, Dec.2009.

[5] P. Fournier-Viger and V. S. Tseng, "Mining top-k sequential rules," in Proc. Int. Conf. Adv. Data Mining Appl., 2011, pp. 180–194.

[6] 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.

[7] 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.

[8] R. Chan, Q. Yang, and Y. Shen, "Mining high-utility itemsets," inProc. IEEE Int. Conf. Data Mining, 2003, pp. 19–26.

[9] V. S. Tseng, C. Wu, B. Shie, and P. S. Yu, "UP-Growth: An efficient algorithm for high utility itemset mining," in Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2010, pp. 253–262.

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

[11] Efficient algorithms for mining Top-K high utility itemsets by Vincent S. Tseng, Senior Member, IEEE, Cheng-Wei Wu, Philippe Fournier-Viger, and Philip S. Yu, Fellow, IEEE

