Mining Top-k High Utility Itemset using Efficient Algorithms

Pramod Pardeshi[1], **Ujwala Patil**[2]

[1, 2]Department of Computer Engineering, R. C. Patel, Institute of Technology Shirpur, MS, India

Abstract: ***Data mining uses different algorithms for seeking interesting information and hidden patterns from the expansive database. Traditional frequent itemset mining (FIM) create substantial measure of incessant itemset without thinking about the amount and benefit of thing obtained. High utility itemset mining (HUIM) gives profitable outcomes as contrasted to the frequent itemset mining. HUIM algorithm helps to enhances the performance of discovering data by considering both quantity and profit of itemset from large database. This paper review algorithm TKU (mining top-k utility itemset) for mining high utility itemset without any need to set minimum utility threshold by using strategy of UP-tree data structure which checks the database twice and upgrades the effectiveness of mining High utility itemset. It discover transaction utility of each transaction and it also compute TWU of each item. Then it rearranges the transaction and develops the Up Tree.***

Keywords: Data mining, Frequent itemset, High utility itemset, utility pattern tree

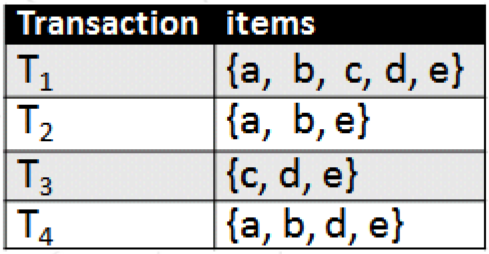
1. Introduction

Data mining is a method to separating the data from the substantial database. Association rule mining discover the large transaction databases for association rules which give the implicit relationship among data attributes. Data mining also called as knowledge disclosure in databases. The most challenging data mining tasks is the mining of high utility item sets efficiently. Identifying the item sets with high utilities is called as Utility Mining. The utility can be estimated based upon cost, profit or other expressions of user preferences.

The normal item set is not sufficient to recreate the actual utility of an item set. Frequent item sets are the item sets that discovered frequently in the transaction data set. The aim of rehashed Item set Mining is to distinguish the frequent item sets in a transaction dataset. Utility mining provide an important topic in the data mining field. Mining high utility item sets from databases also known to finding the item sets with high profits. The significance of item set utility is importance, or effectiveness of an item to users. Utility of an item set is characterized as the product of its external utility and its internal utility [1].

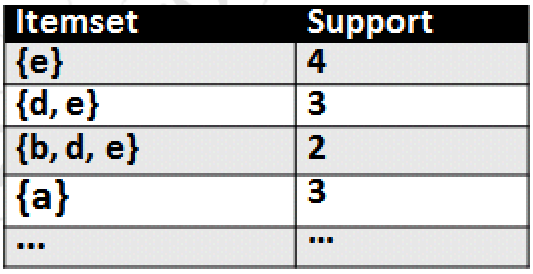
1.1 Frequent item set mining

The issue of high-utility item set mining is an expansion of the problem of repeated model taking out common pattern mining is a popular problem in data mining, which consists in finding frequent patterns in transaction databases. Let me explain first the problem of frequent item set mining Consider the following database. It is a transaction database. A transaction database is a database contain a set of transactions made by customers. A transaction is a set of items buy by a customer. In case the following database, the first customer buy items “a”, “b”, “c”, “d” and “e”, while the second one buy items “a”, “b” and “e” [19]. The target of rehashed item set mining is to discover frequent item sets. Numerous prevalent algorithms have been proposed for this issue such as Apriori, FP Growth, LCM, Eclat, etc. These algorithms take as enter a transaction database and a factor “minsup” called the minimum support threshold. These algorithms then return all set of items that appears in at least minsup transactions [19].



Transaction	items
T_1	{a, b, c, d, e}
T_2	{a, b, e}
T_3	{c, d, e}
T_4	{a, b, d, e}

For example, if we set minsup = 2, in our example, we would discover many frequent item sets such as the following:



Itemset	Support
{e}	4
{d, e}	3
{b, d, e}	2
{a}	3
...	...

Consider the following item set {b, d, e}. It has a support of 3 because it get in three transactions, and it is said to be frequent because the support of {b, d, e} is no less than minsup.

1.2 Frequent item set mining has some important limitations

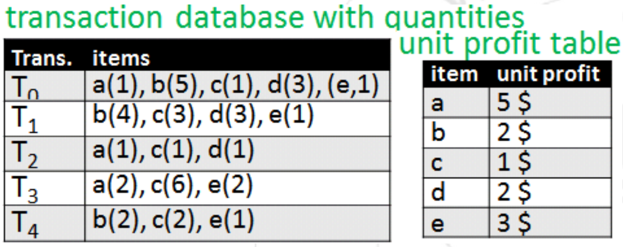
1) The issue of frequent item set mining is prevalent. But it has some vital restrictions when it comes to analyzing customer transactions. An vital constraint is that purchase quantities are not taken into account. Thus, an item may only appear ‘1’ or ‘0’ time in a transaction. Thus, if a customer has bought five breads, ten breads or twenty breads, it is viewed as the same [20].

2) A next critical confinement is that all items are viewed as having the similar significance, utility of weight. For example, if a customer buys a very expensive bottle of wine or a cheap piece of bread, it is viewed as being similar significance [20].

Thus, frequent pattern mining may discover many frequent patterns that are not interesting. For instance, one may discover that {bread, milk} is a frequent pattern. However, from a business perspective, this pattern may be uninteresting because it does not generate much benefit. Moreover, frequent pattern mining algorithms may miss the rare patterns that generate a high profit such as perhaps {caviar, wine} [20].

1.3 High-utility item set mining

To address these limitations, the problem of frequent item set mining has been redefined as the problem of high-utility item set mining. In this issue, a transaction database contains transactions where purchase quantities are taken into considered as well as the unit profit of each item. For example, consider the following transaction database [20].



transaction database with quantities

Trans.	items
T_0	a(1), b(5), c(1), d(3), (e,1)
T_1	b(4), c(3), d(3), e(1)
T_2	a(1), c(1), d(1)
T_3	a(2), c(6), e(2)
T_4	b(2), c(2), e(1)

unit profit table

item	unit profit
a	5 $
b	2 $
c	1 $
d	2 $
e	3 $

2. Literature Survey

Vincent S. Tseng et al , They address the new framework for top-k high utility item set mining, where k is the desired number of HUIs to be mined. Two types of efficient algorithms named TKU (mining Top-K Utility item sets) and TKO (mining Top-K utility item sets in One phase) for mining such item sets without the need to set min_util. They provide a structural comparison of the two algorithms with discussions on their advantages and limitations. Empirical evaluations on both real and synthetic datasets show that the performance of the algorithms is close to that of the optimal case of state-of-the-art utility mining algorithms. [1]

Y. Liu, W. Liao, and A. Choudhary proposed two phase algorithm [2] to overcome the limitation of utility mining and mine high utility item sets from the database. In first stage algorithm demonstrate transaction weighted utilization, and find out the transaction weighted utilization model and this model support transaction-weighted downward closure property. In the last phase one additional database scan is carried out to filter out the overestimated item sets. The proposed algorithm needs some database checks, low memory space and decrease the computation cost. The main important property of proposed algorithm is that, it can efficiently tackle very huge databases in regards to present algorithm.

R. Agrawal, R. Srikant, T. Imielinski proposed an efficient algorithm [3] for generation of significant association rules between items in the database. The algorithm used to enhance good buffer management, novel estimation and search technique. Proposed Algorithm show the output of sales data discovered from a big retailing company.

Serin Lee, Jong Soo Park suggested a new algorithm, TKUL-Miner [4], to mine top-k high utility itemsets efficiently. It uses new utility-list structure for maintaining necessary information at each and every node on the search tree for mining the item sets. To decrease the search space pruning it uses the efficient tree base structure named UP-Tree. Authors shows efficient algorithm to generate the border minimum utility threshold fast. Also, for calculating smaller overestimated utilities, two extra methods are given to prune unpromising itemsets adequately.

S. Shankar, T. P. Purusothoman, S. Jayanthi, and Nishanth Babu proposed a novel algorithm named Fast Utility Mining (FUM) [5] as an alternative to High utility mining. All sets of high utility item sets within the given utility constraint threshold are found by FUM algorithm. To generate different combination of item sets the authors also suggest a technique such as Low Utility and High Frequency (LUHF) and Low Utility and Low Frequency (LULF), High Utility and High Frequency (HUHF), High Utility and Low Frequency (HULF). FUM algorithm is quicker and easy than high utility mining algorithm. As compare to utility mining algorithm, FUM algorithm find outs high utility item sets quicker in the case where the number of item in database are more.

T. Vinothini, V.V. Ramya Shree suggested algorithm named UP-Growth [6] for mining high utility item sets from large transaction database. The algorithm uses the UP-Tree data structure to maintain the information of high utility item sets. By using the UP-Tree the PHUIs can be efficiently discover in only two database scans. They propose several strategies to enhance the search space pruning of high utility item sets and help to reduce the overestimated utilities. The node processing increases with the expanded database size and due to that time utilizations and memory usage also more. To tackle this issue Systolic tree data structure is help to actualize the parallelism in processing node. High throughput and quicker execution time are the key point of systolic tree base structure.

Shuning Xing, Fangai Liu, Jiwei Wang, Lin Pang and Zhenguo Xu by introducing a Fast Utility Tree [7] (FU-Tree) proposed an UP-Tree process. The proposed method utilize the Link Queue to diminish the number of original database checks and embraces prefix utility to minimize the overestimated utility. The theoretical analysis and test outputs demonstrate that FU-Tree requires less time for the development of tree than UP-Tree, and provide best adaptability for mining high utility itemsets.

Junqiang Liu, Benjamin C.M. Fung proposed a novel algorithm named d2HUP [8] for mining high utility itemsets inone phase without generating candidates. It uses the novel strategy such as high utility pattern growth approach, a lookahead strategy and a linear data structure for mining high utility itemset.

Hong Yao, Howard J. Hamilton proposed Mining itemset utilities from transaction databases [9]. A utility based itemset mining way to deal with this confinement. The proposed method permits users to quantify their preferences concerning the handiness of itemsets portrayed utility values. That is, an itemset is intriguing to the user only if it fulfills a given utility requirement. We show that the pruning methods used in old itemset mining approaches cannot be applicable to utility limits.

P. Yamini, Soma Shekar, J. Deepthi proposed "Efficient Algorithms for Mining Top-K High Utility Itemsets" [10], High utility thing sets (HUIs) mining is a rising subject in information mining, which alludes to finding all thing sets having an utility meeting a client determined least utility edge min_util. Notwithstanding, setting min_util suitably is a troublesome issue for clients. As a rule, finding a fitting least utility edge by experimentation is a monotonous procedure for clients. In the event that min_util is set too low, an excessive number of HUIs will be produced, which may bring about the mining procedure to be exceptionally wasteful. Then again, if min_util is set too high, it is likely that no HUIs will be found.

Yogita Khot, Manasi Kulkarni," Survey on High Utility Itemset Mining from Large Transaction Databases" [11]. The proposed method Conventional data mining techniques have targeted more on discovering the items that are more frequent in the transaction databases, which is also called frequent itemset mining. These data mining methods were based on support-confidence model. Itemsets which appear more in the database must be of more valuable to the user from the business point.

Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer, Byeong-Soo Jeong, and Young-Koo Lee"[12], Efficient Tree Structures for High Utility Pattern Mining in Incremental Databases", proposed three novel tree structures to efficiently perform incremental and interactive HUP mining. The first tree structure, Incremental HUP Lexicographic Tree (IHUPL-Tree), is arranged according to an item's lexicographic order. It can catch the gradual data without any rebuilding activity. The second tree structure is the IHUP Transaction Frequency Tree (IHUPTF-Tree), which acquires a minimal size by rebuilding items regarding to their transaction frequency (descending order). To reduce the mining time, the third tree, IHUP-Transaction-Weighted Utilization Tree (IHUPTWU-Tree) is designed based on the TWU value of items in descending order. Extensive performance analyses show that our tree structures are very efficient and scalable for incremental and interactive HUP mining.

Prajakta R. Padhye, R. J. Deshmukh," A marketing solution for cross-selling by high utility itemset mining with dynamic transactional databases"[13] introduce a system which uses HUPID-Tree structure to maintain the information about the database and patterns and it is updated only with the incremented data. It reduces the time overhead of rescanning the database from the beginning. High utility itemsets (HUIs) i.e. the desirable patterns mined from the HUPID-Tree will be used for generating rules. Cross selling benefit of each rule will be evaluated with the assistance of an target function i.e. the rule utility function. Cross selling is the act of selling among the established customers. It utilizes items in the subsequent part of a standard for proposal and provides future benefit information with the application of a standard. Managers can utilize this cross-selling benefit information to expand the profit and the itemsets which will be sold in the later on will also be the high utility itemsets.

Vid Podpecan, Nada Lavrac, and Igor Kononenko," [14] A Fast Algorithm for Mining Utility-Frequent Itemsets" presents a novel proficient algorithm FUFM (Fast Utility-Frequent Min-ing) which figure out all utility-frequent itemsets within the provided utility and support limit threshold. It is faster and simpler than the original 2P-UF algorithm (2 Phase Utility-Frequent), as it is based on efficient methods for frequent itemset mining. Experimental evaluation on artificial datasets show that, in contrast with 2P-UF, our algorithm can also be applied to mine large databases.

Jing Wang, Lei Zhang, Guiquan Liu, Qi Liu and Enhong Chen," On Top-K Closed Sequential Patterns Mining" [15], propose a very efficient algorithm named BI-TSP (Mining top-k closed sequential patterns with BI-Directional checking plan) with no candidate and generation for mining top-k frequent closed sequences with the least length. particular, we consider the BI-Directional Extension for frequent closed sequential patterns identification. View on BI- Directional Extension, we can directly use the closure checking scheme and effectively raise the minimum support threshold without candidate maintenance. In addition, They also propose two Novel pruning strategies by exploiting the properties of minimum length constraint. Our broad performance test with synthetic and real datasets exhibits that BI-TSP outperforms the baselines in both memory and running time.

D.Sathyavani, D.Sharmila," Efficient Algorithm for Finding High Utility Itemsets from Large Transactional Databases Using R-Hashing Technique"[16], They used sorting with R-hashing technique for the memory allocation. Subsequently candidate items are put away with their individual memory in UP tree. The experimental result shows that system is more effective according to the memory space and the number of candidate itemset generation and input and output operations.

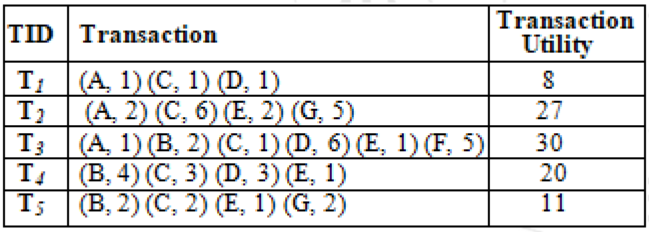
Feng Tao, Fionn Murtagh, Mohsen Farid," Weighted Association Rule Mining using Weighted Support and Significance Framework" [17], They address the issues of discovering significant binary relationships in transaction datasets in a weighted setting. Conventional model of association rule mining is adjusted to handle weighted association rule mining issue where each item is permitted to have a weight. The objective is to steer the mining center to those huge relationships involving items with significant weights rather than being flooded in the combinatorial explosion of insignificant relationships. They identify the challenge of using weights in the iterative process of generating large itemsets. The problem of invalidation of the "downward closure property" in the weighted setting is solved by using an improved model of weighted support measurements and exploiting a "weighted downward closure property". A new algorithm called WARM (Weighted Association Rule Mining) is created based on the enhanced model. adaptable and productive in finding significant relationships in weighted settings as illustrated by experiments performed on simulated datasets.

R.Nandhini, Dr.N.Suguna," Shrewd Technique for Mining High Utility Itemset via TKU and TKO Algorithm"[18], proposed a framework for mining top-k high utility item set , where k is the desired number of HUIs to be mined. Two types of capable algorithms named TKU (mining Top-K Utility item sets) and TKO (mining Top-K utility item sets in one phase) are proposed for mining such item sets without the need to set min_util.

Snehal D. Ambulkar, . Prashant N. Chatur," Efficient Algorithms for mining High Utility Itemset"[19], reviews two kinds of efficient algorithm named TKU (mining top-k utility itemset) and TKO (mining top-k utility itemsets in One phase) for mining high utility itemset compelling reason to set minimum utility threshold by using strategy of UP-tree data structure which checks the database twice and upgrades the proficiency of mining High utility itemset. It find out transaction utility of each transaction and it also compute TWU of each item. At that point it revamps the transaction and constructs the Up Tree.

3. Problem Definition

Let be a finite set of distinct items $I^* = \{I_1, I_2, I_3...., I_m\}$. A Transactional database $D = \{T_1, T_2, T_3, ...T_n\}$.is set of transaction $T \in D$ is a subset of I^* and has an unique identifier r, called Tid. Each item $I_j \in T_r$ has a positive value $Q(I_j, T_r)$ called its internal utility in T_r.



TID	Transaction	Transaction Utility
T1	(A, 1) (C, 1) (D, 1)	8
T2	(A, 2) (C, 6) (E, 2) (G, 5)	27
T3	(A, 1) (B, 2) (C, 1) (D, 6) (E, 1) (F, 5)	30
T4	(B, 4) (C, 3) (D, 3) (E, 1)	20
T5	(B, 2) (C, 2) (E, 1) (G, 2)	11

Each item $I_j \in I^*$ is associated with a positive number $P(I_j, D)$ called its external utility . An *item set* $X = \{I_1, I_2, I_3...., I_L\}$ is a set of L distinct items, where $I_j \in T_r$ and L is the length of X. An L item set is an item set of length L.

Definition 1: Utility of an item.
The absolute utility of an item $I_j \in I^*$ in a transaction *Tr* is denoted as $EU(I_j, T_r)$ and defined as $P(I_j, D) \text{ x } Q(I_j, T_r)$.

Definition 2: Utility of an item set in a transaction
The absolute utility of an item set X in a transaction Tr is defined as $EU(X, T_r) = \sum I_j \in X(I_j, T_r)$

Definition 3: Absolute utility of an item set in a database
The absolute utility of an item set X in D is defined as $EU(X) = \sum_{T_r \in D \wedge X \subseteq T_r} (X, T_r)$.

Definition 4: Transaction utility and total utility.
The transaction utility (TU) of a transaction T_r is defined as TU $(T_r) = EU(T_r, T_r)$.

The total utility of a database D is denoted as $TotalU_{DB}$ and defined as $\sum_{T_r \in D} TU(T_r)$.

Definition 5: Utility of an itemset in a database.
The (relative) utility of X is defined as $U(X) = EU(X) | TotalU_{DB}$.

Definition 6: High utility itemset
An item set X is called high utility item set (HUI) if U(X) is no less than a user-specified minimum utility threshold *min_util* $(0\% \leq min\_util \leq 100\%)$. Otherwise, X is a low utility item set An equivalent definition is that X is high utility if EU(X) $\geq$ *abs_min_util, where abs_min_util =min_util* x $TotalU_{DB}$.

Definition 7: High utility itemset mining
Let $\delta(0\% \leq \delta \leq 100\%)$ be the minimum utility threshold, the complete set of HUIs in D is denoted as $f_{HUI}(D, \delta)$. The goal of HUI mining is to *discover* $f_{HUI}(D, \delta)$.

Definition 8: Transaction-weighted utilization
The trans-action-weighted utilization of an itemset X is the sum of the transaction utilities of all the transactions containing X, which is defined as. $TWU(X) = \sum_{X \subseteq Tr \wedge Tr \in D} TU(Tr)$.

4. Proposed System

Generation of top-k high utility item sets needs complete chain process. This process is delineated in the architecture diagram. According to this architecture first we determined the Transactional Utility and Transactional Weighted Utility by filtering the Transactional Database. This sweep is the first database scan. Subsequent step is to mine the minimum utility threshold. This is the most important step in utility item set mining. The minimum utility threshold can be any esteem paying little heed of dataset. If the minimum threshold is very less, then very large amount of irrelevant data is generated or if it is very high, then very less amount of data is retrieved. Therefore, a new approach will be followed to calculate the minimum utility threshold on the basis of the database.

Presently the unpromising itemsets are expelled because they won't generate any be benefit. Then, the Database is reorganized in the order of the profit values. This will be the second last database scan. This demonstrates the most encouraging itemsets above the less encouraging itemsets.

After rearranged database, this demonstrates the most encouraging itemsets. After that TKU and TKO algorithms are proposed for mining such itemsets without the need to set threshold value. Then the concept of re-ranking is applied to know what are the Top-k sets of items that contribute highest profit to the user'. Fig. 1 shows the Architecture of proposed system.

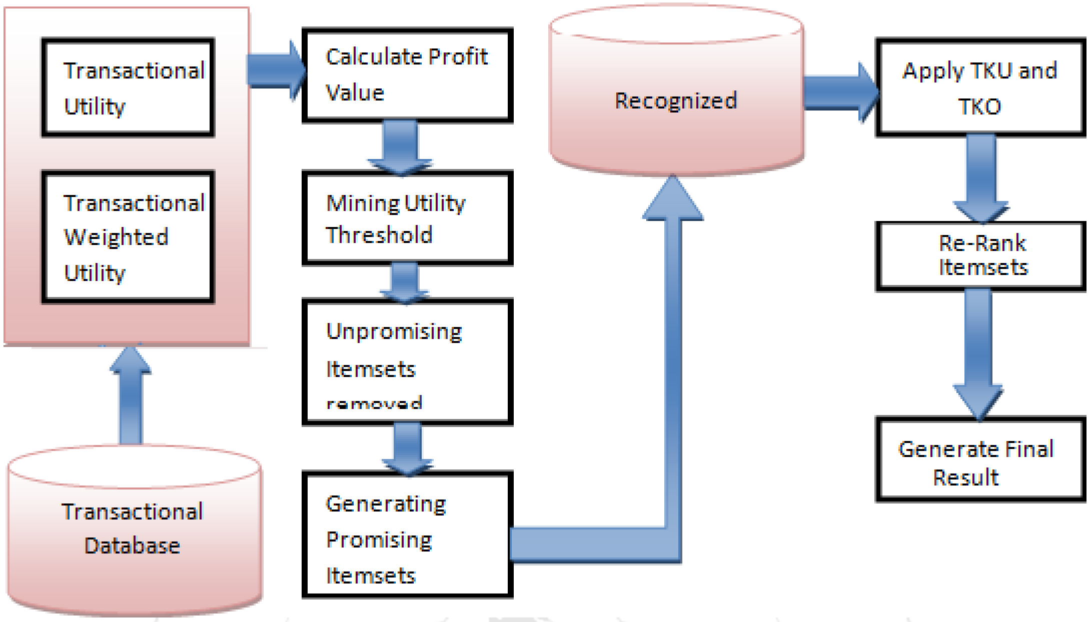


Figure 1: System Architecture

5. Algorithm

TKU Approach

We propose ideas to raise min_util$_{Border}$ during the Phase I of TKU$_{Base}$. Each time a candidate itemset X is found by the UP-Growth search procedure, the TKU$_{Base}$ algorithm checks whether it's estimated utility value ESTU(X) is no less than min_util$_{Border}$. If ESTU(X) is less than min_util$_{Border}$, X and all its concatenations are not top-k HUIs. Besides, Tubas checks whether MAU(X) is no less than min_util$_{Border}$. If MAU(X) is smaller than min_util$_{Border}$, X is not a top-k HUI, Something X is viewed a candidate for Phase II and it is yielded with min {ESTU(X), MAU(X)} according to Property. If X is a valid PKHUI and MIU(X) _ min_util$_{Border}$, MIU(X) can be used to raise min_util$_{Border}$ by the proposed strategy.

Input:
(1) A Database D;
(2) The number of desired HUIs k;

Output
(1) The complete set of PKHUIs C;

Set
$min\_util_{Border} \leftarrow 0; TopK-MIU-List \leftarrow \varnothing; C \leftarrow \varnothing;$
Construct a UP-Tree by scanning D twice;
//Apply a Up-Growth search procedure to generate *PKHUIs;*

For each PKHUI generate with estimated utility *ESTU(X)* **do**

If
$(ESTU(X) \geq \min\_until_{Border} \; \&\& MAU(X) \geq \min\_until_{Border})$
Output X and
$\min\{ESTU(X), MAU(X); C \leftarrow C \cup X;$
$IF(MIU(X) \geq \min\_util_{Border})$
{
$\min\_util_{Border} \leftarrow MC(MIU(X), TopK - MIU - Lit);$
} }

6. Methodology

In the initial step, Administrator keeps up the history of items bought by the customer and also maintains the database of transaction. In everyday life new product is discharged, so administrator would add new product to the stock and refresh the database. In daily market customers purchase various items. Administrator would have to keep up history of user purchase behavior and store in value-based database. In the second step, development of TKU algorithm is happen in three steps. In the absolute first step it constructs UP Tree. Next it produces potential top-k high utility itemsets and in the last step it finds the top-k high utility itemset from a set of potential top-k high utility itemsets. Construction of UP-Tree is performing by scanning the database twice as show in Fig. 2. In first scans, it finds out transaction utility of transaction and it also computes TWU of each and every item. And in second scan it reorganizes the transaction and constructs the Up Tree. The rearrangement of item is done according to their TWU as shown in the above Table 1.

Next step is performed in three steps, in the absolute initial step calculate MIU item. The MIU item is determined in Table and the Definition of Minimum utility of item and Minimum utility of itemset is explain in problem definition 6 and problem definition 7. In the second step calculate MAU of item. The MAU item is calculated in Table 7 and the Definition of Maximum utility of item and Maximum utility of itemset is explain in problem definition 8 and. After that Pre-Evaluation Matrix (PEM) is generated. If the kth value in pre evaluation matrix is higher than calculated the minimum utility border value then minimum utility border is set to kth highest value of pre-evaluation matrix. The pre-evaluation matrix is shown in the Table 1. And in the last step SC figuring is performed. Definition 10 explains how to compute the SC of itemset.

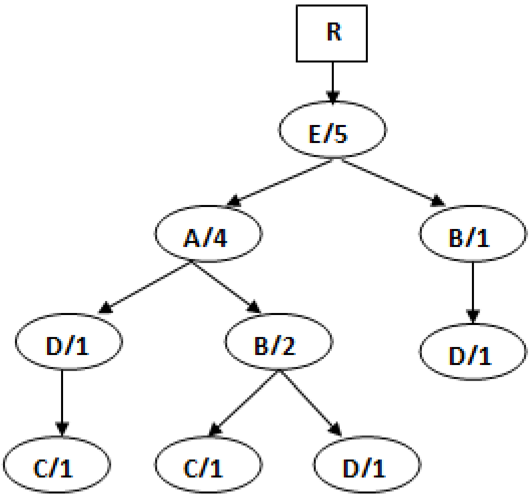


Figure 2: UP-Tree Structure

Table 1: Pre-evaluation Matrix

Item	B	C	D	E
A	33	28	36	73
B		0	16	13
C			0	0
D				0
E				

In the last step, TKO algorithm is executed in only single stage phase and generates top-k high utility itemsets. TKO algorithm uses utility-list structure and whenever an item is produced by the TKO; its utility is compute by utility-list structure without any needs to checks the database

7. Conclusion

The algorithm named TKU and TKO that mines the complete set of TKHUIs from a transactional database without any need to edge esteem threshold value has been proposed. The TKU and TKO algorithm work proficiently on incremental database and accomplish great versatility by utilizing diverse parameters like time and database size. It also decreases search space, memory uses and multiple database scan. TKU algorithm uses UP-Tree for keeping the information of high utility itemset. UP-Tree is constructed in two database scans. UP-Tree data built calculates the Transaction-weighted utility to upgrade the proficiency for finding high utility itemsets. UP-Tree uses various strategies for removing the irrelevant information from transaction database. TKO algorithm is executed in only just a single stage for finding top-k high utility itemsets from a transaction database.

References

[1] Vincent S. Tseng, Cheng-wei Wu, Philippe Fournier-Viger and Philip S. Yu, “Efficient Algorithms For Mining Top-K High Utility Itemsets”, In IEEE Transactions on Knowledge and Data Engineering, vol.28 .

[2] Y. Liu, W. Liao, and A. Choudhary, “A two-phase algorithm for fast discovery of high utility itemsets”, In Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, Vol. 3518, pp. 689-695, 2005.

[3] R. Agrawal and R. Srikant, “Fast Algorithms For Mining Association Rules”, In Proceedings of the 20th International Conference on Very Large Data Bases, Santiago, Vol. 1215, pp. 487-499, 1994.

[4] Serin Lee, Jong Soo Park, “Top-K High Utility Itemset Mining Based On Utility List Structures”, In Proceedings of IEEE International Conference on Data Mining (ICDM), Maebashi, pp. 101 - 108, 2016.

[5] S.Shankar, T.P.Purusothoman, S. Jayanthi, N.Babu, “A fast algorithm for mining high utility itemsets”, In Proceedings of IEEE International Advance Computing Conference (IACC 2009), Patiala, India, pp.1459-1464.

[6] T. Vinothini, V.V.Ramya Shree, “Mining High Utility Itemsets from Large Transactions using Efficient Tree Structure”, In International Journal of Latest Trends in Engineering and Technology, vol. 6,2016.

[7] Shuning Xing, Fangai Liu, Jiwei Wang, Lin Pang, Zhenguo Xu, “Utility Pattern Mining Algorithm Bases On Improved Utility Pattern Tree”, In 8thinternational Symposium On Computational Intelligence and Design, pp.258 – 261,2015.

[8] Junqiang Liu, Benjamin C.M. Fung, “Mining High Utility Patterns in One Phase without Generating Candidates”, In IEEE Transaction Knowledge in Data Engineering, vol. 10, no. 12, pp. 1-14, Dec.2015.

[9] Hong Yao, Howard J. Hamilton,” Mining itemset utilities from transaction databases”, In Science Direct on Knowledge and Data Engineering, 59 (2006) 603–626.

[10] P. Yamini, Soma Shekar, J.Deepthi “Efficient Algorithms for Mining Top-K High Utility Itemsets”, International Journal of Computational Science, Mathematics and Engineering Volume-4-Issue-2-February-2017.

[11] Yogita Khot, Manasi Kulkarni,” Survey on High Utility Itemset Mining from Large Transaction Databases”,In International Journal of Emerging Trends & Technology in Computer Science (IJETTCS).

[12] Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer,Byeong-Soo Jeong, and Young-Koo Lee”In IEEE Transactions On Knowledge And Data Engineering, Vol. 21, No. 12, December 2009.

[13] Prajakta R. Padhye, R. J. Deshmukh," A marketing solution for cross-selling by high utility itemset mining with dynamic transactional databases" In 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT).

[14] Vid Podpecan, Nada Lavrac, and Igor Kononenko," A Fast Algorithm for Mining Utility-Frequent Itemsets".

[15] Jing Wang, Lei Zhang, Guiquan Liu, Qi Liu and Enhong Chen," On Top-K Closed Sequential Patterns Mining",In 2014 11th International Conference on Fuzzy Systems and Knowledge Discovery.

[16] D.Sathyavani, D.Sharmila," Efficient Algorithm for Finding High Utility Itemsets from Large Transactional Databases Using R-Hashing Technique",In International Journal of Innovative Research in Computer and Communication Engineering, Vol. 2, Special Issue 3, July 2014.

[17] Feng Tao, Fionn Murtagh, Mohsen Farid," Weighted Association Rule Mining using Weighted Support and Significance Framework",

[18] R.Nandhini, Dr.N.Suguna," Shrewd Technique for Mining High Utility Itemset via TKU and TKO Algorithm",In International Journal of Computer Science and Information Technologies, Vol. 6 (6) , 2015, 5261-5264.

[19] Snehal D. Ambulkar, . Prashant N. Chatur," Efficient Algorithms for mining High Utility Itemset", In 2017 International Conference on Recent Trends in Electrical, Electronics and Computing Technologies.

[20] Philippe Fournier-Viger,"An Introduction to Frequent Pattern Mining", on http://data-mining.philippe-fournier-viger.com/introduction-frequent-pattern-mining/10/2013.

Author Profile

Ujwala M. Patil has completed her master of technology in Computer Engineering, Dr. Babasaheb Ambedkar Technological University, Lonere, Raigad, Maharashtra, India in 2007 and pursuing her Ph.D. in Computer Engineering from North Maharashtra University, Jalgaon, Maharashtra, India. She is working as an Associate Professor in the Computer Engineering Department at R.C. Patel Institute of Technology, Shirpur (Maharashtra), India. She has 14 years of teaching experience. Her research interests lie in machine learning, web usage mining, data mining, and their applications.

Pramod Pardeshi has completed his Bachelor of Engineering in Computer Engineering from North Maharashtra University, Jalgaon, Maharashtra, India in 2010 and pursuing his master of Engineering in Computer Engineering from North Maharashtra University, Jalgaon, Maharashtra, India.