Implementation and Comparison of Efficient Algorithms for Extraction of Highly Profitable Item sets from Dense Datasets

Miss.Gauri Mathad
ME Student
Modern College of Engineering

Prof.Mrs.Deepti Nirwal
Faculty : Department of Computer Engineering
Modern College of Engineering

Abstract—The extraction of High Utility Item sets (HUI) is the popular problem of data mining.We can say that HUI mining is basically a branch or an extension of frequent pattern mining. As we are already aware of frequent pattern mining and its applications we might further proceed to finding not only frequently purchased materials but also profit gained from these materials.To store the HUI here UP Growth algorithm[9] is used. TKO (Top k in one phase) and TKU algorithms(Top k in Utility phase)[1] are used to acomplish the task of HUI mining. Here the framework is implemented using TKU and TKO so as to reduce the time and space complexity of HUI mining task . It is useful in the applications like online shopping , etc. It is mostly used in analysis of the market basket, where if the customer buys the item the system suggests him other similar items so that he can buy them so as to maximize the benefit for both the customer and supplier.

Index Terms—Utility mining, highly profitable itemsets , top k Pattern mining, top k high utility itemset set mining.

I. INTRODUCTION

A. BACKGROUND

To overcome the above drawbacks Top K models are introduced which consist of two phases. In the first phase, called phase I, potential candidate itemsets are generated which are capable of becoming one of the top k HUIs. In phase II, we calculate the exact utilities of itemsets by scanning the database and finally the exact HUIs are obtained. Here in this traditional approach users had to face the problem of mentioning the value called min utility threshold .Min utility threshold means the minimum profit value which you decide so that the itemsets with the profit greater than min util threshold are obtained. The value of min util threshold decided the number of itemsets generated.If the value was too small the huge number of itemsets were generated if the value was too large the itemsets satisfying the conditions were rarely found. This inturn affected the performance the mining algorithms.

B. MOTIVATION

When the users want to analyze the most profitable itemsets they have to mention the value of min util threshold as discussed previously in traditional apllications . Mentioning min util value is a more difficult task and so to solve this problem the top k algorithms have been designed which will take k (the no. of itemsets as an input).K depends upon database characteristics and so users can easily decide the value of k. Also the difference betwwen traditional HUI mining algorithms and TKO and TKU is that the border minimum utility value increases dynamically in case TKO and TKU whereas it is fixed in case of traditional HUI mining algorithms.

C. OBJECTIVES

To compare two algorithms TKU and TKO and check the effeciency on the basis of their execution time and the results generated.The execution time of Top K in One Phases is less than other algorithms such as the Top K in Utility Phases. But the issue with Top k in one Phase is that though its execution time is less,the results are not accurate.The execution time of TKU algorithm is more but results are accurate. It is very challenging issue. Hybrid algorithm (TKO WITH TKU) is efficient than TKU algorithm. The time factor is very important in these algorithms.Hybrid algorithms achieve significantly better performance and the future objective of this paper is to develop the hybrid algorithm for HUI mining.

II. REVIEW OF LITERATURE

[1] ”Efficient algorithms for mining top k high utility itemsets” : Two algorithms, one phase and two phase algorithms for mining high utility itemsets are introduced.Many definitions for calculation of the utilities of itemsets and strategies for increasing the min util threshold are mentioned. The two algorithms are compared with REPT algorithms which is also a two phase algorithm.One phase algorithm is found to be more efficient in terms of execution time.

[2] ”Efficient tree structures for high-utility pattern mining in incremental databases” : Frequency values of items and profit values of each items are used for calculatingthe utilities of item sets in traditional applications. In this paper incremental and interactive data mining processes are suggested which have the ability to use the data structures and mining results

that were stored previously in order to reduce the time required for calculations whenever the database is updated or whenever the minimum utility threshold is changed.

[3].”Mining high-utility item sets” : One of the drawbacks of association rule mining algorithms is that it
generates a large number of frequent rules which do not provide useful information. Hence in this paper more efficient method is introduced which takes into consideration utilities of patterns so as to capture the highly desirable patterns which presents a level-wise mining algorithm.

[4] ”Mining top-k frequent closed patterns without minimum support” :The procedure for top-k frequent closed pattern mining is introduced .Here an important concept i.e min utility is introduced which is minimum lengt of each pattern.TFP is an algorithm used to fulfill the task of top-k frequent closed pattern mining without mentioning the minimum support.

[5]. ”Mining frequent patterns without candidate Generation” : Candidate sets are are mainly generated
in algorithms like Mining frequent patterns in transaction databases, times Series databases etc. Mining cost is increased due to candidate set generation due to the existence of long patterns.So Frequent pattern tree structure is introduced here which is used for storing compressed, crucial information
about frequent patterns and helps in developing FP tree based mining method called FP-growth.

[6] ”Novel Concise Representations of High Utility Item sets Using Generator Patterns” : In some applications,the set of HUIs can be very large. This may cause long execution times and will lead to huge memory consumption. This issue can be solved using the technique called the concise representations of HUIs.

III. METHODOLOGY

To Get High Utility item Set here TKO and TKU[1] algorithms with Parallel Mining are used.In this system the goal is to extract the series of elements without the need to establish a utility minimum for which two types of calculations have been introduced TKU and TKO.TKO algorithm is fast because it is one phase algorithm but it has the main disadvantage of not mainly accumulating the result.So it returns the garbage value in some cases in the set of high utility items. The result of the TKU algorithm is accurate but the execution time is high[1], so the alternative solution is to find the efficient algorithm in the proposed combination of the TKO and TKU algorithms system. But in this framework currently the two methodologies have not been combined and we just compare the performances of TKO and TKU algorithms individually.

Dataset used :
The dataset used is a product dataset.The different features of product dataset are product class id,product id,brand name,product name etc. It is the database consisting of information about food products of different brands.

Link :
https://github.com/demidovakatya/mashinnoyeobucheniye/blob/master/4-stats-for-dataanalysis/
week 3/foodmart.products.tsv

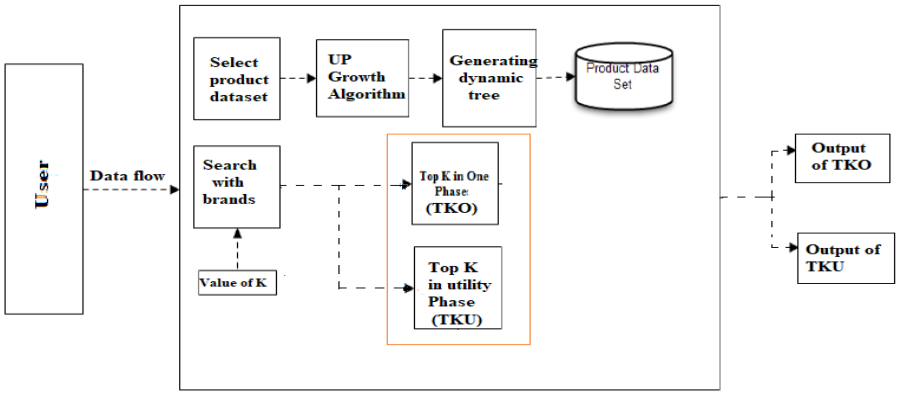
	A	B	C	D	E	F	G
1	id	brand_name	product_name	SKU	SRP	gross_weirth	net_weigth
2	1	Washington	Washington Berry Juice	90748583674	2.85	8.39	6.39
3	2	Washington	Washington Mango Drir	96516502499	0.74	7.42	4.42
4	3	Washington	Washington Strawberry	58427771925	0.83	13.1	11.1
5	4	Washington	Washington Cream Sod	64412155747	3.64	10.6	9.6
6	5	Washington	Washington Diet Soda	85561191439	2.19	6.66	4.65
7	6	Washington	Washington Cola	29804642796	1.15	15.8	13.8
8	7	Washington	Washington Diet Cola	20191444754	2.61	18	17
9	8	Washington	Washington Orange Juic	89770532250	2.59	8.97	6.97
10	9	Washington	Washington Cranberry J	49395100474	2.42	7.14	5.13
11	10	Washington	Washington Apple Juice	22114084362	1.42	8.13	7.13
12	11	Washington	Washington Apple Drin	17074288725	3.51	20	19
13	12	Jeffers	jeffers Oatmeal	49031038880	1.54	8.9	6.89
14	13	Jeffers	Jeffers Corn Puffs	13229009509	2.65	10.4	7.39
15	14	Jeffers	Jeffers Wheat Puffs	92942813038	1.93	21.6	20.6
16	15	Jeffers	Jeffers Grits	26378549933	2.29	21.3	20.2
17	16	BlueLabel	Blue Label Canned Beet	62908702492	3.83	21.2	18.2
18	17	BlueLabel	Blue Label Creamed Co	79484335780	2.9	6.91	3.9
19	18	BlueLabel	Blue Label Canned Strin	85252254605	2.68	12.6	10.6
20	19	BlueLabel	Blue Label Chicken Sou	47163524031	3.19	15.2	12.1
21	20	BlueLabel	Blue Label Canned Yam	22169209122	2.78	21.3	19.2
22	21	BlueLabel	Blue Label Vegetable Sc	43318244814	2.33	19.7	18.7

Table1

Fig. 1. Product Data Set

Advantages of System:
1. No need to set the minimum utility threshold value.
2. The algorithms have less search space so they need less memory.
3. They scan the database only once unlike other HUI mining algorithms.
4. They are easy to implement.
5. Their performances are good in dense datasets

A. Architecture



Explanation:
Here once the user logs in into his account he will first have to select the brand. Various items of that particular brand will be displayed. User can just explore the items or buy the items. If the user just explores the item i.e if he just clicks on the item and does not buy the item then that item's hit parameter will increase by one. If the user will buy that item then the item's buy parameter will increase by one. All the information about the items that a particular user bought or explored is stored in Dynamic tree generated by UP Growth algorithm. In the second Module i.e Search brands user can search for
top K best items. He just have to select the category i.e brand name then mention the parameter by which top K items have to be discovered i.e Buy rate or hit rate and then mention the K value i.e the number of items to be displayed. Then accordingly the TKO and TKU algorithms will generate the TOP K itemsets.

B. Algorithms

1.) Top K in one phase (TKO) :

Min-util-border is first set to zero and min heap structure is initialized for maintaining HUI during search.Database is scanned twice to build initial utility lists. Then TKO algorithm is executed which is the combination of RUC(Raising threshold by Utility of candidates)[1] and HUI Miner search procedure.During the search as top k list is updated the value of min-util-border threshold rises.On termination complete set
of top k HUIs is captured from the database.

Steps in TKO algorithm :
Input :
1) util(P) : utility list for prefix P;
2) C[P] : a set of itemsets w.r.t the prefix P;
3) ULS[P] : a set of utility list w.r.t the prefix P;
4) _ : Border minimum utility threshold ;
5) TOP-k-list : a list for storing itemsets ;

Results :
All the top-k HUIs are stored in TOP-k-list.

Procedure :
Step 1 : if the itemsets X=x1,x2,x3.... belong to C[P] then,
Step 2 : Calculate the sum of utilities of all the items in transaction and check whether it is greater than or equal to min util threshold.
Step 2.1 : if sum of utilities is greater than min util threshold then, raise the value of min util threshold using RUC strategy.
Step 3 : if sum of sum of utlitities and sum of remaining utilities of itemset is greater than min util threshold then,
Step 3.1 : Set C[X]=null and ULS[X] =null
Step 3.2 : For itemset Y=y1,y2,y3...yn belonging to Class [P] and yL _ xL do,
Step 3.2.1 : Z = x U Y;

Step 3.2.2 : ul(Z)= constructed the utility list;
Step 3.2.3 : concatenate the elements of C[X] with Z and store in C[X];
Step 3.2.3 : concatenate the set of utility lists ULS[X] utility list of z i.e ul(Z);
Step 4 : Recursively call this procedure by giving X,ULS[X] , C [X] , min util threshold , Top-k-CI-list as parameters.

2.) Mining Top K utility itemsets (TKU) :

TKU is a two phase algorithm used to accomplish then task of HUI mining without the need of specifying min util threshold. In the first phase TKUbase algorithm is executed which is derived from tree based algorithm called UPGrowth which maybe used for maintaining the information about transactions. Steps involved in TKU base are :
1.)Construction of UPTree
2.)Generation of potential top-k high utility itemsets from UP-Tree

Procedure for construction of UPTree :

We use header table for traversal of the UP-Tree. Nameof the item ,its estimated utility value and a link are the features of header table. The first node of UP-tree consist of the link pointing the item name which is same as item name at the entry. UPtree is constructed by scanning the entire database twice. In the first scan transaction utilities of transactions and and utilities each items are computed. After computing the utilities items are stored in the header table in descending order of their utility values. From the header table transactions are recognized and are inserted in UPTree structure during second database scan.

Steps in TKU algorithm:-
Input : 1)A database D and 2) The number of desired HUIs k;
Output : The complete set of PKHUIs C;
Steps :-
1.) Initialize min util border to 0.
2.) An UPTree is constructed by double scanning of database.
3.) PKHUIs are generated by applying the UPGrowth procedure.
4.) For each generated PKHUI with estimated utility ESTU(X) do
5.) If (ESTU(X) _ min util Border and MAU(X) _min util Border)
6.) Produce Output X along with min ESTU(X),MAU(X) ;
7.) If(MIU(X) _ min-utilBorder)
8.) Raise min-util Border by the strategy MC;
10.) Stop

V. RESULTS AND DISCUSSION

TKO has the best performance as compared to TKU. As TKO is one phase algorithm and TKU is two phase algorithm, execution time taken by TKO is much less than TKU which can be proved from the graph. In case of TKO as the value of K increases, the execution time decreases but in case of TKU as the value of k increases the execution time increases.This is because TKO uses RUC and RUZ strategies which avoids generation of low utility items. TKU cannot effectively raise the border minimum utility threshold value and suffers from very long runtime on dense datasets.

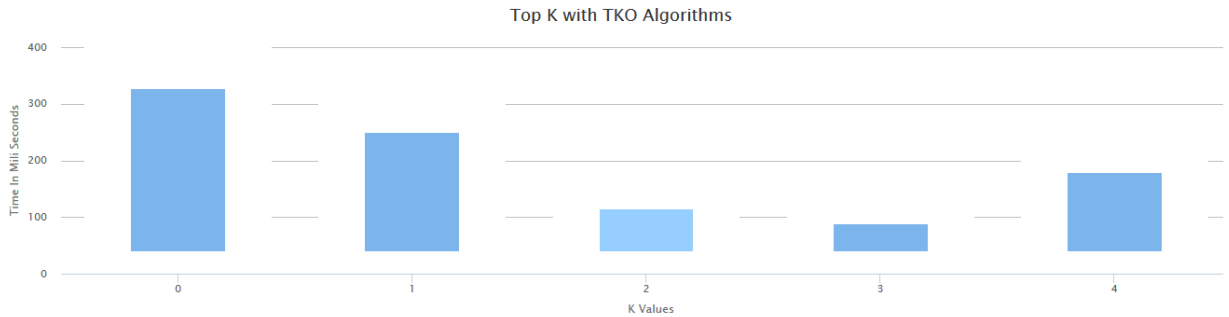


Fig. 5. Graph of TKO algorithm for different K values

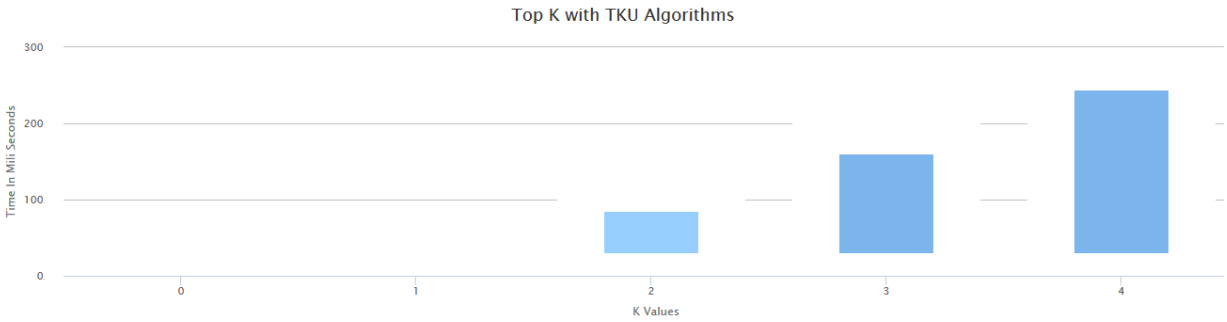


Fig. 6. Graph of TKU for different K values

VI. CONCLUSION

Hence , the paper focuses on HUI mining and the two algorithms which are implemented for the same.The main goal was to develop an application for HUI mining which eliminated the problem of mentioning the min util value. Instead users have to just mention the value of k which represents the number of itemsets according to the user's wish. Here an online foodmart application is developed which will return the set of highly profitable food items on the bases of hit and buy parameters which will be considered as utility values of itemsets.Though TKO and TKU algorithms have many advantages over traditional HUI mining algorithms, there are some drawbacks associated with both algorithms and to eliminate these drawbacks , the future work would be combining the two algorithms and creating a hybrid algorithm which will posses both, the accuracy of TKU and time efficiency of TKO.

REFERENCES

[1] Vincent S. Tseng, Senior Member, IEEE, Cheng-Wei Wu, Philippe Fournier-Viger, and Philip S. Yu, Fellow, IEEE,, "Efficient Algorithms for Mining Top-K High Utility Itemsets",IEEE transactions on knowledge and data engineering, VOL. 28, NO. 1, Jan 2018.

[2] C. Ahmed, S. Tanbeer, B. Jeong, and Y. Lee, Efficient tree structures for high-utility pattern mining in incremental databases, IEEE Trans. Knowl. Data Eng., vol. 21, no. 12, pp. 17081721, Dec. 2009.

[3] R. Chan, Q. Yang, and Y. Shen, Mining high-utility itemsets, in Proc. IEEE Int. Conf. Data Mining, 2003, pp. 1926.

[4] 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. 211218.

[5] J. Han, J. Pei, and Y. Yin, Mining frequent patterns without candidate generation, in Proc. ACM SIGMOD Int. Conf. Manag. Data, 2000, pp.112

[6] P. Fournier-Viger, C. Wu, and V. S. Tseng, Novel concise representations of high utility itemsets using generator patterns, in Proc. Int. Conf. Adv. Data Mining Appl. Lecture Notes Comput. Sci., 2014, vol. 8933, pp.
3043.

[7] J. Liu, K. Wang, and B. Fung, Direct discovery of high utility itemsets without candidate generation, in Proc. IEEE Int. Conf. Data Mining, 2012, pp. 984989.

[8] Y. Lin, C. Wu, and V. S. Tseng, Mining high utility itemsets in big data, in Proc. Int. Conf. Pacific-Asia Conf. Knowl. Discovery Data Mining, 2015, pp. 649661
.
[9] V. S. Tseng, C. Wu, B. Shie, and P. S. Yu, UP-Growth: An efficient algorithm for high utility itemset mining, in Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2010, pp. 253262.

[10] Y. Li, J. Yeh, and C. Chang, Isolated items discarding strategy for discovering high-utility itemsets, Data Knowl. Eng., vol. 64, no. 1, pp. 198217, 2008.

[11] T. Quang, S. Oyanagi, and K. Yamazaki, ExMiner: An efficient algorithm for mining top-k frequent patterns, in Proc. Int. Conf. Adv. Data Mining Appl., 2006, pp. 436 447.