INTERNATIONAL JOURNAL OF RESEARCH
In Advanced Engineering Technologies

Volume 6, Issue 1 JUN 2017

Mining High Transaction-Weighted Utilization Itemsets through TKU and TKO Algorithm

[1] N. NAVEEN KUMAR, [2] THIRANDASU SANDEEP KUMAR

[1]Assistant Professor, Department of CSE, School of Information Technology JNTUH, Village KPHB, Mandal Kukatpally, District RangaReddy, Telangana, India.

[2]M.Tech Student, Department of CSE, School of Information Technology JNTUH, Village KPHB, Mandal Kukatpally, District RangaReddy, Telangana, India.

Abstract— ***In this paper Mining high utility itemsets from data sets is an promising topic in data mining, that refers to the discovery of item-sets with utilities more than a user-specified minimum utility threshold min-util. though many studies are carried out on this subject, setting an applicable minimum utility threshold could be a difficult drawback for users. If minutil is about too low, too several high utility item-sets are generated, which can cause the mining algorithms to become inefficient or maybe run out of memory. On the opposite hand, if minutil is about too high, no high utility item-set are found. Setting applicable minimum utility thresholds by trial and error may be a tedious method for users. During this paper, I tend to address this drawback by proposing a brand new framework named top-k high utility item-set mining, wherever k is that the desired variety of high utility item-sets to be mined. an efficient algorithm named TKU (Top-K Utility item-sets mining) is planned for mining such item-sets without setting min-util. many options were designed in TKU to resolve the new challenges raised during this drawback, like the absence of anti-monotone property and also the requirement of lossless results. Moreover, TKU integrate many narrative methods for pruning the search space to achieve high efficiency. Results on real and synthetic datasets show that TKU has excellent performance and scalability.***

Index Terms -- ***Utility Mining, High Utility Item Set Mining, Top-K Pattern Mining, Top-K High Utility Item Set Mining, Data Mining, Frequent Item-set, Transactional Database;***

I. INTRODUCTION

Data mining additionally known as data discovery is that the method of analyzing information from completely different angles and summarizing it into helpful information, data mining may be a tool for analyzing information, It permits users to investigate data from different levels or angles, arrange it, and also the relationships among the information are found. Data mining is that the process of finding patterns among comfortable of fields in giant relative databases. Many studies are done to HUI mining it is very difficult for users to decide on a minimum utility threshold. According to the value of threshold, the output size is often very small or very massive. The selection of the threshold greatly influences the performance of the algorithms. If the threshold is too low, too several HUIs are going to be generated and it is very difficult for the users to understand the results. An outsized variety of HUIs additionally causes the mining algorithms to become inefficient. If the algorithms generate additional

HUIs, It uses additional resources additionally. If the edge is about too high, no HUI are going to be generated. To search out value for the minutil threshold, users need to attempt completely different values by guesswork and re-executing the algorithms. This method is extremely time consuming. To limit the output size and to manage the item-sets with the best utilities without setting the thresholds, a much better resolution is to change the task of mining HUIs as mining top-k high utility item-sets. Here the users specify k. Here k is that the number of desired item-sets, instead of specifying the minimum utility threshold. Setting k is easier than setting the brink as a result of k is the variety of item-sets that the users wish to seek out whereas choosing the threshold depends on information characteristics that are unknown to users. Parameter k is used rather than the minutil threshold; it is very helpful for several applications. This concept is used to analyze client purchase behavior. High k HUI mining is used to search out, what are the top-k sets of products that contribute the best profit to the corporate and the way to with efficiency found these item-sets without setting the minutil threshold. Top-k HUI mining is essential to several applications, it is not a simple task for developing efficient algorithms for mining such patterns .Two algorithms named TKU and TKO are projected for mining the complete set of high k HUIs in databases while not the requirement to specify the minutil threshold. The TKU algorithm uses a tree-based structure named UP-Tree; it is used to maintain the data of transactions and utilities of item-sets. TKU inherits helpful properties from the TWU model and it consists of two phases. In phase I, potential top-k high utility item-sets are generated. In phase II, top-k HUIs are known from the set of PKHUIS generated in phase I. consequent algorithm is TKO; it uses a list-based structure named utility-list to store the utility data of item-sets within the information. It uses vertical information representation techniques to search out top-k HUIs in one phase.

II. RELATED WORK

High Utility Item-set Mining High Utility Item-set Mining may be a popular concept and many algorithms are projected for HUI mining like two-phase, IHUP, IIDS, UP-Growth, and HUI-Miner. These algorithms will be generally classified in two types: Two-phase and one-phase algorithms. The characteristics of two-phase algorithm are that it consists of two phases. Within the initial section, they produce a group of candidates that are potential high utility item-sets. Within the second part, they calculate the precise utility of every candidate found within the initial part to identify high utility item-sets. Two-phase, IHUP, IIDS, and UP-Growth are two-phase based algorithms. 2. Top-k Pattern Mining several studies are planned to mine numerous varieties of top-k patterns, like top-k frequent item-sets, top-k frequent closed item-sets, top-k association rules, and top-k ordered rules. The selection of data structures and look for strategy affect the effectiveness of a top-k mining algorithm in terms of each memory and execution time. Apriori may be a illustrious algorithm used in data mining The Apriori algorithm is predicated on the construct that if a subset H appears N times, any other subset H' that contains H can appear N times or less. So, if H doesn't pass the minimum support threshold, neither will H'. There is no need to calculate H', it is discarded apriori. Now here progressing to show an example of this algorithm. Let's suppose here a client john with transactions [[pen, pencil, book], [pen, book, bag], [pen, bag], [pen, pencil, book]], and a minimum support threshold m of fifty percentage a pair of transactions. Initial step: Count the singletons apply threshold The singletons for john are: pen: 4, pencil: 2, book: 3, bag: a pair of all of the only things appear L or additional

times, therefore none of them are discarded. Second step: Generate pairs, count them and apply threshold. The pairs created were: pen, pencil, pen, book, pen, bag, pencil, book, pencil, bag, book, and bag. Currently I have a tendency to proceed to count them and applying the threshold. Pen, pencil: a pair of pen, book: three pen, bag: a pair of pencil, book: a pair of pencil, bag: 0 book, bag: one pencil, bag and book, bag have not passed the brink, so they are discarded and the other sub combination each of them will generate. The left over pairs are place during a temporary set. Ass = pen, pencil, pen, book, pen, bag, pencil, and book Step M: can generate triplets, quadruplets, etc., add up them, apply threshold and remove containing item-sets. I tend to generate triplets from our pairs. Triplets = pen, pencil, book, pen, pencil, bag, pen, book, bag, pencil, book, bag. currently I tend to count them: pen, pencil, book: a pair of pen, pencil, bag: 0 pen, book, bag: one pencil, book, bag: 0 only pen, pencil, book has passed the threshold, therefore currently I tend to proceed to feature it to Ass, but first, I have to remove the subsets that pen, pencil, book contains. Before adding our left over triplet Ass is appeared like this: pen pencil, pen, book, pen, bag, pencil, and book. Once I add the triplet and take away the subsets that are within it pen, pencil, pen, book and pencil, book are those that should go. Ass now appears like pen, pencil, book, pen, bag, and this can be the ultimate result. If I tend to hand over one triplet once apply the threshold, I tend to proceed to generating the quadruplets, enumeration them, applying the threshold, adding up them and removing the subsets that each quadruplet contains.

III. FRAME WORK

The concept of transaction weighted utilization (TWU) model was introduced to facilitate the performance of the mining task. During this model, an item set is termed high transaction-weighted utilization item set (HTWUI) if its TWU is no less than minutil, wherever the TWU of an item set represents an upper bound on its utility. Therefore, a HUI should be a HTWUI and all the HUIs must be included within the complete set of HTWUIs. A classical TWU model-based algorithm consists of two phases. Within the initial part, referred to as phase I, the complete set of HTWUIs is found. Within the second part, referred to as phase II, all HUIs are obtained by calculating the exact utilities of HTWUIs with one database scan. Benefits of planned System: one. Two economical algorithms named TKU (mining Top-K Utility item-sets) and TKO (mining Top-K utility item sets in one phase) are planned for mining the entire set of top-k HUIs in databases without the necessity to specify the minutil threshold. 2. The development of the UP-Tree and prune additional unpromising things in transactions, the amount of nodes maintained in memory could be reduced and therefore the mining algorithm might achieve higher performance. The TKU Algorithm: Here, projected an algorithm named TKU for finding top-k HUIs without specify min-util. The Baseline approach of TKU is and seize of UP-Growth, a tree-based algorithm for mining HUIs. TKU uses the UP-Tree structure of UP-Growth to manage the information of transactions and top-k HUIs. TKU is worked in three steps. (1) Constructing the UP-Tree, (2) generating potential top-k high utility item-sets from the UP-Tree, (3) identifying top-k HUIs from the set of PKHUIs, UP-Tree structure, UP-Tree structure is described in, Each node N of a UP-Tree have five entries: N.name is the item name of N; 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 link 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. A header table entry contains an item

name, an estimated utility value, and a link. The link points to the first node in the UP-Tree having the same item name as the entry. The nodes whose item names are the same can be traversed efficiently by following the links in header table and the node links in the UP-Tree. Construction of UP-Tree: A UP-Tree can be constructed by scanning the original database twice. Within the initial scan, the transaction utility of each transaction and TWU of the each item are completed. Subsequently, items are inserted into the header table in descending order of their TW use. During the second database scan, transactions are reorganized and then inserted into the UP-Tree primarily; the tree is produced with a source.

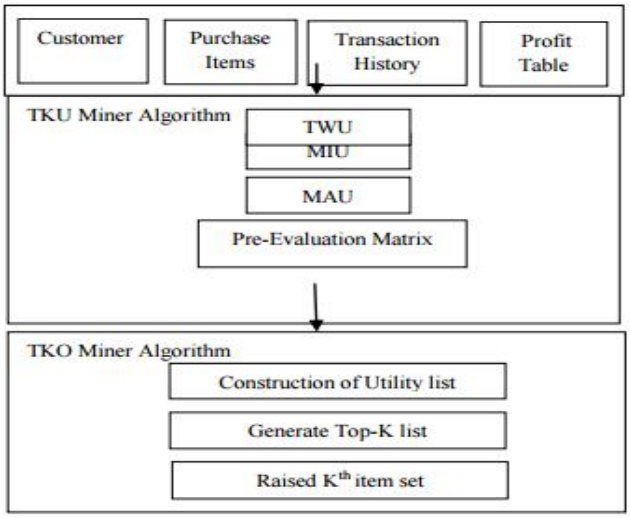
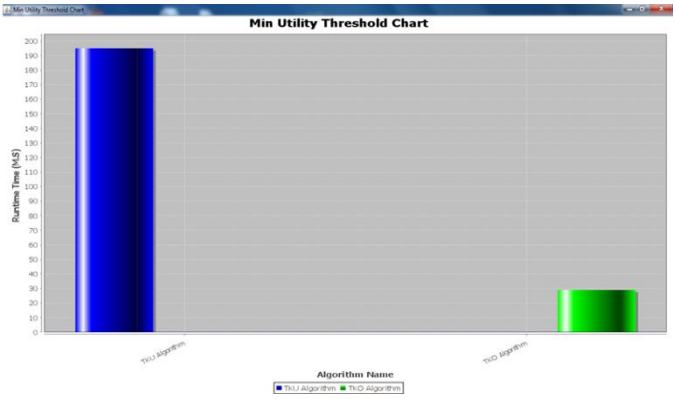


Figure 1: Proposed System Architecture

When a transaction is retrieved, items in the transaction are sorted in descending order of TWU. A transaction after the above reorganization is called reorganized transaction and its transaction utility is called reorganized transaction utility. After inserting all the reorganized transactions, the construction of the UP-Tree is completed. The TKO Algorithm: It can realize top-k HUIs in exactly one phase. It uses the fundamental search procedure of HUI-miner and its utility-list structure. Whenever an itemset is generated by TKO, utility of the generated itemset is calculated by its utility-list without scanning the initial database. Construction of Utility-list structure: Utility list is represented in, within the case of TKO algorithms, every item related with a utility-list. The utility-list of things is generally referred to as initial utility lists. These are created by scanning the database doubly. In the first scan, the TWU and utility values of items are calculated. Throughout the second scan, things each dealing are sorted so as TWU values and the utility list of every item is constructed and UP tree will be generate subsequently run the TKU and TKO algorithm.

IV. EXPERIMENTAL RESULTS

In our experiments, any number of users can read the transactions from datasets that transaction loaded into the system after that read the profit from dataset the profit will be loaded into the system after that compute the each transaction after computing the each transaction then compute the TWU and also build the UP tree TWU means sum of all the TU values of the current item in all the transactions During the second scan, items in each transaction are sorted in order TWU values and the utility list of each item is constructed and UP tree will be generate after that run the TKU and TKO algorithm based on algorithm transaction values will be generate. In the below chart I can observe that difference between the length of TKU Algorithm and TKO Algorithm



I can observe that Minimum utility threshold comparison chart between TKU and TKO algorithms chart TKU algorithm length is higher than TKO Algorithm length.

The difference will be shown in the sense of Runtime. So I can consider that by using this TKU and TKO algorithm to read the item-sets easily. Through our implementation I can improve the performance of the system at lower cost then compare to current methods.

V.CONCLUSION

In this paper, I have studied the problem of top-k high utility item sets mining, wherever k is that the desired range of high utility item sets to be mined. Two efficient algorithms TKU (mining Top-K Utility item sets) and TKO (mining Top-K utility item sets in one phase) are projected for mining such item sets without setting minimum utility thresholds. TKU is the initial two-phase algorithm for mining Top-k high utility item sets, which effectively raise the border minimum utility thresholds and additional prune the search area. On the other hand, knockout is that the initial one-phase algorithm developed for top-k HUI mining that integrates the novel methods RUC, RUZ and EPB to greatly improves its performance. Empirical evaluations on different types of real and synthetic datasets show that the projected algorithms have good measurability on giant datasets and also the performance of the projected algorithms is close to the optimal case of the state-of-threat two-phase and one phase utility mining algorithms.

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.

[11] J. Liu, K. Wang, and B. Fung, “Direct discovery of high utility item-sets without candidate generation,” in Proc. IEEE Int. Conf. Data Mining, 2012, pp. 984–989.

[12] Y. Lin, C. Wu, and V. S. Tseng, “Mining high utility item-sets in big data,” in Proc. Int. Conf. Pacific-Asia Conf. Knowl. Discovery Data Mining, 2015, pp. 649–661.

[13] Y. Li, J. Yeh, and C. Chang, “Isolated items discarding strategy for discovering high-utility item-

sets," Data Knowl. Eng., vol. 64, no. 1, pp. 198–217, 2008.

[14] G. Pyun and U. Yun, "Mining top-k frequent patterns with combination reducing techniques," Appl. Intell., vol. 41, no. 1, pp. 76–98, 2014.

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