ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 9, September 2019

Mining Algorithm for Mining High Utility Itemset Using TKO with TKU

Krishna P N, Prof. Sharmila Shinde

JSPM Imperial College of Engineering and Research, Wagholi, Pune, India

JSPM Imperial College of Engineering and Research, Wagholi, Pune, India

ABSTRACT: The mining of high utility element sets (HUI) or the Utility Item-sets (UI) mines the frequent item sets according to the minimum threshold. The mining of top K HUI mines HUIs as per user interestingness instead of the minimum threshold. Recent studies suggest different approaches for mining HUIs. In the existing HUI algorithms, candidate item sets will be spawned in mining process and when the data set is huge, the time and space performance may get affected. Providing the minimum threshold of the utility is also another issue. This paper proposes a new framework for finding top k HUI mining by combining two algorithms named TKO – extraction of K high utility element sets - in one phase) and TKU (mining of K high utility item sets Top-K Utility). It can be used in business analysis and online shopping etc. During this study, it can be proved that by using the proposed hybrid approach, the mining have significantly better performance with accuracy without providing the minimum threshold utility.

KEYWORDS: TOP K utility mining, Mining Utility item set , Mining TOPK high utility item set, high utility item set, utility pattern mining.

I. INTRODUCTION

Data mining is extracting the efficient and vivid data from a large set of unprocessed data. One of the well-liked problems in data mining is frequent pattern mining, and it consists of finding frequent patterns in transaction databases. Find the frequent item sets is the goal of frequent item set mining. Apriori, FP Growth algorithms are example for some of the popular algorithms proposed for frequent item set mining. The input to these algorithms is transaction database and the parameter min sup alias minimum support threshold. And it's output is all itemsets which present in at least minsup transactions. The drawback of frequent item set mining is while analyzing customer transactions, the purchase quantities are not considered. Another important drawback is that all items are considered to be having the same weight, utility of weight and may find many frequent patterns that are not remarkable to the user. In Frequent item set mining (FIM) [10], only the frequent elements are discovered instead of high utility item-sets or and the reason for the same is the neglecting of the amount of purchase. Utility mining is an important topic in data mining and introduced to deal with the drawback of frequent pattern mining by allowing for user's interest. Depending on the user, utility can be based on interest, importance, or profitability of an item. The two aspects of the utility of an item set is external utility (significance of divergent items) or internal utility (the significance of items present in transactions). The product of external and internal utilities can be defined as the utility of an item set [1]. The popular problem of data mining is the mining high utility element sets or, utility item sets mining. The predicament of HUI is primarily the preface to the set of frequent elements depending on the user interestingness. Most of the existing utility mining algorithm uses a two phase candidate generation [2]. The first phase (phase 1) is finding the candidate with high utility pattern by scanning the database. In the second phase, the database is again read or scanned for identifying the high utility patterns among the candidates. The problem of HUI – high utility itemset mining is to locate the itemsets that produce high profits when sold in concert. The value of minimum utility threshold (minutil) has to be provided by the user. If an item's utility is not less than a user specified utility, it is considered as a high utility item set (HUI).Selecting an appropriate minimum threshold is difficult for users. Based on the specified threshold, the size of the output can vary from very small to very large. The preference of the threshold extensively impacts the performance of the algorithms. Too many HUI will be displayed to users

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 9, September 2019

when the threshold is too low and it will be difficult for users to locate the apt results. Data mining algorithms is considered unproductive or out of memory when a large amount of HUIs are found, and subsequently use more resources. When the threshold is too high, HUI will not be found

II. BACKGROUND

In Utility mining, every unique item will associated with a utility such as unit profit and a weighted of transaction, like quantity. The utility figures the importance of it. When the utility of an item is greater than or equal to the minimum utility (min util) specified by the user, it is called as a high utility itemset (HUI). If the threshold is not selected properly, the performance of the algorithm will be affected. If the threshold set to low, lot of HUIs will be displayed and difficult for the end user to decide. To find the appropriate min util threshold, numerous execution should be performed by the user with different values, which is a time consuming and inconvenient. When there is a large data set is present it will be wasting the memory and process time also. Deciding the minimum utility threshold value finding is difficult because it primarily depends on the database characteristics and is unknown to user. Effective mining of HUIs in databases is difficult task due to the downward closure property used in frequent item set mining and it does not hold for the efficacy of item sets. Some Sometimes, a super set of a low utility item set can be displayed as a high utility, so it is difficult for pruning the search space and finding HUI.

2.1 Related Work

Frequent Pattern Tree [1] (FP-tree) structure is an extension of the Xtree structure.The FP-tree based mining uses a pattern fragment growth method and avoids the generation of large number of candidate sets and thus reducing the cost. For efficiently mining top-k frequent closed item sets (CHUIs) [6], without minimum support the TFP algorithm [2] is used. To accelerate closed frequent pattern mining, a top-down and bottom up combined FP-tree mining approach has been developed. The performance analysis reveals that in most cases, TFP outperforms the efficient frequent closed pattern mining algorithms CLOSET and CHARM, even though they are executing with the best tuned min-support. Mining high utility item sets [3] from a large database the existing HUI miner [8] is less efficient than UP-Growth and UP-Growth+ [1] [6]. A tree-based data structure known as utility pattern tree (UP-Tree) [9] [10] is used to store the data of high utility item sets in the UP Growth algorithm. The UP tree [9] [10] maintains the information about the transactions and item sets and helps to reduce repeated scan of the database. The UP tree scans the database twice to obtain the candidate items and manage in an efficient and structured way. Utility mining addresses the limitation of frequent item set and this is achieved by introduce the interest measure of the user expectation. The issue with the utility mining is to hold the non anti-monotone interest property [4] and which causes the scalability issue because of the high number of candidate generation. Sequential top-k HUIM [5] an approach without generating the candidate known as the growth approach and is a single phased one, finds the top-k high utility sequential patterns by the itemset and sequence concatenation process. For Mining High Utility Item Sets in Big Data (for example on Hadoop), PHUIGrowth [7] (Parallel mining High Utility Item sets by design Growth) algorithm is used for parallel mining and proved that PHUI-Growth has privileged on vast scale data sets and beats cutting edge non-parallel kind of HUI mining algorithms. [7] TKUL-Miner [8] is used for proficiently mining the top-k high utility item sets. For effectiveness, this uses a separate utility structure for storing data in every step on the look tree mining and use utilizing scan arrange for explicit locale to raise the fringe least utility limit quickly. [8]. An improved UP-Growth High Utility Item set mining, suggests an improvement in the extraction of objects with high utility UP-Growth [8]. The UP-tree, stores the details of the transactions and its data sets which improves mining performance by avoiding frequent scanning of the database [9].

III. PROPOSED METHOD

Mining is major task in any application and performances of mining with different algorithms such as one Phase and different phases. To Get High Utility item Set TKO and TKU with Parallel Mining. Two algorithms called TKU (mining of sets of utility elements Top-K) and TKO (mining Top-K itemsets in one phase) to mine them set of items without providing a utility minimum. The TKO algorithm has the disadvantage that result of TKO may contain garbage value in the set high utility items. The result of the TKU algorithm is accurate but the execution time is high. Another solution is to locate the efficient algorithm in the anticipated combination of the TKO and TKU algorithm. The result of TKO-Top K in one phase is given to the TKU - Top K in the utility result of TKO and the output is accurate and the execution time is low. It has been observed that the execution time of TKO algorithm is less but result is inaccurate (garbage value can be present). The execution time of TKU algorithm is more but result is accurate. It is very

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 9, September 2019

challenging issue how hybrid algorithm (TKO WITH TKU) is efficient than TKU algorithm. The time factor is very important while considering effectiveness of a system. Require to achieve significantly high performance.

3.1 ARCHITECTURE

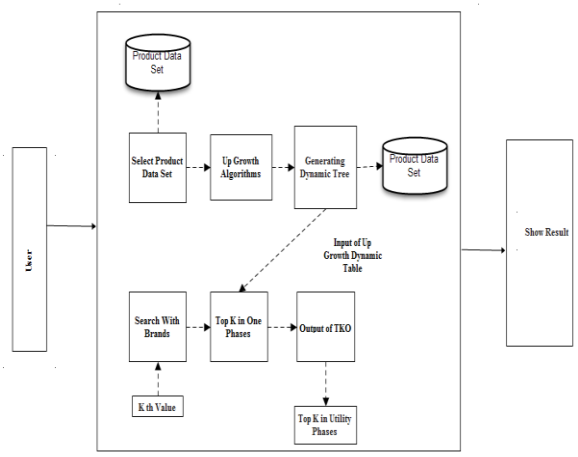

Figure 1 System Architecture

In the proposed framework, the high utility item sets (HUIs) found by parallel mining using the algorithms TKU and TKO. Two algorithms called TKU (mining Top-K HUIs) and TKO (mining Top-K UIs in one phase) combined to mine the itemset without providing a utility minimum. The disadvantage of TKO algorithm is collecting garbage values in the high utility items sets. The result of the TKU algorithm is accurate but the execution time is high. The solution for this is to discover a proficient algorithm by combining the TKO and TKU algorithms. It can be said that the result of TKO (Top K in one phase) is given as input to TKU (Top K in the utility). The result has accuracy in the data and the execution time is low. In the proposed system, a new algorithm is generated for combining the name TKO and TKU as TKMHUI Top k high utility itemsets. Two competent algorithms named TKO (mining Top-K UI in one phase) and TKU (mining Top-K UIs) are anticipated for mining the top-k HUIs without providing the min_util threshold. The proposed algorithm has less search space so it needs less memory. User can search and order the products from the data set. When the user successfully places an order, the UP-Growth algorithm is executed. If a user requires finding the top-K product, user has to provide the parameters hit or buy and number of itemset (k) to be displayed. Finding HUI and PHUI, in the existing systems uses the TKO and TKU respectively. Even though the TKO is more efficient in terms of the execution time, the HUI may be incorrect and cannot rely. For TKU, the HUI will be correct it is a time consuming process. The implemented system receives the k value (number of itemsets to be displayed) and use the data is processed using TKO. Output of TKO is given to TKU which removes the garbage values present in the input. Time consumption is half when compared the time consumed by TKU and better accuracy when compared to TKO output.

3.2 Mathematical Model

The mathematical model is given below –
1. S=Tp, RTU, TU, UP Tree, TWU, PHUI+ UP growth where
2. S=system
3. Tp=represents the set of transaction and profit of each item
4. Tu= Utility of Transaction
5. TWU= Transaction Weighted utility
6. RTU=Utility of Recognized Transaction

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 9, September 2019

7. UUI = Unpromising Item's utility
8. UP tree= Utility Pattern Tree.
9. UP growth+ = utility Pattern Growth + algorithm
10. Improved UP growth (Up growth +)= Advanced UP growth algorithm
11. PHUI= Potentially HUI
12. Task: Finding HUI
13. **Input**: Data Base of Item set
14. Transaction T Belong DB i=1,k=1
15. Ip is the Internal Utility value of item
16. H: refers to the HUI
17. CK: Candidate set of items
18. Minimum Utility threshold min_Util
19. **Output**: HUI- H

3.3 ALGORITHMS

3.3.1 TKO Algorithm

The TKO algorithm is used to find TOP-K high utility item sets in one phase and uses the standard search procedure of existing HUI miner and its utility list structure. Even though the execution time is low, the disadvantage of this method is it may contain garbage values in the result and user cannot rely on the result.

id	Brand name	Product name	SKU	Unit price
1	BN 1	Berry Juice	9074674	30
2	BN 1	Mango Drink	9651499	18
3	BN 1	Strawberry Drink	5842925	17
4	BN 1	Cream Soda	6441747	26
5	BN 1	Diet Soda	8556439	7
6	BN 1	Cola	2980796	14

Table 1: Sample data set

Consider, the user require to find the Top K HUIs for parameter hit or buy and k =7. The output of TKO algorithm is given in table 2. The data in bold is the representation of garbage value.

id	brand name	Product name	SKU	Unit price	P-Hit	P-Buy
1	BN 1	Berry Juice	9074674	30	20	**XYZ**
4	BN 1	Cream Soda	**$fhafjk**	26	18	12
6	BN 1	**@asdk**	2980796	18	10	10
2	BN 1	**23jhj**	9651499	17	9	9
5	BN 1	Diet Soda	8556439	**45m**	8	6
3	BN 1	Strawberry Drink	5842925	7	**VZ 1**	8

Table 2: Output TKO

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 9, September 2019

3.3.2 UP Growth Algorithm

The UP (Utility Pattern) growth [1] [6] algorithm creates a tree structure named utility pattern tree for finding HUIs and to maintain the key data linked to utility patterns in the database. The data of HUIs is preserved in a tree data structure known as UP-Tree (Utility Pattern Tree) [9] [10] so that the candidate itemsets can be spawned efficiently by scanning the database twice. The Utility Pattern Growth (UP- Growth) used a tree like data structure to store and maintains the high utility itemsets. An itemset is called a high utility itemset when its utility is greater than or equal to the user-specified minimum utility threshold (min_util) else considered as low utility itemset.

Each node of an UP Tree contains the column headers such as N.name(name of the item in the node), N.count(support count of the node), N.nu(utility of node), N.parent(parent of the current node), N.hlink(node link which points to a node whose item name is the same as N.name) and a set of child nodes. For UP Tree traversal, make use of the header table component. In the header table each entry is poised by name of an item, value of the utility, and the last occurrence of the node pointed by the link and which contains the same item as the entry in the UP-Tree. The link in the header table and the nodes in UP-Tree, are used to find the nodes whose item names are the same can be traversed efficiently.

An UP-Tree is constructed by performing by scanning of the original database twice. The transaction utility of each transaction is computed in the first scan of database and in parallel TWU of each item is calculated. After traversing the database, items and corresponding TWUs are found. If the TWU of an item is less than minimum utility threshold, its supersets are considered as unhopeful to be high utility item sets and called as unpromising items. An item "ip" is called a promising item if min_util ≤ TWU (ip). Else it considered as an unpromising item. Transactions are inserted into UP-Tree in the subsequent search of the database. Primarily, the tree is formed with root R. When a transaction is repossessed, removal of the unpromising items is done and only the super sets of promising items are considered as the high utility item sets. The promising items in the transaction are arranged in the descending order of TWU.

id	Brand name	Product name	SKU	Unit price	P-Hit	P-Buy
1	BN 1	Berry Juice	9074674	30	20	15
4	BN 1	Cream Soda	6441747	26	18	12
6	BN 1	Cola	2980796	18	10	10
2	BN 1	Mango Drink	965199	17	9	9
5	BN 1	Diet Soda	8556439	14	8	6
3	BN 1	Strawberry Drink	5842925	7	7	8

Table 3: UP Growth table

3.3.3 TKO with TKU algorithm

For obtaining top-k HUIs without specifying min util, the TKU (mining Top-k Utility item sets) algorithm is used.TKU is an annex of the tree-based algorithm(UP Growth) for HUI mining. For maintaining the top-k HUIs and transaction information, TKUbase adopts the UP-Tree structure of UP-Growth. The steps involved in executing TKUbase: (1) construct the UP-Tree then generate PKHUIs (Potential top- K HUIs) from the UP Tree, and finally identify top-k HUIs from the PKHUIs.

For traversing the UP-tree, the header table is used. The header table entry contains name of item, the utility value, and a link. The link points to the first node in the UP Tree having the same item name as an entry. A UP-Tree can be constructed by scanning the initial database. The TWU of each item are computed and transaction utility of each

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 9, September 2019

transaction is calculated in the first scan and result is, the items and corresponding TWUs are found. The items are inserted into the header table in the decresasing order of its TWUs. Transactions are reorganized and inserted in up tree by the second step.

The output of TKO is given as input to the TKU algorithm and when TKU while reading the input, when it reads garbage value, it will compare the data with the existing UP Growth table, otherwise it will construct the UP Growth table. In the hybrid algorithm, by using the output of the TKO, finding the HUIs is less consuming. For the removal of the inaccurate value, the UP-Growth tree is used which saves the scanning of database and which helps to improve the execution time than the TKU algorithm.

Input

1) Data base DB and 2) the number of desired HUIs k;

Output

The complete set of PKHUIs P;

Steps:-

1 Set min_util Border ← 0; TopKHUI ─ MIU ─ List ─; P ─;
2 Construct an UP <=> Tree by scanning DB two times;
3 Apply UP-Growth algorithms for generating PKHUIs;
4 For each PKHUI spawned with expected utility ESTU(X) do
5 If (ESTU(X) = min_util Border and MAU(X) min_util Border)
6 Output X and min ESTU(X),MAU(X) ;
7 If (MIU(X)min utilBorder)
8 Lift up min utilBorder by the approach MC;
9 Min utilBorder-MC(MIU(X),TopK-MIU-Lit);
10 Stop

Id	Brand name	Product name	SKU	Unit price	P-Hit	P-Buy
1	BN 1	Berry Juice	9074674	30	20	15
4	BN 1	Cream Soda	6441747	26	18	12
6	BN 1	Cola	2980796	18	10	10
2	BN 1	Mango Drink	9651499	17	9	9
5	BN 1	Diet Soda	8556439	14	8	6
3	BN 1	Strawberry Drink	5842925	7	7	8

Table 4: TKO with TKU result

IV. RESULT AND DISCUSSION

Experiments are done by a personal computer with configuration: Intel (R) Core (TM) i3-2120 CPU @ 3.30GHz, 4GB memory, Windows 7, MySQL 5.1 backend database and Jdk 1.8. The application is web application used tool for design code in Eclipse and execute on Tomcat server.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 9, September 2019

The experiment is done on the product data base which contains more than 1500 records. TKO graph shows the execution time required for finding TOP K value. The X axis displays the time in milliseconds and Y axis is the search index. Mouse over will display the value of the execution time in mille-seconds.

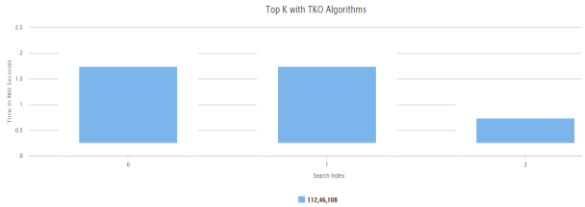

Figure 2: Execution time graph of TKO algorithm

TKU graph shows the execution time required for finding TOP K value using TKU algorithm. The X axis displays the time in milliseconds and Y axis is the search index. Mouse over will display the value of the execution time in Millie-seconds. At execution time, the data is stored in database and used to create the graph.

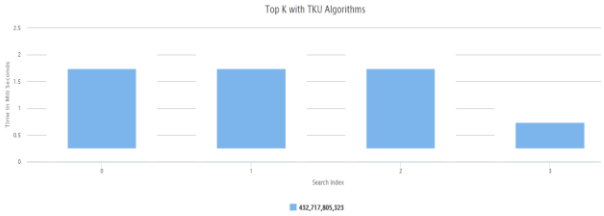

Figure 6: Execution time graph of TKU algorithm

Execution graph of hybrid algorithm shows the execution time required for finding the value for the implemented hybrid algorithm. The X axis displays the time in milliseconds and Y axis is the search index. Mouse over will display the value of the execution time in mille seconds.

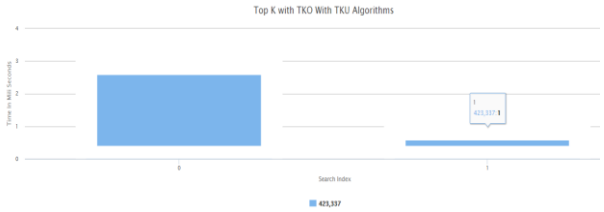

Figure 7: Execution time graph of Hybrid algorithm

The execution time of the hybrid method is less than the TKU algorithm. When compared to the TKO, data is accurate. The comparison graph of all three algorithms shows the difference in the execution time.

The Hybrid algorithm is proved efficient as its execution time is less compared to TKU and accurate when compared to TKO.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 9, September 2019

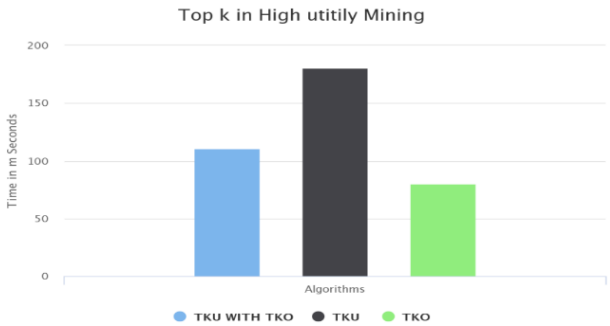

Figure 8: Comparison Graph of TKO WITH TKU

Number	Algorithms used	Execution Time in MS	K value as input
1	TKO	521	3
2	TKU	23059	3
3	TKO WITH TKU	19418	3

Table 5: Algorithms compared based on the execution time

Sr. No	Parameter	TKO	TKU	TKO with TKU
1	Anti-monotonicity	Yes	No	Yes
2	Adjoining HUI	Yes	Yes	Yes
3	Finding Garbage value in Algorithm	Yes	No	No

Table 6: Compare implemented system for different parameters.

V. CONCLUSION

This work looked at the problem of the finest sets of high-utility item sets, where k is the desired number of high utility item sets to mine. The most capable permutation of TKO WITH TKU is put into action to mine such sets of items without providing minimum utility threshold. On the other hand, TKO is the first single-phase algorithm used for mining top-k HUIs named PHUI (potential high utility item sets) and PHUI is the input to TKU in the service phases. Experimental valuations on real data sets show that the implemented algorithms have high scalability in huge data sets and the performance of the implemented algorithms are close to the optimal case. TKO's execution time is less but the data is inaccurate (due to the presence of garbage values) and TKU's execution time is high even though the data is accurate. The hybrid algorithm executed in moderate time(less compared to TKU's execution time) with better accuracy (compared to TKO) of data.

In our research attempt, we observed that enhancement of hybrid algorithm to reduce execution time more significantly. It can be integrated with other utility mining tasks to find out different types of top-k high utility item sets. It can be widely applied with different domains like product recommendation system, web page recommendation system, online course recommendation system etc.

REFERENCES

[1] Vincent S. Tseng, Bai-En Shie, Cheng-WeiWu, and Philip S. Yu, Fellow, IEEE"Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases" IEEE Trans. Knowledge and Data Engineering, vol. 25, no. 8, August 2013.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 9, September 2019

[2] H. Simpson, *Dumb Robots*, 3rd ed., Springfield: UOS Press, 2004, pp.6-9. Junqiang Liu; KeWang ; Benjamin C.M. Fung, Mining High Utility Pattern in One Phase Without Generating Candidates- IEEE Transactions on Knowledge and Data Engineering Volume: 28 , Issue: 5 , May 1 2016

[3] J. Liu, K.Wang, and B. Fung, Direct discovery of high utility itemsets without candidate generation, IEEE 12th International Conference on Data Mining, pp. 984-989, December 10-13, 2012.

[4] Y. Lin, C. Wu, and V. S. Tseng, Mining High Utility Itemsets in Big Data -Advances in Knowledge Discovery and Data Mining. PAKDD 2015. Vol 9078. Springer, Cham

[5] J. Yin, Z. Zheng, L. Cao, Y. Song, W. Wei, Efficiently Mining Top-K High Utility Sequential Patterns - IEEE International Conference on Data Mining, ICDM. 978-0-7695-5108-1. 2013

[6] V. S. Tseng, C. Wu, P. Fournier-Viger, P. S. Yu, Efficient algorithms for mining the concise and lossless representation of high utility itemsets IEEE Transactions on Knowledge and Data Engineering . Volume: 27, Issue: 3, March 1 2015.

[7] Savarimuthu N., Ramaraju, C. A conditional tree based novel algorithm for high utility itemset mining, Int. Conf. on Data mining, 2011.

[8] Serin Lee ; Jong Soo Park, Top-k high utility itemset mining based on utility list structures - 2016 International Conference on Big Data and Smart Computing (BigComp), Hong Kong, 2016, pp. 101-108.

[9] O Srinivasa Rao, Adinarayana reddy B , MHM Krishna Prasad, , An Improved UPGrowth High Utility Itemset Mining - International Journal of Computer Applications 58(2):25-28, November 2012

[10] P. Asha, Dr. T. Jebarajan, G. Saranya, A Survey on Efficient Incremental Algorithm for Mining High Utility Itemsets in Distributed and Dynamic Database IJETAE Journal, Vol.4, Jan 2014.