

## International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijirset.com](http://www.ijirset.com)

Vol. 6, Issue 7, July 2017

# Improving Performance of the Mining Itemsets without Setting Minimum Utility Thresholds

N. Naveen Kumar<sup>1</sup>, Thirandasu Sandeep Kumar<sup>2</sup>

Assistant Professor, Department of CSE, School of Information Technology, JNTUH, Kukatpally, District Ranga  
Reddy, Telangana, India<sup>1</sup>

M.Tech Student, Department of Software Engineering, School of Information Technology, JNTUH, Kukatpally,  
District Ranga Reddy, Telangana, India<sup>2</sup>

**ABSTRACT:** The problem of frequent itemset mining is popular. But it has some important limitations when it comes to analyzing customer transactions. An important drawback is that purchase quantities are not taken into account. Thus, an item may only appear once or zero time in a transaction. A second major drawback is that all items are viewed as having the same importance, utility of weight. Thus, frequent pattern mining may find many frequent patterns that are not interesting. To address these limitations, the problem of frequent itemset mining has been redefined as the problem of **High-Utility Itemset (HUI) mining**. In this paper I propose top-k high utility itemset mining, where k is the desired number of HUIs to be mined. In this propose work, I am using two efficient algorithms to mine HUIs without set minimum utility threshold.

**KEYWORDS:** Utility mining, high utility itemset mining, top-k pattern mining, top-k high utility itemset mining

### I. INTRODUCTION

Data mining is that the method of finding patterns among security of fields in big relative databases. Several studies are done to HUI mining it's terribly tough for users to determine on a minimum utility threshold. In step with the worth of threshold, the output size is usually terribly little or terribly huge. Mining high utility itemsets from databases refers to finding the itemsets with high profits. High utility itemsets mining has become one among the foremost attention-grabbing data processing tasks with broad applications and it identifies itemsets whose utility satisfies a given threshold. An itemset is named a high utility itemset if its utility isn't but a user such that minimum utility threshold value; else that itemset is treated as an occasional utility itemset. Mining high utility itemsets from transactional databases is a crucial data processing task that refers to the invention of itemsets with high utilities (e.g. high profits). Though many studies are disbursed, current strategies might gift too several high utility itemsets for users, that degrades the performance of the mining task in terms of execution and memory potency. The choice of the edge greatly influences the performance of the algorithms. If the edge is just too low, too many HUIs are about to be generated and it's terribly tough for the users to grasp the results. An oversized type of HUIs to boot causes the mining algorithms to become inefficient. If the algorithms generate further HUIs, It uses further resources to boot. If the sting is regarding too high, no HUI are about to be generated; to look out worth for the minutil threshold, users got to try fully completely different values by estimate and re-executing the algorithms. This technique is extraordinarily time overwhelming. To limit the output size and to manage the item-sets with the most effective utilities while not setting the thresholds, a far higher resolution is to alter the task of mining HUIs as mining top-k high utility item-sets. Here the users specify k. Here k is that the amount of desired item-sets, rather than specifying the minimum utility threshold. Setting k is simpler than setting the brink as results of k is that the type of item-sets that the users would like to hunt out

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijirset.com](http://www.ijirset.com)

Vol. 6, Issue 7, July 2017

whereas selecting the edge depends on data characteristics that are unknown to users. Parameter  $k$  is employed instead of the  $\text{min\_util}$  threshold; it's terribly useful for many applications. This idea is employed to research consumer purchase behavior. High  $k$  HUI mining is employed to look out, what are the top- $k$  sets of merchandise that contribute the most effective profit to the company and also the thanks to expeditiously found these item-sets while not setting the  $\text{min\_util}$  threshold. Top- $k$  HUI mining is important to many applications, it's not an easy task for developing economical algorithms for mining such patterns. Two algorithms named TKU and technical knockout are projected for mining the entire set of high  $k$  HUIs in databases whereas not the need to specify the  $\text{minutil}$  threshold. The TKU rule uses a tree-based structure named UP-Tree; it's accustomed maintain the information of transactions and utilities of item-sets. TKU inherits useful properties from the TWU model and it consists of 2 phases. In phase I, potential top- $k$  high utility item-sets are generated. In clinical test, top- $k$  HUIs are far-famed from the set of PKHUIS generated in phase I. resulting rule is TKO; it uses a list-based structure named utility-list to store the utility information of item-sets among the data. It uses vertical data illustration techniques to look out top- $k$  HUIs in one section.

## II. RELATED WORK

Rakesh Agrawal, Ramakrishnan Srikant given 2 new algorithms, Apriori and AprioriTid, for locating all vital association rules between things in a very massive info of transactions. They compared these algorithms to the antecedently known algorithms, the AIS [AIS93b] and SETM [HS93] algorithms. They gave experimental results, victimization each artificial and real-life knowledge; showing that the planned algorithms continuously vanquish AIS and SETM. The performance gap magnified with the matter size, an ranged from an element of 3 for little issues to over an order of magnitude for giant issues. Vikram Goyal, Siddharth Dawar planned a unique arrangement, UP-Hist tree for locating high utility itemsets. The inclusion of bar graph reduced the calculable utility and helped in pruning the search area higher. Experimental results on real and artificial datasets demonstrate the higher performance of our planned arrangement compared to the progressive algorithms in terms of total execution time and range of candidates. K. Chuang, J. Huang, and M. subgenus Chen studied a much necessary mining downside, particularly mining top- $k$  frequent/closed itemsets within the presence of the memory constraint. to attain this, I have a tendency to planned the MTK/MTK\_Close algorithms, that square measure devised as level-wise search algorithms supported a good approach to constrain the amount of candidates which will be generated-and-tested in every info scan. Since the minimum support to retrieve top- $k$  frequent itemsets can't be known prior to, a unique search approach, referred to as the  $\delta$ -stair search, is devised in MTK and MTK\_Close to with efficiency retrieve top- $k$  frequent/closed itemsets. As incontestable within the empirical study on real knowledge and artificial knowledge, rather than solely providing the pliability of placing a compromise between the execution potency and also the memory consumption, MTK will each deliver the goods high potency and have a unnatural memory certain, showing its distinguished advantage to be a practicable rule of mining frequent patterns. C. Wu, P. Fournier-Viger, P. S. Yu, and V. S. Tseng self-addressed the matter of redundancy in high utility itemset mining by proposing a compact illustration of all high utility itemsets named closed+high utility itemsets. To our data, this can be the primary study on compact and lossless illustration of high utility itemsets. To mine this new kind of itemsets, I have a tendency to planned associate degree economical rule named CHUD. 3 effective methods named REG, RML and DCM were any planned to reinforce the performance of CHUD. To with efficiency recover all high utility itemsets from this illustration, I have a tendency to planned a top-down technique named DAHU. Real and artificial datasets with varied characteristics were wont to perform an intensive performance analysis. Existing ways of association rule mining think about the looks of associate degree item in a very dealing, whether or not or not it's purchased, as a binary variable. However, customers could obtain over one in all an equivalent item, and also the cost could vary among things. Utility mining, a generalized kind of the share mining model, tries to beat this downside. Since the Apriori pruning strategy cannot determine high utility item sets, developing associate degree economical rule is crucial for utility mining. This study proposes the Isolated things Discarding Strategy (IIDS), which may be applied to any existing level-wise utility mining technique to cut back candidates and to enhance performance. The foremost economical known models for share mining square measure SHFSM and DCG that conjointly works adequately for utility mining yet. By applying IIDS to SHFSM and DCG, the 2 ways FUM and DCG+ were enforced, severally. For each artificial and real datasets, experimental results reveal that

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijirset.com](http://www.ijirset.com)

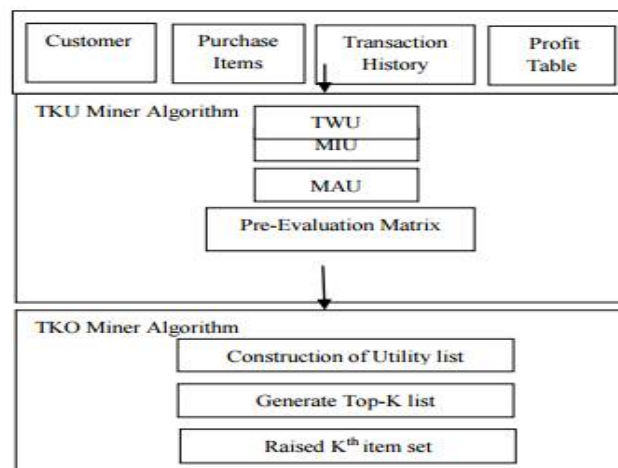
Vol. 6, Issue 7, July 2017

the performance of FUM and DCG+ is additional economical than that of SHFSM and DCG, severally. Therefore, IIDS is a good strategy for utility mining.

## III. FRAME WORK

### A. System Architecture

In this paper I tend to propose 2 economical algorithms like TKU (mining Top-K Utility itemsets) and KO (mining Top-K utility itemsets in one phase) algorithms. The TKU algorithmic program adopts a compact tree-based structure named UP-Tree to keep up the knowledge of transactions and utilities of itemsets. TKU inherits helpful properties from the TWU model and consists of 2 phases. In phase I, potential top-k high utility itemsets (PKHUIs) are generated. In clinical test, top-k HUIs are known from the set of PKHUIs discovered in phase I.



**Figure 1: Proposed System Architecture**

On the opposite hand, the KO algorithmic program uses a list-based structure named utility-list to store the utility info of itemsets within the information. It uses vertical information illustration techniques to search out top-k HUIs in mere one section.

The construct of group action weighted utilization (TWU) model was introduced to facilitate the performance of the mining task. Throughout this model, AN item set is termed high transaction-weighted utilization item set (HTWUI) if its TWU is not any but min\_util, where the TWU of AN item set represents A boundary on its utility. Therefore, a HUI ought to be a HTWUI and every one the HUIs should be enclosed among the entire set of HTWUIs. A classical TWU model-based algorithmic program consists of 2 phases. Among the initial half, cited as phase I, the entire set of HTWUIs is found. Among the second half, cited as clinical test, all HUIs are obtained by shrewd the precise utilities of HTWUIs with one information scan.

### Benefits of planned System:

1. 2 algorithms named TKU (mining Top-K Utility item-sets) and KO (mining Top-K utility item sets in one phase) are planned for mining the complete set of top-k HUIs in databases while not the requirement to specify the min-util threshold.
2. The event of the UP-Tree and prune extra inauspicious things in transactions, the quantity of nodes maintained in memory may well be reduced and thus the mining algorithmic program would possibly deliver the goods higher performance.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijirset.com](http://www.ijirset.com)

Vol. 6, Issue 7, July 2017

## B. The TKU algorithm

Here, projected AN algorithmic program named TKU for locating top-k HUIs while not specifies min-util. The Baseline approach of TKU is and seize of UP-Growth, a tree-based algorithmic program for mining HUIs. TKU uses the UP-Tree structure of UP-Growth to manage the knowledge of transactions and top-k HUIs. TKU is worked in 3 steps. (1) Constructing the UP-Tree, (2) generating potential top-k high utility item-sets from the UP-Tree, (3) distinguishing top-k HUIs from the set of PKHUIs, UP-Tree structure, UP-Tree structure is delineate in, every node N of a UP-Tree have 5 entries: N.name is that the item name of N; N count is that the support count of N; N.nu is that the node utility of N; N. parent indicates the parent node of N; N link may be a node link which can purpose to a node having a similar item name as N.name. The Header table may be a structure used to facilitate the traversal of the UP-Tree. A header table entry contains an item name, a calculable utility worth, and a link. The link points to the primary node within the UP-Tree having a similar item name because the entry. The node whose item names are a similar is traversed with efficiency by following the links in header table and therefore the node links within the UP-Tree.

## Construction of UP-Tree:

A UP-Tree is created by scanning the first information doubly. Among the initial scan, the group action utility of every group action and TWU of the each item are completed. Later on, things are inserted into the header table in descending order of their TW use. Throughout the second information scan, transactions are organized then inserted into the UP-Tree primarily; the tree is created with a supply.

When a group action is retrieved, things within the group action are sorted in descending order of TWU. A group action once the higher than reorganization is termed organized group action and its group action utility is termed organized group action utility; once inserting all the organized transactions, the development of the UP-Tree is completed.

## C. The TKO Algorithm

It will understand top-k HUIs in precisely one section. It uses the basic search procedure of HUI-miner and its utility-list structure. Whenever AN item set is generated by KO, utility of the generated item set is calculated by its utility-list while not scanning the initial information. Construction of Utility-list structure: Utility list is delineated in, among the case of KO algorithms, each item connected with a utility-list. The utility-list of things is usually cited as initial utility lists. These are created by scanning the information doubly. Within the 1st scan, the TWU and utility values of things are calculated. Throughout the second scan, things every dealing as TWU values and therefore the utility list of each item is built and UP tree are generate later on run the TKU and KO algorithmic program.

## IV. EXPERIMENTAL RESULTS

In our experiments, any wide variety of users can read the transactions from datasets that transaction loaded into the machine after that examine the take advantage of dataset the profit will be loaded into the device after that compute the each transaction after computing the each transaction then compute the TWU and also construct the UP tree TWU manner sum of all the TU values of the cutting-edge item in all of the transactions During the second test, gadgets in every transaction are looked after in order TWU values and the software list of each item is constructed and UP tree can be generate after that run the TKU and TKO set of rules primarily based on set of rules transaction values could be generate. In the under chart I can study that distinction among the length of TKU Algorithm and TKO Algorithm.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijirset.com](http://www.ijirset.com)

Vol. 6, Issue 7, July 2017



I can study that Minimum software threshold contrast chart among TKU and TKO algorithms chart TKU set of rules period is higher than TKO Algorithm duration. The distinction could be shown in the experience of Runtime. So I can consider that via the usage of this TKU and TKO set of rules to examine the item-sets without problems. Through our implementation I can improve the performance of the system at decrease value then examine to current methods.

## V. CONCLUSION

In this paper, I conclude that I solved the top-k high utility itemsets mining problem without setting min\_util thresholds. For that, I proposed an efficient two algorithms named as TKU (mining Top-K Utility itemsets) and TKO (mining Top-K utility itemsets in One phase). These two algorithms are used UP-Tree algorithm. From experimental results, I can say the proposed algorithms are efficient to mine the top-k high utility itemsets.

## REFERENCES

- [1] R. Chan, Q. Yang, and Y. Shen, "Mining high-utility item-sets," in Proc. IEEE Int. Conf. Data Mining, 2003, pp. 19–26.
- [2] P. Fournier-Viger and V. S. Tseng, "Mining top-k sequential rules," in Proc. Int. Conf. Adv. Data Mining Appl., 2011, pp. 180–194.
- [3] P. Fournier-Viger, C. Wu, and V. S. Tseng, "Mining top-k association rules," in Proc. Int. Conf. Can. Conf. Adv. Artif. Intel. 2012, pp. 61–73.
- [4] 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.
- [5] J. Han, J. Wang, Y. Lu, and P. Tzvetkov, "Mining top-k frequent closed patterns without minimum support," in Proc. IEEE Int. Conf. Data Mining, 2002, pp. 211–218.
- [6] S. Krishnamoorthy, "Pruning strategies for mining high utility item-sets," Expert Syst. Appl., vol. 42, no. 5, pp. 2371–2381, 2015.
- [7] C. Lin, T. Hong, G. Lan, J. Wong, and W. Lin, "Efficient updating of discovered high-utility item-sets for transaction deletion in Dynamic databases," Adv. Eng. Informat., vol. 29, no. 1, pp. 16–27, 2015.
- [8] G. Lan, T. Hong, V. S. Tseng, and S. Wang, "Applying the maximum utility measure in high utility sequential pattern mining," Expert Syst. Appl., vol. 41, no. 11, pp. 5071–5081, 2014.
- [9] Y. Liu, W. Liao, and A. Choudhary, "A fast high utility item-sets mining algorithm," in Proc. Utility-Based Data Mining Workshop, 2005, pp. 90–99.
- [10] [14] M. Liu and J. Qu, "Mining high utility item-sets without candidate generation," in Proc. ACM Int. Conf. Inf. Knowl. Manag., 2012, pp. 55–64.