Association Mining for Super Market Sales using UP Growth and Top-K Algorithm

Harshal Bhope[1,*], *Yash* Mahajan[2,**], *Swapnil* Deore[3,***], and *Vimla* Jethani[4,****]

[1]Ramrao Adik Institute of Technology,Nerul,Navi Mumbai
[2]Ramrao Adik Institute of Technology,Nerul,Navi Mumbai
[3]Ramrao Adik Institute of Technology,Nerul,Navi Mumbai
[4]Ramrao Adik Institute of Technology,Nerul,Navi Mumbai

Abstract. Frequent itemsets(HUIs) mining is an evolving field in data mining, that centers around finding itemsets having a utility that meets a user-specified minimum utility by finding all the itemsets. A problem arises in setting up minimum utility exactly which causes difficulties for users. By setting minimum utility underneath average, too many incessant itemsets will be generated, which in turn will make the mining process quite inefficient. No frequent itemsets will be found if the minimum utility is set too huge. The research focuses on generating frequent itemsets by using the transaction weighted utility of each product. While using UP growth methodology for discovering high utility items from large datasets it takes more time and consumes more memory due to which it is less efficient. So to overcome these drawbacks of UP growth we use the Top-K algorithm which makes it more scalable and efficient. Therefore, we use the Top-K algorithm which does not require a minimum threshold.

1 Introduction

In day to day transactions, High utility mining is used for generating frequent items and existing patterns in user database. Predefined thresholds for measures like support and confidence are used for detection of user interestedness in utility mining. The utiity value of each item is not appropriately determined due to use of these approaches. In business world real time request and frequent itemset mining is in trend these days. Algorithms like aproiri and FP growth seem quite familiar in the transactional system. The above stated methodologies states the frequent items without taking acquisition weights of item into consideration. Finding higher utilities in business decision plays a vital role and helps in earning profits.

Business analytics in research stresses more on FIM these days. The sets with low prioritized value which are frequently repeated are viewed in huge amount by major FIM approaches. The loss of data occurs due to sets with less selling frequencies. Therefore, it can't fulfill client necessity who needs high utilization, for example, more benefits. It comprises of amount as item utility and unit benefit as number of event tally in every dataset. As per client determination the utilization of set constitute to as far as transaction weighted utility, tally , limit or other information. On the other side,it is possible that utility of HUI is more than client minimum value, at that point sets are called as high utility item set and accordingly it has critical job in market analysis.

Mining high frequent items prompts the revelation of connection among items in huge value based or social datasets. With large amount of information being constantly gathered and saved numerous industries are getting keen on mining such examples from their databases. The disclosure of compulsive connection among large amount of transaction record can help in numerous business decision making procedure, for example, customer shopping behaviour analysis. The disclosure of these fascinating relation can assist retailers with creating business advertising systems by picking up information into which items are habitually bought together by clients.

2 Literature Survey

Ameena Aiman and Raafiya Gulmeher[1],"Efficient Algorithms for Mining Top-K High Utlity Itemsets" They proposed TKU(Top-k Utility Itemsets) and TKO(Top-K utility itemsets in one phase) algorithms without the need to set minimum utility.

Myneni Madhu Bala and Rahit Dandamudi[2],"Efficient High Utility Pattern Mining Algorithm for E-Business" by analysis of market behaviour and customer interests of transactional data. This technique uses UP growth strategy(UP tree) and TKU method and discovering top-k utility in one phase approach with TKO method to mine frequent without any presumptions of minimum limit threshold which increases performance and scalability.

*e-mail: harshalbhope67@gmail.com
**e-mail: mahajanprajwal@gmail.com
***e-mail: mahajanswapnil31@gmail.com
****e-mail: vimlajethani@gmail.com

Serin Lee and Jong Soo Park[5],"High Utility Itemset Mining based on Utility-List Structures". They proposed a new strategy TKUL-miner, to draw top-k high utility items efficietly. It utilizies another utility list structure which stores fundamental data at every hub(node) on the queue tree for mining the itemsets. This strategy has a technique utilizing search request for specific region to raise the outskirt least utility threshold value.

V.Kavitha and B.G.Geetha[6],"Review on High Utility Itemset Mining Algorithms". They have discussed a comparative study of three mining approaches such as FHM, HUI Miner and Two phase miner.

Junqiang Liu, Ke Wang and Benjamin C.M. Fung[7],"Mining High Utility Patterns in one phase without generating candidates". This paper introduces Utility Pattern Growth approach. This model developement approach is to draw a reverse set enumeration tree and to prune search tree by utility upper bounding.

3 Proposed Work

In this research, we are discussing different kind of data mining approaches, that is, Top-K(TKU and TKO) and UP growth algorithms in order to obtain frequent itemsets from large datasets. The first stage involves TKO Algorithm, the process of generating Utility List Structure using Transaction Utility(TU) and Transaction Weighted Utility(TWU)[3]. After generating utility list structure, we obtain the base value called Threshold. The second involves the TKU Algorithm, which is the process of constructing UP Tree and generates PKMUIs. At last we obtain two tables, Minimum Utility of an Item Table(MIU) and Maximum Utility of an Item Table(MAU), which provides us the range to generate frequent Itemsets.

So,we are doing the comparative study of two algorithms which are Top-k and UP growth.

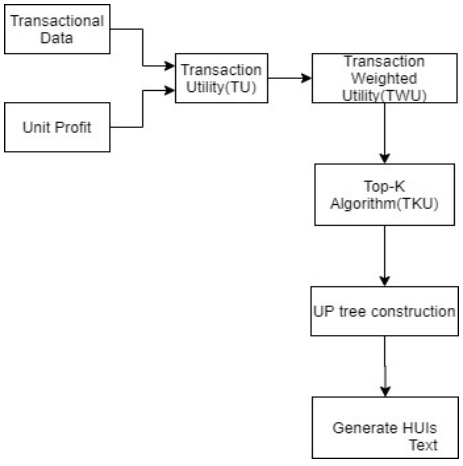

Figure 1. Flow Diagram

4 Methodology

4.1 UP Growth

We outline the feature of UP-growth for effectively producing PHUIs(Potentially high utility itemset) from the UP-tree with two techniques, in particular DLU (Discarding nearby unpromising thing) and DLN (Decreasing neighborhood hub utilities). By utilizing these two strategies, the unpromising set of items having least utility are disposed of from utility at the instant of building a basic UP-Tree. The general methodology of mining utility example is to count every subset of and text if X has utility over the limit. Anyway a comprehensive count is infeasible because of large number of subsets of J and thus it is analytical to utilize tough pruning procedures.

Transaction Utility(TU):

$$\sum_{i\epsilon n} unit\ \ profit(t) * quantities(c)$$

Transaction weighted Utility(TWU):

$$\sum_{i=1}^{n} TU$$

Full Prefix Expression of an Item(fpe):

$$\sum u(fpe(X,t)t)$$

Let J be the set of items. Let B be a sets of transactions (t1, t2, t3,..tn) where each transaction tj belongs to J. Each item in a transaction is allocated a non-zero value. Each clear-cut item has a benefit self-sufficient of any transaction, given by an Internal and External Utility Table (UT). The issue is to discover all high utility patterns.

Table 1. Transaction Table

Transactions		
TID	**Items**	**Product Utility**
T1	(k,2),(m,2),(o,2)	12
T2	(k,7),(l,3),(m,3),(p,5)	38
T3	(k,2),(l,2),(m,3),(n,3),(o,6), (q,6)	57
T4	(k,4),(l,3),(n,4),(o,3)	35
T5	(k,3),(l,2),(n,3),(p,3)	25

Consider the information or model of a supermarket.Quantity ,that is, value of each product is listed in Table 1 for each shopping transaction where J= k,l,m,n,o,p,q and B= t1, t2, t3, t4, t5 and Table 2 lists the profit of each product. For transaction t2 = k, l, m, p, we have iu(k, t2)= 7, iu(l,t2)= 3, iu(m,t2)= 3, iu(p,t2)= 5, eu(k)= 1, eu(l)= 2, eu(m)= 5, and eu(p)= 2. Here, u(j, t) is the product of iu(j, t) and eu(j). Thus, u(k, t2)= 7, u(l,t2)= 6, u(m,t2)= 15, u(p,t2)= 10, and so on.

Now, we will construct the tree by considering count(s), utility(u), TWU, Ufpe respectively. As, it is not

Table 2. Items with Profit

Item	Profit
K	1
L	2
M	5
N	4
O	3
P	2
Q	1

Table 3. Utilities(minU=45)

utility table for root node				
Item	**count(s)**	**u**	**twu**	**ufpe**
K	5	18	167	18
L	4	20	155	36
M	3	40	107	61
N	3	40	105	78
O	3	33	104	86
P	2	16	63	55
Q	1	6	57	57

possible to show entire tree together because it requires wide space. So we are going node by node.

Step 1:- By taking threshold=45,we will compare utility of full prefix expression(Ufpe) of each node with threshold. If Ufpe>threshold, then the nodes have greater utility as compared to threshold will consider that nodes such as nodes M,N,O,P,Q have Ufpe>30.

Step 2:- Here, First we are going with Node M.

Table 4. After comparing with threshold expanding node have ufpe>threshold

Node M				
Item	**count(s)**	**u**	**twu**	**ufpe**
K	3	51	61	44
L	2	49	49	49

Step 3:- Compare Ufpe values of K and L with threshold. As Ufpe(B)>30. Therefore,

Table 5. K is High Utility Item

Item	count(s)	u	twu	ufpe
K	2	49	49	49

We will repeat the above process until we get the high utility itemset for all node which have ufpe greater than minU(threshold).

4.2 Top-K

The proposed algorithm improves the efficiency of the process by reducing the low utility items. The enormous value-based database is taken by applying TKU strategy. TWU is determined depending on unit benefit and amount. At some point, large tallies of each set is recognized, and UP tree is developed dependent on Top-K high utility item set. A top-k high utility mining strategy works as: First it sets an central minimum utility to 0, and begins to traverse the seek space. At some point, when k high utility itemsets are discovered, the central minimum utility is raised to the utility of the example having the most minimal utility among the present top-k models. Then, the seeking proceeds and for every high utility itemset found, the set of the present top-k sample is revised just like the minimum limit.

Transaction Utility(TU):

$$\sum_{i\epsilon n} unit\ \ profit(t) * quantities(c)$$

Transaction weighted Utility(TWU):

$$\sum_{i=1}^{n} TU$$

Table 6. Transaction Table

Transactions		
TID	**Items**	**Product Utility**
T1	(k,2),(m,2),(o,2)	12
T2	(k,7),(l,3),(m,3),(p,5)	38
T3	(k,2),(l,2),(m,3),(n,3),(o,6),(q,6)	57
T4	(k,4),(l,3),(n,4),(o,3)	35
T5	(k,3),(l,2),(n,3),(p,3)	25

Table 7. Items with Profit

External Utility	
Item	**Profit**
K	1
L	2
M	5
N	4
O	3
P	2
Q	1

In table 8, we have shown calculated TWU of items with their respective count.

Table 8. Transaction Weighted Utility

TWU		
Item	**Count**	**TWU**
K	5	167
L	4	155
M	3	107
N	3	105
O	3	104
P	1	63
Q	1	57

Now, we construct UP tree taking first node as a root node depending upon count of items and transaction wise.

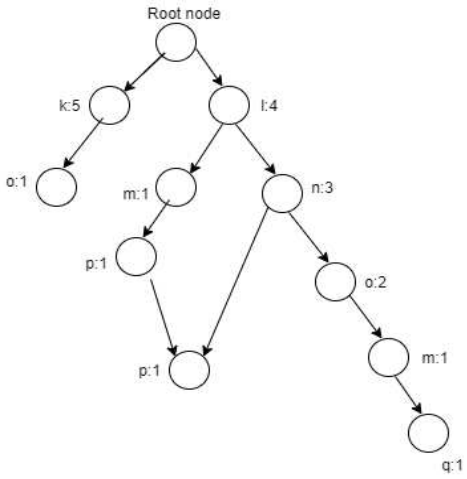

Figure 2. UP tree based on counts of respective items transaction wise.

	A	B	C	D	E
1	voiceNo	Description	Quantity	UnitPrice	CustomerID
2	536365	WHITE HANGING H	6	2.55	17850
3	536365	WHITE METAL LAN	6	3.39	17850
4	536365	CREAM CUPID HEA	8	2.75	17850
5	536365	KNITTED UNION FL	6	3.39	17850
6	536365	RED WOOLLY HOTT	6	3.39	17850
7	536365	SET 7 BABUSHKA N	2	7.65	17850
8	536365	GLASS STAR FROST	6	4.25	17850
9	536366	HAND WARMER UN	6	1.85	17850
10	536366	HAND WARMER RE	6	1.85	17850
11	536367	ASSORTED COLOUF	32	1.69	13047
12	536367	POPPY'S PLAYHOUS	6	2.1	13047
13	536367	POPPY'S PLAYHOUS	6	2.1	13047
14	536367	FELTCRAFT PRINCE	8	3.75	13047
15	536367	IVORY KNITTED MU	6	1.65	13047
16	536367	BOX OF 6 ASSORTE	6	4.25	13047

Figure 3. Optimal Attributes of SuperMarket Datasets[8]

Table 9. Comparison of UP and Top-k

Comparison		
Parameters	**UP Growth**	**Top-K**
Method	1.Construction of UP tree. 2.Generates Potential High Utility Itemsets. 3.Identify high utility items using PHUI values.	1.TKO(Top-K in One phase). 2. TKU(Top-K Utility Patterns)
Strategy	1.Discarding Global Unpromising Item(DGU Strategy) is used for contruction of UP tree. 2.Discarding Global Node Utilities(DGN strategy)	1.TU and TWU are calculated . 2.UP tree is constructed in TKO. 3.TKU is used to generate High Utility items.
Time Taken	More as compared to Top-k.	Less as compared to UP growth.
Memory Consumption	More as compared to Top-K.	Less as compared to UP growth.

5 Results and Analysis

The experimented results were conducted on SuperMarket Datasets. The dataset is collected from online Kaggle datasets repository(https://www.kaggle.com/jihyeseo/online-retail-data-set-from-uci-ml-repo/data)[8]. The attributes taken from the datasets are shown below in figure 3.

The complete evaluation of UP Growth and Top-K(TKU) is with UP tree. To contrast UP tree client has to pick ideal and separate parameters. Super Market is the only dataset which incorporates both buy amount and unit benefit. Here, Super Market dataset has been considered for performance evaluation of UP growth and Top-k(TKU).

Table 10. Time Taken Values

Time		
HUIs	**Top-K**	**UP Growth**
5	0.2635s	0.567s
10	0.187s	0.364s
15	0.094s	0.183s
20	0.109	0.204s

In figure 4, we can see that the two algorithms UP growth and Top-K are compared on the basis of time consumption. In below graph, it visible that Top-K takes less time to generate high utility items as compared to UP growth algorithm. The graph generated shows that with respect to time Top-K is more scalable than UP growth.

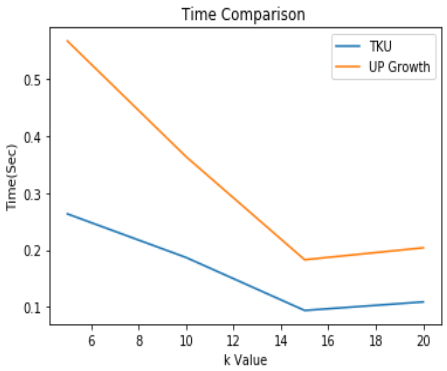

Figure 4. (a)Graph showing comparative study of UP growth and TKU based on Time.

In figure 5, The two algorithms TKU and UP growth are compared on the basis of memory usage. As we can see, UP growth consumps more memory as compared to TKU due to which less efficient.

Table 11. Memory Consumption values

Memory		
HUIs	**Top-K**	**UP Growth**
5	22.7492 MB	22.285 MB
10	29.7471 MB	31.747 MB
15	42.5744 MB	43.310 MB
20	42.59 MB	43.0742 MB

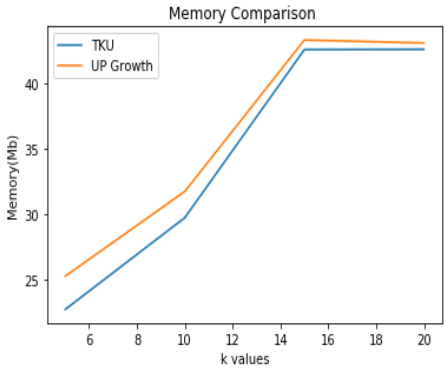

Figure 5. (b)Graph showing comparative study of UP growth and TKU based on Memory Consumption.

6 Conclusion

In this research, the high utility mining issue is resolve by utilization of weighted utility without considering threshold values. For that the proposed strategy Top-K having two periods of execution TKU and TKO. TKU calculation gives UP tree using count and TWU. From the normal outcomes, the proposed method is productive to mine the high utility models with more utilization of items without considering any base utility that is minutil. This assessment of procedure gives great versatile and productive execution on large and dense data.

References

[1] A. Aiman , R. Gulmeher, “Efficient Algorithms for Mining Top-K High Utility Itemsets” at *International Journal of Computer Science and engineering*, July (2018).

[2] M. M. Bala and R. Dandamudi, “ HUPM: Efficient High Utility Pattern Mining Algorithm for E-Business” at *8th International Advance Computing Conference(IACC)*, (2018).

[3] S. D. Ambulkar and Dr. P. N. Chatur,"Efficient Algorithms for mining High Utility Itemset" at *International Conference on Recent Trends in Electrical, Electronics and Computing Technologies* ,(2017).

[4] V. S. Tseng, Cheng Wei Wu, P. Fourier-Viger, and Philip S. Yu, “Efficient Algorithms for Mining Top-K High Utility Itemsets” at *IEEE Transactions On Knowledge And Data Engineering*, January (2016).

[5] S. Lee and Jong Soo Park, "Top-K High Utlity Itemset Mining Based on Utility-List Structues" at *International Conference on Big Data and Smart Computing (BigComp)*, January (2016).

[6] V.Kavitha and Prof. B.G.Geetha."Review on High Utility Itemset Mining Algorithms", at *IEEE Sponsored World Conference on Futuristic Trends in Research and Innovation for Social Welfare*, (2016).

[7] Junqiang Liu, Ke Wang and Benjamin C.M. Fung, “Mining High Utility Patterns In One Phase without Generating Candidates ” at *IEEE Transactions On Knowledge And Data Engineering*, (2015).

[8] https://www.kaggle.com/jihyeseo/online-retail-data-set-from-uci-ml-repo/data.