International Journal of Computer Applications (0975 – 8887)
Volume 165 – No.3, May 2017

High Utility Itemset Mining with Top-k CHUD (TCHUD) Algorithm

Anu Augustin
P.G. Scholar
Sahrdaya College of Eng. & Technology, Kodakara

Vince Paul, PhD
H.O.D,Computer Science Dept.
Sahrdaya College of Eng. & Technology, Kodakara

Vishnu G. Nair
Asst. Professor,Computer Sc. Dept.,Sahrdaya College of Eng.
Vishnu G Nair

ABSTRACT

High utility itemset mining is an uncommon term. But we are using it while we are doing online purchases etc. It is a part of business analytics. Its main application area is market basket analysis where when a customer purchases an item he can buy another item to maximize profit. So both the customer and business vendors earn profit. This one is not a new concept and is derived from frequent itemset mining. Here we proposes an algorithm for mining closed high utility itemset using top-k algorithm. So that execution time will be less and space efficiency can also be achieved. Both the concept of closed high utility itemset and top-k mining are existing. The new concept is that integrating the merits of them together. The algorithm used for closed hui mining is CHUD.Similarly the algorithm used for top-k mining is TKU,TKO etc. Also recovering all HUIs from complete set of CHUIs using DAHU algorithm.

General Terms

Frequent Itemset Mining, High Utility Itemset, Closed High Utility Itemsets,Top-k Mining, Transaction Utility.

Keywords

Minimum Utility Threshold, CHUD, Top-K, TWU, Support count.

1. INTRODUCTION

Data mining is the efficient discovery of valuable and vivid information from a large collection of data.. There are various subdivisions for it. Frequent itemset mining(FIM) discover only frequent items. But profit of the items are not considered. This is because purchase quantity not taken into account, all items viewed as having same importance and also discover frequent patterns that are not interesting. An item can be present or absent in a transaction. The commonly used algorithms are Apriori, Eclat , LCM, Pre Post, FIN and FP-Growth algorithm. Frequent itemset mining(FIM) doesn't satisfy the requirement of the customer and seller. Certain association rules are associated with this mining. For example those who buy bread and butter are likely to buy milk too. In FIM computational cost is low but miss more important patterns to users.

High utility mining is an extension to the problem of frequent pattern mining. High utility mining means identifying itemsets having high profits when they are sold together. Here the utility of an itemset is measured in terms of weight , profit , cost, quantity or other information depending on the user preference. The profit generated and transaction count is taken into account. A transaction database means a database with a list of transactions like butter,bread,milk etc. In High Utility Itemset (HUI) mining the itemsets that generate a profit higher than minimum threshold is called as high utility itemsets.So there will be a transaction database and a unit profit table for each of the items in the transaction database. The hui mining is interesting because of two main reasons. One is that discover itemsets that generate a high profit in customer transactions than those that are bought frequently. Also no Apriori property or anti-monotone property in hui mining. The anti-monotone property is that if an itemset is infrequent then all its superset also infrequent and can be pruned. To overcome this the concept of transaction weighted utilization(twu) introduced. Since there is no anti-monotone property inefficient in terms of time and memory requirement or even run out of memory.

Different algorithms for hui mining are UP-Growth,Up-Growth+,IHUP,IIDS,d^2HUP,HUI-Miner etc. The algorithms for hui mining can be two or one phases.Hui mining takes more time and space for execution. So incorporated a concept called closed hui mining. An itemset is said to be closed if there exists no other itemset such that the first one is the subset of other and also there support count should not be equal. The various algorithms used are CHUI-Miner, CLS-Miner[15] etc.

Setting the threshold is a tedious job in hui mining. If the threshold chosen is not a proper value there is a chance of missing some items or itemsets.Top-k high utility itemset mining overcomes this difficulty. In top-k hui mining, k is the desired number of HUIs to be mined. So only less values have to be stored saving time and memory. There are various algorithms for top-k mining like TKU,TKO etc. A few top-k frequent pattern algorithms are BTK(TB-tree)[13], TFP [14](Top-k Frequent Pattern) etc. In top-k no need of setting the threshold, instead it is initially set to zero and raise automatically. Using top-k we can easily find the top-k sets of products that contribute the highest profits to the company. The top-k can be applied to CHUD algorithm after phase I to decrease the number of candidates stored so that storing only top-k values from the IT-tree.IT-tree is used in CHUD algorithm. Here the k is the minimum utility threshold specified by us. Only top-k values are stored based on the minimum utility threshold.

We are unaware about the applications of high utility mining. It includes streaming analysis, market analysis, mobile computing, biomedicine. Clickstream analysis is the process of looking at clickstream data for market research or other purposes. A clickstream is a rendering of user activity on a website, where a user clicks on a computer display screen and how that movement translates to other web activity. Clickstream analysis is also known as clickpath analysis. Web site is different and their occurrences are not limited to a 0/1 value[23].The main point of clickstream tracking is to give webmasters insight into what visitors on their site are doing[11]. Analyzing the data of clients that visit a company website can be important in order to remain competitive. This analysis can be used to generate two findings for the company, the first being an analysis of a user's clickstream while using a website to reveal usage patterns, which in turn gives a heightened understanding of customer behavior. This use of the analysis creates a user profile that aids in understanding the types of people that visit a company's website. A clickstream analysis can be used to predict whether a customer is likely to purchase from an e-commerce website. Clickstream analysis can also be used to improve customer satisfaction with the website and with the company itself. This can generate a business advantage, and be used to assess the effectiveness of advertising on a web page or site[12].This is performed by counting the keystrokes.

M-commerce (mobile commerce) is the buying and selling of goods and services through wireless handheld devices such as cellular telephone and personal digital assistants (PDAs).Here we use utility mining algorithms for extracting the knowledge data from the data owners when the knowledge consumers are in need of a particular knowledge data. Utility Mining as a service in a distributed computing environment which can be applied in business such as cross selling approach. This novel architectural based approach is experimented in online shopping cart system. Also used in different genetic algorithms. Therefore, utility mining represents real world market data. By utility mining, several important business area decisions like maximizing revenue or minimizing marketing or inventory costs can be considered and knowledge about itemsets/customers contributing to the majority of the profit can be discovered. In this way it is used for cross-marketing in retail stores. Other application areas, such as stock tickers, network traffic measurements, web server logs, data feeds from sensor networks, and telecom call records can have similar solutions.

Mining tools are increasingly more accessible to biologists and these can often be applied to answer scientific questions in combination with other bioinformatics tools. It can relate to many other categories in health and biological related fields. The system could first identify text regions that are rich in scientific content, retrieve documents that have many such regions and focus on fact extraction from these regions. Section2 explains literature survey. The algorithm used is detailed in section3.In section4 experimental setup and the dataset used are described. Experimental evaluation, conclusion and future works are cited in section5, section6 and section7 respectively.

1.1 A Working Example

Table1: Profit Table

Item	A	B	E	F	G	W
Unit Profit	1	1	2	3	1	1

Table2: An Example Transactional Database

TID	ITEMSETS	TRANSACTION UTILITY(TU)
T1	A,B,E,W	5
T2	A,B,E	8
T3	A,B,F	8
T4	E,G	5
T5	A,B,F	11

Demonstrating frequent itemset mining:-

During the first scan the support count(the number of occurrences) of each candidates are A-4 ,B-4,E-3,W-1,F-2,G-1.Suppose the minimum support count is 2,the new list of generated candidates are {A,B,E}.Again various combinations of two are taken from obtained list. Then the candidates are AB-4,AE-2,BE-2,AF-2,BF-2,EF-0.Then triples taken to form ABE-2,ABF-2,BEF-0,AEF-0.ABE and ABF satisfies the minimum support. Then quadruples taken so that ABEF, with support zero.ABEW's support is one. No such combination occurs. So the frequent itemset generated is ABE and ABF with minimum support 2.Candidate itemset generated list will be very high if the number of transactions is more. Only the support count is considered. So no profit generated items are discovered. The results of applying hui mining in the same example database are shown below. Here we are using a value called transaction utility(tu) for calculating the profit generated by individual items or itemsets.The item/itemsets generated are A-4,B-4,E-12,W-1,F-15,G-1,AB-8,AE-10,AW-2,BE-10,BW-2,EW-3, AF-17, BF-17,EG-5,ABE-12,ABW-3,AEW-4,BEW-4,ABF-19, ABEW-5 etc(only considerd needed ones). Here the minimum threshold is 10.Therefore high utility itemsets are E,F,AE,BE,AF,BF,ABE,ABF.

The absolute utility of A=4×1.

AE=(1×1 + 1×2) + (1×1 + 3×2)=10.

ABE=(1×1 + 1×1 + 1×2) + (1×1 + 1×1 + 3×2) =12.

x Ɛ L is closed iff there exists no itemset y Ɛ L such that x subset of y and SC(x) = SC(y). Closed high utility itemset means satisfies both minimum threshold and support. List of closed itemsets considered only support. Suppose minimum support is 1.They are E;3,EG:1,AB:4,ABE:2, ABF:2, ABEW:1.The complete set of closed itemsets in the database are E,ABE,ABF.HUI is said to be maximal if it is not a subset of any other hui.The set of maximal hui's in the above example is ABE,ABF[1].While applying top-k to the set of closed itemsets the itemsets retrieved will be same but difference in memory and execution time. Transaction weighted utilization (twu) of an itemset is the sum of the transaction utilities of all the transactions containing that itemset.The absolute utility of an item in a transaction is defined as the product of internal and external utilities. Internal utility means purchase quantity. Profit is the external utility.

2. LITERATURE SURVEY

Vincent S.Tseng, Cheng-Wei Wu., Philippe Fournier Viger, Philip S. Yu et at in [1] explains the limitations of frequent pattern mining, high utility mining. Also introduces a new concept called closed high utility mining. Recovery for missed items is also done. References [3,7] includes the ideas of high utility mining. UP-Growth an Up-Growth+ algorithms are explained. Construction of UP-tree also explained and various strategies for reducing the number of candidates. UPGrowth+, not only reduce the number of candidates effectively but also outperform other algorithms substantially in terms of runtime, especially when databases contain lots of long transactions. The various strategies discarding utilities are DGU, DLU, DGN, DLN. Reference[4] is the basis for most of the data mining algorithms. About association rules in finding itemsets in large databases are described in it. The basic algorithm Apriori for utility mining is cited here. Generation of synthetic data is also explained.

The different top-k concepts are included in references [2, 5, 8, 9, 10].In top-k no need of threshold. Top-k values are retrieved. Efficient in terms of memory and space. The different algorithms are also explained. Top-k frequent pattern mining is explained in [5].Top-k with effective threshold raising strategies are detailed in [8].In [10] top-k sequential patterns are mined. Sequential pattern mining means finding frequent subsequence's in a sequence database. The concept of high utility mining came from frequent itemset mining. The commonly used algorithm FP-Growth and basic idea is given in references[5,6].FP-tree is also explained.

References [11,12] describes the importance of high utility itemset mining. The application clickstream analysis is explained in it.The concept of top-k frequent pattern mining is referenced from [13,14].Top-k frequent mining can be with minimum threshold or not. In [15] algorithm CLS-Miner explained which gives the basis for the algorithm CHUD. M. J. Zaki,C. J. Hsiao et at in [16] describes the Eclat algorithm. References [17,18] explains DCI-Closed algorithm and the concept of closed high utility itemsets.Both Eclat and DCI-Closed algorithm forms the basis for CHUD algorithm. Éclat also provides foundation for IT-tree data structure. Mohammed J. Zaki,Ching-Jui Hsiao et at in [19] explained the IT-tree in detail.IT-tree is the search tree. Its closure, properties, subsumption checking all are explained in this reference. Further references for the IT-tree are [20,21,22].

3. METHODOLOGY USED

3.1 About CHUD and IT-tree

It consists of two phases. Phase I - potential closed high utility itemsets found(PCHUI).PCHUIs are defined as a set of itemsets having estimated utility no less than absolute minimum utility. Phase II-By scanning the database once, CHUIs are identified from the set of PCHUIs found in Phase I and their utility unit array are computed. Utility unit array for lossless representation. CHUD [Closed+ High Utility Itemset Discovery] is an extension of Eclat[16] and DCI-Closed[17] algorithm. It considers vertical database and mines CHUIs in a depth-first manner. An itemset is closed+ high utility itemset(CHUI) if it is a closed high utility itemset and that itemset should be included in utility unit array.

IT-Tree[19] means Itemset-Tidset pair tree. It is for finding CHUIs.Each node has an itemset,Tid,two ordered sets of items named PREVIOUS-SET and POST-SET.Figure1 shows an IT-tree for the transaction database shown in Table2. Also here each node is attached with an estimated utility value. But it is not shown in the figure1,but explained in previous sections. A data structure called transaction utility table (TU-table) is adopted for storing the transaction utilities of transactions. It is a pair with Tid and the second value is the transaction utility. So if Tid given, we can efficiently retrieve the TU value[18].The working of IT-tree similar to Eclat (Equivalence Class Transformation) algorithm. Given an itemset X ,t(X) is the set of all tids that contains X .From the above transactional database; Example:-(ABE)=12.Difference between Eclat and Apriori is that how they traverses prefix tree and how they determine the support of an itemset.Apriori is traversed in breadth-first order and support calculated by scanning the whole database. Éclat is depth-first ordering and the support of a new itemset by computing the intersection between tidsets.Given a tidset Y ,i(Y) is the set of all common items to all the tids in Y. Example:-i(123)=AB. Each node is represented as X × t(X) where X is the itemset and t(X) is the tidset.Example:-AB×1235.All the children of node X share the same prefix and belong to an equivalence classs.Using IT-Tree structure, the support values of the candidate itemsets can be computed rapidly using tidset intersections.

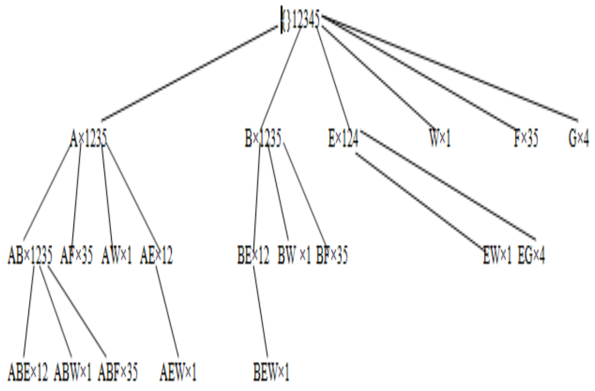

Figure1: An example IT-tree

Closure of an itemset X [c(X)] is the smallest closed set that contains X[20,21].To find the closure of an itemset X:-

1) Compute the image of X in the transaction space to get t(X).
2) Next map t(X) to its image in the itemset space using the mapping to get i(t(X)).
   Then that resulting itemset must be closed. That is, c(X) = i ᵒ t(X) = i(t(X)).

From Table2,c(ABE) = i(t(ABE)) = i(12) = ABE. Therefore ABE closed. Then consider AE; c(AE) = i(t(AE)) =i(12).As already seen i(12) can be A, B, AB, AE, ABE. So it is not closed. The support of an itemset X is equal to support of closure[21,22].In an IT-tree framework, for a given node or prefix class, one can perform intersections of the tidsets of all pairs of elements in a class, and check if minimum support is met; support counting is simultaneous with generation. Each resulting frequent itemset is a class unto itself, with its own elements, that will be recursively expanded. That is to say, for a given class of itemsets with prefix *P*, [*P*] = {l1, l2, · · · , ln}, one can perform the intersection of t(Pli) with all t(Plj) with j > i, to obtain a new class of frequent extensions,{ [Pli] = {lj | j > i and σ(Pli * lj) ≥ minimum support}. Also subsumption checking performed. Let Xi and Xj be two itemsets,then Xi subsumes another itemset Xj,if and only if Xj subset of Xi and support count(Xi) = support count(Xj).The need for subsumption checking is that after adding a closed set when we explore subsequent branches, we may generate another set, which can't be extended further and the new set is the subset or equal to the former and their support counts are equal. In this case the new set is non-closed set subsumed by the former and it should not be added to closed set. For example From Table2,{F} is subsumed by {ABF} because {F} ⊂ {ABF} and SC({F}) = SC({ABF}). An item is a promising item if transaction weighted utilization of the item is greater than or equal to absolute minimum utility. Otherwise,it is an unpromising item[1].

Utility unit array is to make the representation lossless.Each closed HUI is annotated with a special structure called utility unit array such that the resulting itemset is called a closed+ high utility itemset[1,24]. The idea of utility unit array makes the set of CHUIs lossless because HUIs and their utilities can be derived from this set without accessing the original database. Let X=[v1,v2,……..vk] and contains k utility values. The ith utility value v_i in v(X) is denoted as v(X,a_i). The utility value of X can be expressed as u(X) =sum of all(X, a_i).From Table2, the first utility value in V{ABF} is V{ABF,A}=absolute utility(A,T3) + absolute utility(A,T5) = 2. V{ABF,B}=absolute utility(B,T3) + absolute utility(B,T5) = 2. V{ABF,F}=absolute utility(F,T3) + absolute utility (F,T5) =15.Therefore the utility unit array of {ABF} is V{ABF}=[2,2,15].

During the first phase subsume check. Compute closure of itemsets.Then exploration of the IT-tree is done.

ALGORITHM SubsumeCheck

Input: Node X, Prev-set(X).
Output:- True: If X is non-closed and subsumed by other itemsets.
False: If X is not subsumed by other itemsets.

```
Step1: For each item i an element of PREV-SET(X) do
Step2: If g(X) Subset or equal to g(a) then true
else false.
```

ALGORITHM Closure-itemsets

Input: The node of X;N(X),POST-SET(X).
Output: The closure of X;X_c.

```
Step1: Initialize $X_c$=X.
Step2: For each item i Ɛ POST-SET(X) do
Step3: If g(X) subset or equal to g(i) then
Step4: POST-SET(X) = POST-SET$_X$/{i}.
Step5: $X_c$= $X_c$ U {a}.
Step6: Finally $X_c$ is returned.
```

ALGORITHM : Top-k CHUD(TCHUD)

Input: The transactional database, Minimum utility value. Output: The complete set of CHUIs.

```
Step1: Scan the database to convert into vertical database.
Step2: After scan promising items are collected into an ordered list, in the increasing order of support. Unpromising items are removed from the utility table.
Step3: For each item generates the node in the IT-tree such that candidates are generated in recursive manner, starting from single promising items and recursively joining items to them to from a larger list.
Step4: From the IT-tree, Top-k items are only stored for future accesses. Here k is the absolute minimum utility.
Step5: Top-k hui-search is performed[2](upto this phaseI).
Step6: Again Phase II to obtain all CHUIs.The arguments passed are the database and minimum utility.
```

Recovery is performed on closed top-k utility items to find missed items or itemsets if there are some one. Recovery is done using DAHU(Derive All High Utility Itemsets)

ALGORITHM : DAHU

Input: Maximum length of itemset,Result of Top-k CHUD(the complete set of CHUIs),Absolute minimum utility. HC={HC_1,HC_2…….HC_{ML}}.
Output: The complete set of CHUIs.

```
Step1: $H_{ML}$= $HC_{ML}$
Step2: $H_k$ is constructed from k=(ML-1) to k=1.
Step3: In each iteration $H_{k-1}$ is recovered by using $HC_k$.
Step4: For each itemset X,if absolute utility of X<absolute minimum utility; outputs hui X and all its (k-1) subsets.
Step5: For each item $a_i$ in X do,
Step6:Y=X- $a_i$.
Step7:Absolute utility(Y)= :Absolute utility(X)-Utility unit array(X, $a_i$).)
Step8: Absolute utility(Y) ≥ Absolute minimum utility do step 9 to 13.
Step9:Again Y an element of $HC_{k-1}$ and Support(X) > Support(Y) then
Step10: Support(Y) =Support(X).
Step11:Otherwise,if Y not an element of $HC_{k-1}$,then
Step12: $HC_{k-1}$= $HC_{k-1}$ U Y.
Step13: Support(Y) =Support(X).
```

4. EXPERIMENTAL SETUP AND DATASET USED

Experiments are performed with Intel Core i5 processor windows operating system doing with 2GB RAM. Algorithms are implemented in Java in Netbeans IDE. The backend is MySQL. Real life datasets like Mushroom, Foodmart and BMS WebView1 are used for evaluation.

Table3: Characteristics of Datasets with different parameters for Synthetic Datasets[1]

Dataset	N	T	D
Mushroom	119	23	8124
Foodmart	1559	4.4	4141
BMSWebView1	497	2.51	59601

N – Number of distinct items.
T – Average Transaction Length.
D – Total number of transactions.

Mushroom, Foodmart and BMSWebView1 is a real-life dataset, Mushroom contains 23 .BMSWebView1 is a click-stream data with a mix of short and long transactions from an e-commerce application.Foodmart contains real external and internal utilities and is obtained from Microsoft foodmart 2000 database. The characteristics and count distributions of the datasets varies depending on the applications. The three kinds of datasets encountered in real-life scenarios are (1)dense dataset,(2) sparse dataset, and(3) dataset containing long transactions.Mushroom,Foodmart,BMSWebView1 respectively represent the above three real cases.

5. EXPERIMENTAL EVALUATION

5.1 Mushroom Dataset

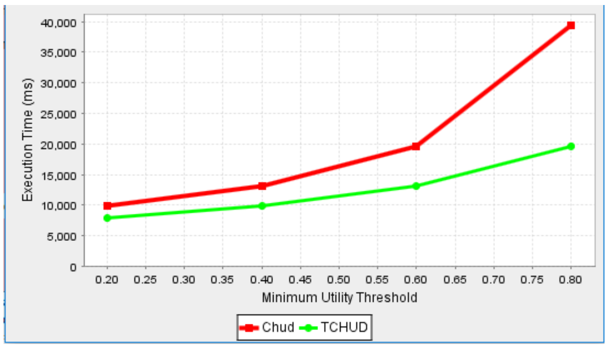

Figure2: Execution time graph for mushroom dataset

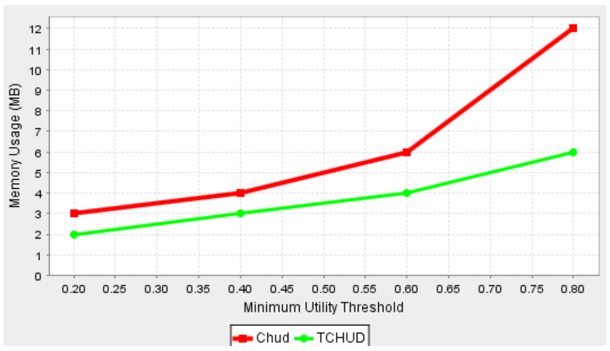

Figure3: Memory graph for mushroom dataset

From the graph it is clear that the execution time required for TCHUD is lower than that of CHUD.The time is approximately 39329ms.Similar is the case of memory and it is 12.78MB.Candidate count is 1404584.Closed high utility itemset with top-k is 230585.The recovered count is 238709.This is shown in Figure2 and Figure3 above.

5.2 Foodmart Dataset

Figure4 and Figure 5 shows the memory and execution time graph.TCHUD outperforms CHUD both in space and time saving. The execution time is 3983ms.6.84MB is the memory used. Candidate count is 4735964.The CHUIs with top-k is 6655.Also with recovery is 6680.

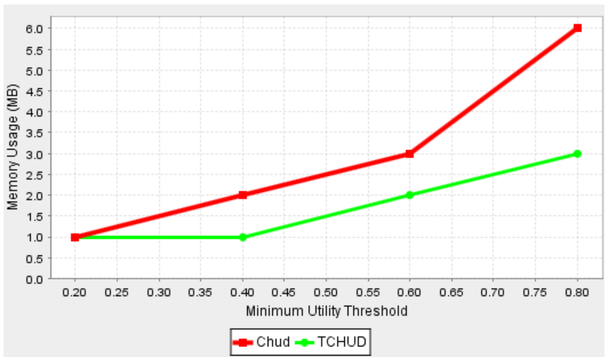

Figure4: Memory Usage Graph for foodmart dataset

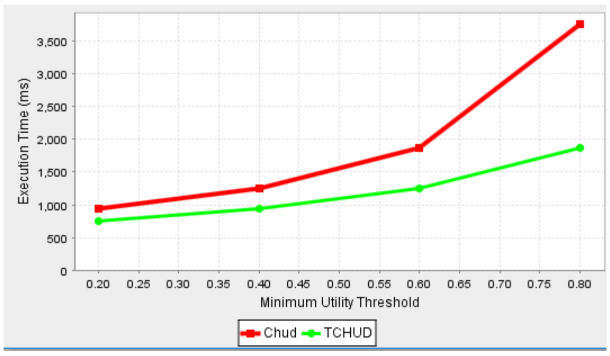

Figure5: Execution time graph for foodmart dataset

5.3 BMSWebView1

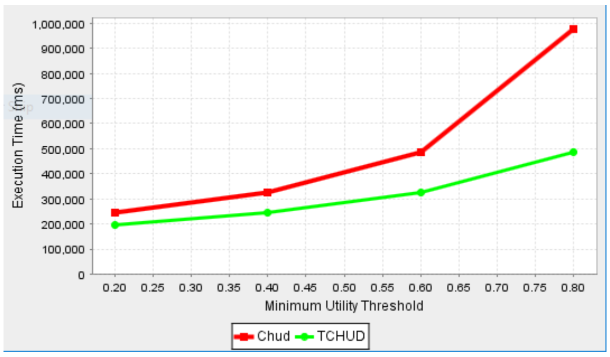

Figure6: Execution time graph for BMSWebView1

Here also the execution time and memory less for TCHUD compared to CHUD.BMS is a very large dataset. Therefore the candidates generated are 24306564.Execution time and memory used are 974774ms and 15.69MB respectively. Closed high utility itemset count with top-k is 1423592.With recovery is 1427215.Figure6 and Figure7 shows this.

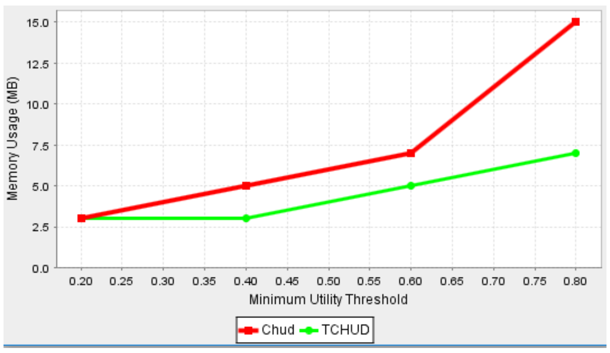

Figure7: Memory usage graph for BMSWebView1

6. CONCLUSIONS

Frequent itemset mining is good if weightage is given to profit earning items or itemsets. It considers only the count of occurences. High utility mining considers only profit. While closed+ high utility itemset mining considers both profit and the count of their occurrences. Using CHUD it can be concise and lossless. Top-k with CHUD will further reduce the memory consumption and will increase the speed. Also the number of candidates for PhaseII will be less. DAHU used to efficiently recover all high utility itemsets from closed+ high utility itemsets. Results show that TCHUD outperforms CHUD.

7. FUTURE-WORKS

Top-K can be applied to various high utility mining techniques.There are several other compact representations like free-sets, non-derivable sets,disjunctive,maximal itemsets etc. So that redundant patterns can be reduced. Also top-k HUI mining, it has not yet been incorporated with other utility mining tasks to discover different types of top-k high utility patterns such as top-k high utility episodes, top-k closed+ high utility itemsets, top-k high utility web access patterns and top-k mobile high utility sequential patterns. These leave wide rooms for exploration as future work[2].We can try different strategies for reducing the candidates generated during PhaseI.

This idea can be used with other types of databases other than transaction database.

8. ACKNOWLEDGMENTS

Thanks to the almighty and to those who guided me to complete my project. Also to the staff in the Computer Science department who supported us for their discussions and valuable suggestions.

9. REFERENCES

[1] Vincent S.Tseng, Cheng-Wei Wu, and Philippe Fournier-Viger,PhilipS.Yu. 2016 Efficient Algorithms for Mining Top-k High Utility Itemset. IEEE Transactions on Knowledge and Data Eng.,1-13.

[2] Vincent S.Tseng, Cheng-Wei Wu, and Philippe Fournier-Viger,PhilipS.Yu. 2016 Efficient Algorithms for Mining Top-k High Utility Itemsets. IEEE Transactions on Knowledge and Data Eng.,1-13.

[3] Vincent S.Tseng, Bai-En Shie,Cheng-Wei Wu and Philip S.Yu. 2013 Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases. IEEE Transactions on Knowledge and Data Engineering,1772-1786.

[4] R. Agrawal and R. Srikant. 1994 Fast Nearest Neighbor Search with keywords. In Proc. of Int'l Conf. on Very Large Data Bases, 487-499.

[5] G. Pyun and U. Yun. 2014 Mining Top-K Frequent Patterns with Combination Reducing Techniques. Applied Intelligence,76-98.

[6] J. Han, J. Pei and Y. Yin. 2000 Mining Frequent Patterns without Candidate Generation. ,In Proc. of ACM SIGMOD Int'l Conf. on Management of Data, 1-12.

[7] Vincent S.Teng,Cheng-Wei Wu,Bai-En Shie and Philip S.Yu . 2010 UP Growth : An efficient algorithm for High Utility Itemset Mining. In Proc. of the ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, 253-262.

[8] H. Ryang and U. Yun. 2015 Top-K High Utility Pattern Mining with Effective Threshold Raising Strategies. Knowledge-Based Systems ,109-126.

[9] C. Wu, B. Shie, V. S. Tseng and P. S. Yu. 2012 Mining Top-K High Utility Itemsets. In Proc. Of the ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining,1776 - 7886.

[10] J. Yin, Z. Zheng, L. Cao, Y. Songand W.We. 2013 Mining Top-K High Utility Sequential Patterns. In Proc. of IEEE Int'l Conf. on Data Mining,1259-1264.

[11] Ting, I. H.; Kimble C; Kudenko. D. 2005. UBB Mining: Finding Unexpected Browsing Behaviour in Clickstream Data to Improve a Web Site's Design. In Proc. Of Intl.Conf. on Web Intelligence,179-185.

[12] Patrali Chatterjee, Donna L. Hoffman and Thomas P. Novak. 2003 Modeling the Clickstream: Implications for Web-Based Advertising Efforts. in Marketing Science, 520-541.

[13] Dam TL,Li K.,Fournier-Viger P.,Duong QH. 2016 An efficient algorithm for mining top-rank-k frequent patterns. Application Intelligence Springer 45(1),96-111.

[14] Wang JY,Han JW Lu Y.,Tzvetkov P. 2005 TFP:An efficient algorithm for mining top-k frequent closed itemsets. IEEE Transactions on Knowledge and Data Eng.17(5),652-664.

[15] Dam TL,Li K.,Fournier Viger P.,Duong OH. 2016 CLS-Miner:Efficient and Effective closed high utility itemset mining. Frontiers of Computer Science.Springer.

[16] M. J. Zaki and C. J. Hsiao.2005 "Efficient algorithms for mining closed itemsets and their lattice structure. IEEE Trans. On Knowl. And Data Eng. 462–478.

[17] C. Lucchese, S. Orlando, and R. Perego. 2006 Fast and memory efficient mining of frequent closed itemsets. IEEE Trans. Knowl. Data Eng.,21–36.

[18] C.-W Wu, P. Fournier-Viger, P. S. Yu, and V. S. Tseng. 2011 Efficient mining of a concise and lossless representation of high utility itemsets. in Proc. IEEE Int. Conf. Data Mining,824–833.

[19] Mohammed J. Zaki, Ching-Jui Hsiao. CHARM:An efficient algorithm for closed mining. 2002 in Proceedings of SIAM intl. conf. on data mining,Society for industrial and applied Mathematics,457-473.

[20] B. Ganter and R. Wille. 1999 Formal Concept Analysis: Mathematical Foundations. Springer-Verlag.

[21] M. J. Zaki. 2000 Generating non-redundant association rules. In 6th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining.

[22] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. 1999 Discovering frequent closed itemsets for association rules. In 7th Intl. Conf. on Database Theory.

23] C. F. Ahmed, S. K. Tanbeer, B.-S. Jeong, and Y.-K. Lee. 2009 Efficient tree structures for high utility pattern mining in incremental databases. IEEE Trans. Knowl. Data Eng., 1708– 1721.

[24] Cheng-Wei Wu, Philippe Fournier-Viger, Jia-Yuan Gu, Vincent S. Tseng. 2015 Mining Closed+ High Utility Itemsets without Candidate Generation. In IEEE Conference on Technologies and Applications of Artificial Intelligence (TAAI).

[25] The full wiki http://www.thefullwiki.org