ISSN: 0976-3104
SPECIAL ISSUE: Emerging Technologies in Networking and Security (ETNS)
Velayudhan et al.

THE IIOAB JOURNAL

ARTICLE OPEN ACCESS

DEVELOPMENT OF AN EFFICIENT TECHNIQUE FOR MINING TOP-K CLOSED HIGH UTILITY ITEMSETS

Baby Velayudhan, Sakthivel and Subasree

Dept of Computer Science and Engineering, Nehru College of Engineering and Research Centre, Pampady, Thiruvilwamala, Thrissur, Kerala, INDIA

ABSTRACT

Discovery of High Utility Itemsets (HUI) or pattern from database is very useful in processing business. Recently proposed two techniques for data mining Closed High Utility Itemsets (CHUIs) and Direct Discovery of High Utility Pattern have been identified and have implemented and analyzed. From the result it is found that CHUI mining is the compact mining technique which reduces the number of itemsets by several orders of magnitude. Still the end user has to select the minimum utility for selecting CHUI which is a tedious process. CHUI refers to discover all itemsets having a utility meeting a user-specified minimum utility threshold min util. User has to set appropriate minimum utility threshold by trial and error, which is a tedious process. If min util is set too low, too many CHUIs will be generated, which may cause the mining process to be very inefficient. On the other hand, if min util is set too high, it is likely that Setting min util high, it is likely that no CHUIs will be found. This project addresses the above issues by proposing a new framework for top-k closed high utility itemset mining (TopK-CHUI), where k is the desired number of CHUIs to be mined. Results shows that the user can easily retrieve Closed High Utility Itemset by specifying k, which is the TopK number of CHUIs. TopK-CHUI is efficient and user friendly. Results on real datasets show that the technique TopK-CHUI is very efficient for the end user. So TopK-CHUI is an user friendly data mining technique.
.

Published on: 08th– August-2016

KEY WORDS

Closed high utility itemsets, TopK-CHUI mining, High utility itemset, Utility mining, Data mining.

***Corresponding author: Email:** babyvellayudhan@gmail.com ; Tel.: +91 77 36 830775

INTRODUCTION

Discover the sets of items (itemsets) with high utilities such as high profits is the important requirement of a business. For this utility mining is used. In each transaction, each item has a weight (e.g. unit profit) and appear more than once in each transaction (e.g. purchase quantity), these are utility mining. The important of an itemset is represented by the utility. Utility can be measured in terms of weight, profit, cost, quantity or other information depending on the user preference. Comprehension will be very difficult for the users if the algorithm gives a large number of high utility patterns.

Candidate pattern's Transaction Weighted Utilization (TWU) [2, 5, 7] is the transaction's utility sum. If an itemset's utility is no less than a user-specified minimum utility threshold then it is called a High Utility Itemset (HUI) [2, 4, 5, 10] otherwise, it is called a low utility itemset. Utility mining is an important task and has a wide range of applications such as biomedical applications, website click stream analysis, cross marketing in retail stores and mobile commerce environment.

If a HUI is not a subset of any other HUI then it is said to be maximal [1, 6]. The reason is that without scanning the database the utilities of the subsets of a maximal HUI cannot be known. If a high utility itemset has no proper superset having the same utility then it is said to be closed [1, 6]. For any non-closed high utility itemset Y, Y does not appear in a transaction without its closure Z. Moreover, the utility (e.g.profit/user preference) of Z is guaranteed to be higher than the utility of Y. For these reasons,users are more interested in finding Z than Y. Moreover, closed itemsets having high utilities are useful in many applications. For example, in market basket analysis, Z is the closure of Y means that no customer purchases Y without its closure Z. Thus, when a customer purchases Y, the retailer can recommend Z - Y to the customer, to maximize profit.

CHUI integrates high utility itemset mining into the concept of closed itemset.Closed High Utility itemset Discovery (CHUD) [1] algorithm mines CHUIs in a depth-first search by using vertical database. CHUD takes as parameter the

THE IIOAB JOURNAL
ISSN: 0976-3104

abs min utility threshold and a database D. In practice it is difficult for users to choose an appropriate minimum utility threshold. The output size can be very small or very large depending on the threshold.To discover the itemsets with the highest utilities and precisely control the output size without setting the thresholds, a promising solution is to redefine the task of mining HUIs as mining Top-k High Utility Itemsets (Top-k HUIs) [3].

Instead of specifying the minimum utility threshold, let the users specify k, i.e., the number of desired itemsets. Alphabet k represents the number of itemsets that the users want to find whereas choosing the threshold depends primarily on database characteristics, which are often unknown to users so setting k is more intuitive than setting the threshold. For many applications it is very desirable that using a parameter k instead of the min_util threshold. For example, Top-k HUI mining serves as a promising solution for users who desire to know ”What are the top-k sets of products (i.e., itemsets) that contribute the highest profits to the company?” and ”How to efficiently find these itemsets without setting the min_util threshold?”, to analyze customer purchase behavior.

In top-k HUI mining The min_util threshold is not given in advance .The search space can be efficiently pruned by the algorithms by using a given min util threshold in traditional HUI mining . However, no min util threshold is provided in advance in the scenario of top-k HUI mining . Therefore, the designed algorithm has to gradually raise the threshold to prune the search space by setting the minimum utility threshold initially to zero [3]. Such a threshold is an internal parameter of the designed algorithm and is called the border minimum utility threshold min_utilKBorder in TopK-CHUI technique.

BACKGROUND AND PROBLEM DEFINITION

Problem Definition

CHUI is a compact and lossless representation of HUIs. From the analyses conducted it is found that CHUI is the best techniques which retrieves very less number of HUIs that is why CHUI is compact and lossless representation of HUIs. For mining CHUIs user has to produce the min util and the dataset, but prediction of min util is a tedious job for the user. User has to follow trial and error method for selecting the value of min util. If the user is giving min util near to zero then mining will produce too many CHUIs and comprehension of these are very difficult. If the user is giving min util as high value then there may not be any itemset to display. So for using the mining technique efficiently user must know the highest utility value available in the dataset which is not easy. If the user knows the highest value then he can get the CHUIs by giving percentage of highest value as min util. At the same time if he wants to see the highest k CHUIs as the output then it is easy for the user to say the value of k.

The problem statement is: Given a transactional database D and the desired number of CHUIs k, the problem of top-k closed high utility itemsets mining is to discover all the itemsets having a utility no less than delta in D.

Related Works

This subsection introduces related works about top-k closed high utility itemset mining, including closed high utility item-set mining and top-k high utility itemset mining.

1) Closed high utility item discovery: Closed High utility itemset incorporates the concept of closed itemset with high utility itemset mining. Closure on the utility of itemsets can define as a high utility itemset is said to be closed if it has no proper superset having the same utility. In real dataset it is unlikely to achieve a high reduction of the number of extracted itemsets since not many itemsets have exactly the same utility as their supersets. The join order between the closed constraint and the utility constraint is defined as a) Mine all the high utility itemsets first and then apply the closed constraint. b) Mine all the closed itemsets first and then apply the utility constraint. The two constraints can be applied in any order during the mining process.

The closed high utility itemset discovery technique con-siders vertical database and mines CHUIs in a depth-first search. CHUD takes as parameter a database D and the abs min utility threshold. CHUD convert D into a vertical database during the first scan of D . At the same time, CHUD calculates TWU [2, 5, 7] of items and computes the trans-action utility for each transaction TR. When a transaction is retrieved, its transaction utility and Tid(transaction identifier) are loaded into a GTU(global TU-Table). If an item’s esti-mated utility (e.g. its TWU) is no less than abs min utilityis then it is called a promising item. After the database scan, promising items are collected into an ordered list O, sorted according to a fixed order such as increasing order of support. Since supersets of unpromising items are not CHUIs, only promising items are kept in O.

CHUD uses Itemset - TidsetpairTree(IT-Tree) [8] to find CHUIs. Each node consists of an itemset X, Tidset g(X), two ordered sets of items PREV-SET(X) and POST-SET(X) and estimated utility. The TU-Table stores the transaction utility with transaction id.In a recursive manner CHUD generates candidates , starts with candidates containing a single promising item and recursively joins items to them to form larger candidates. This is done by using the total order < , the complete set of itemsets is divided into n non-overlapping subspaces, where the k^{th} subspace is the set of itemsets containing the item a_k but no item $a_i < a_k$. For each item a_k belongs to O, CHUD creates a node N(fa_k g) and puts items a_1 to a_{k-1} into PREV-SET(a_k) and items a_{k+1} to a_n into POST-SET(a_k).

THE IIOAB JOURNAL
ISSN: 0976-3104

2) Top-k high utility itemset mining: High utility itemset mining algorithms can be generally categorized into two types: two-phase and one-phase algorithms. The main characteristic of two-phase algorithms is that they consist of two phases. In the first phase, they generate a set of candidates that are potential high utility itemsets. In the second phase, they calculate the exact utility of each candidate found in the first phase to identify high utility itemsets.

Let C be the set of candidates produced in Phase I. Can-didates in C are sorted in descending order of their estimated utilities. Thus, candidates with higher estimated utility values will be considered before those having lower estimated utility values. During the phase II, if the utility of a newly considered CHUI X is larger than min $util_{KBorder}$, X and EU(X) are inserted into a min-heap structure named TopK-CHUI-List. Then, min $util_{KBorder}$ is raised to the utility of the k-th HUI in TopK-CHUI-List, and HUIs having a utility lower than min $util_{KBorder}$ are removed from TopK-CHUI-List. If the estimated utility of the current candidate Y is less than the raised min $util_{KBorder}$, Y and the remaining candidates do not need to be considered any more because the upper bounds on their utilities are less than min $util_{KBorder}$. When the algorithm completes, TopK-CHUI-List captures all the Top-k CHUIs in the database.

TOPK-CHUI TECHNIQUE

In this section, an efficient technique named TopK-CHUI (mining Top-K closed utility itemsets)is proposed for discov-ering Top-k CHUIs without specifying min util.The strategy used in TopK-CHUI technique is that raising the threshold by sorting exact utility of candidates.

ALGORITHM: TopK-CHUI
Input: (1) A database D; (2) The number of desired CHUIs k; Output: (1) Top K CHUIs;
01. Set min_util$_{KBorder}$=0; TopKCHUI-List=; 02. Calculate the Transaction Weighted Utility 03. Create utility list of items 04. TopK-CHUI-Search(min_utilKBorder,closedset,utilitylist,preset,postset); 05. Output TopKCHUI-List

Fig: 1. TopK-CHUI

Figure-1 shows the pseudo code of TopK-CHUI.Each time a candidate itemset X is found by the CHUI search procedure, the TopK-CHUI algorithm checks whether its utility value is no less than min $util_{KBorder}$.If utility is less than min $util_{KBorder}$, X is not a TopK-CHUI. Otherwise, X is considered as TopK-CHUI.Each time a TopK-CHUI X is found and its utility is higher than min $util_{KBorder}$, X is added into TopKCHUI-List.If there are less than k itemsets in TopKCHUI-List, min $util_{KBorder}$ will not change. Once k itemsets are found and the k-th utility value in TopKCHUI-List is higher than min $util_{KBorder}$, min $util_{KBorder}$ is raised to k-th minimum utility value in TopKCHUI-List.The strategy used in TopK-CHUI Technique is raising the threshold by sorting the exact utilities of the candidates. If the utility of a newly considered CHUI X is larger than min $util_{KBorder}$, X is added into TopKCHUI-List.

Experiments were performed on a computer with a 1.7GHz Intel i3 processor and 4GB of memory, running Windows 8.1. Algorithms are implemented in Java. Real datasets were used in the experiments.Datasets were acquired from archive.ics.uci.edu/ml/datasets.html. It includes real datasets with synthetic utility values. Internal utility values generated using a uniform distribution in [9]. External utility values generated using gaussian (normal) distribution.

Table: 1. Dataset Characteristics

Dataset	#Transactions	#Distinct items	Avg. trans. Length
Accident	340183	468	33.8
BMS	59601	497	4.8
Connect	67557	129	43
Foodmart	4141	1559	4.4
Mushroom	8124	119	23

Table- 1 shows characteristics of the datasets. In real life scenarios there are three kinds of datasets that are commonly encountered: (1) dataset containing long transactions, (2) dense dataset, (3) sparse dataset. In the experiments have done by using real-life datasets BMS, Mushroom, Foodmart to respectively represent the above three real cases.

THE IIOAB JOURNAL
ISSN: 0976-3104

ALGORITHM: TopK-CHUI-Search

Input: (1) d: border minimum utility threshold min $util_{KBorder}$;
(2) Closed set;
(3) Utility list of closed set;
(4) List of Preset;
(5) List of Postset;
Result: (1) TopK-CHUI-List

1. For all i in postset
2. Calculate the tidset of closedset U i 03. If the first time it is the tidset of it
4. Otherwise intersect the tidset of closed set and the tidset of i
5. If the newly generated closed set has high utility supersets
6. Create the itemset for newly generated
7. If newly generated is not a duplicate
8. Calculate tidset for new closed set
9. for each item J in the postset
10. If J is smaller than newly generated according to the total order on items then skip it
11. Otherwise add item to the new closed set
12. Construct tid for new closed set –
13. Add to the new postset
14. If sum of the item utilities in new closed set= min $util_{KBorder}$
15. If TopK-CHUI-List contains less than k
16. Add to TopK-CHUI-List
17. Otherwise remove the smallest from TopK-CHUI-List and add new itemset to it
18. Sort the new list and min $util_{KBorder}$= smallest utility value in the TopK-CHUI-List.
19. new preset=preset–
20. TopK-CHUI-Search(min $util_{KBorder}$, new closed set, new closed set tids, new preset, new postset
21. preset=preset U i

Fig:2. TopK-CHUI-Search

RESULT AND DISCUSSION

Figure- 3(a) shows the runtime of the algorithm on the datasets with varied k respectively. **Figure- 3(b)** shows the minimum utility selected for the algorithm on the datasets with varied k respectively. **Figure- 3(c)** shows the memory used by the algorithm on the datasets with varied k respectively. Execution time varies according to the value of k which is given by the user as input. If user wants only Top10 itemsets then the execution time will be less. Execution time is high when the user uses large datasets, for example when k=10 the execution time is high for Accidents and BMS datasets.

In dense datasets like Mushroom memory usage is high but the execution time is less. For large dataset like Accidents, BMS and connect memory usage is less but the execution time is more. Minimum utility percentage is selected according to the utility in the dataset available. For example BMS have highest utility value. Memory usage is high for Mushroom dataset. **Table- 2**, **Table- 3** and **Table- 4** shows the runtime, minimum utility and memory usage on the datasets with varied k respectively. TopK-CHUI technique cannot compare with CHUI because CHUI takes input as the minimum utility but TopK-CHUI as the number of CHUI itemsets needed. So analysis have been done by running the technique with different real datasets for different values for k. It is also found from the analysis that if we are running the CHUI-Miner[1] for the minimum utility selected for k=10 then the execution time is less for TopK-CHUI technique

Table: 2. Execution Time (Seconds)

Database	K=10	K=100	k=1000	k=10000
Accidents	466.17	623.758	694.858	1001.56
BMS	365.54	1008.368	875.521	1290.21
Connect	55.794	60.511	56.882	168.916
Foodmart	3.12	3.727	4.094	6.335
Mushroom	17.97	29.265	35.108	74.034

THE IIOAB JOURNAL
ISSN: 0976-3104

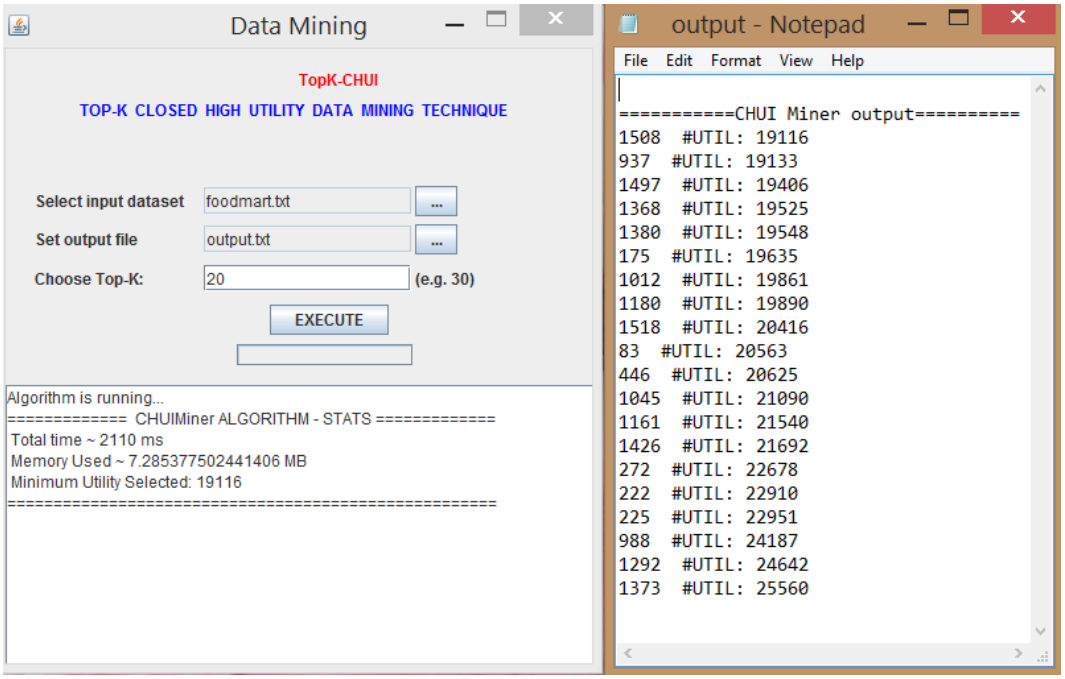

Fig: 3. Sample Output

Table: 3. Minimum Utility (%)

Database	K=10	K=100	k=1000	
Accidents	93.37	82.54	68.34	52.15
BMS	20.53	9.55	5.47	3.47
Connect	98.3	94.94	88.33	71.26
Foodmart	80.69	57.9	24.53	0
Mushroom	73.76	61.08	34.68	12.07

Table: 2. Memory Used (MB)

Database	K=10	K=100	k=1000	k=10000
Accidents	7.29	7.29	11.52	11.25
BMS	16.34	14.98	13.29	17.78
Connect	7.1	7.56	17.16	22.52
Foodmart	9.68	7.63	11.66	8.45
Mushroom	39.13	35.42	33.34	22.91

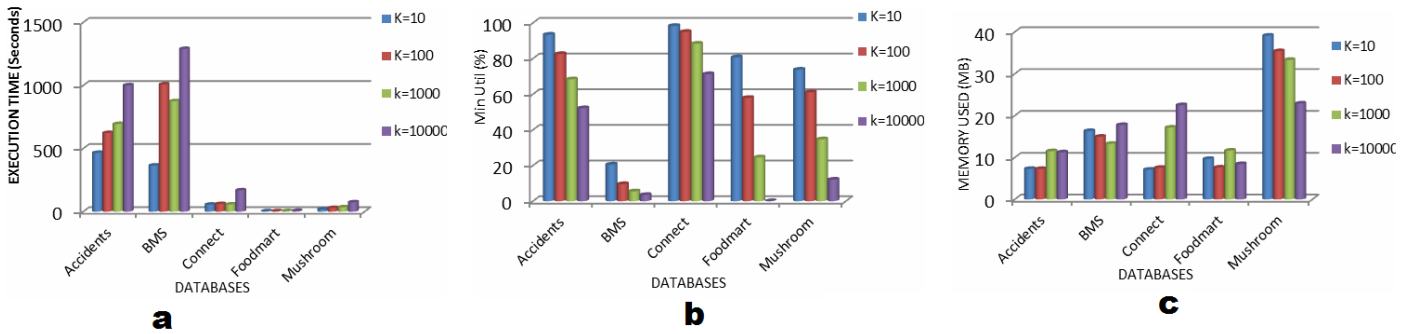

Fig: 4. a) Execution time, b) Minimum Utility and c) Memory used

THE IIOAB JOURNAL
ISSN: 0976-3104

CONCLUSION

The problem of top-k closed high utility itemsets mining, where k is the desired number of closed high utility itemsets to be mined is discussed. An efficient technique TopK-CHUI is proposed for mining such itemsets without setting minimum utility threshold. The strategy used with this is raising the threshold by sorting exact utilities of candidates. It is user friendly because user need not to do trial error method for determining the value of minimum utility, end user can directly give the value of k for to k closed high utility itemsets. Evaluations on different types of real datasets show that the proposed algorithm has good scalability on large datasets. It can be incorporated with other mining techniques such as top-k high utility web access patterns and top-k mobile high utility sequential patterns.

CONFLICT OF INTERESTS

The authors declare that they have no conflicts of interest.

ACKNOWLEDGEMENT

None.

FINANCIAL DISCLOSURE

NIL

REFERENCES

[1] Vincent S Tseng, Cheng-Wei Wu, Philippe Fournier-Viger, Philip S Yu.[2016] Efficient Algorithms for Mining the Concise and Lossless Repre-sentation of High Utility Itemsets, *IEEE Trans. Knowl. Data Eng*., 27(3): 726 -739.

[2] Junqiang Liu, Ke Wang, and Benjamin C.M. Fung. [2016] Mining High Utility Patterns in One Phase without Generating Candidates, *IEEE Trans. Knowl. Data Eng*, 28(5):1245- 1257

[3] Vincent S Tseng, Cheng-Wei Wu, Philippe Fournier-Viger, and Philip S Yu.[2016] Efficient Algorithms for Mining Top-K High Utility Itemsets, *IEEE Trans. Knowl. Data Eng*, 28(1): 54- 67.

[4] GC Lan, TP Hong, VS Tseng.[2014.]An efficient projection based indexing approach for mining high utility itemsets, *KAIS*, 38(1): 85-107

[5] VS Tseng, CW Wu, BE Shie, PS Yu.[2010] UP- Growth: An efficient algorithm for high utility itemset mining," in Proc. ACMSIGKDD *Int Conf Knowl Discov Data Mining* , 253-262.

[6] CW Wu, P Fournier-Viger, PS Yu, VS Tseng.[2011]Efficient mining of a concise and lossless representation of high utility itemsets, *in Proc. IEEE Int Conf Data Mining*, 824-833.

[7] VS Tseng, BEShie, CW Wu, PS Yu.[2013] Efficient algorithms for mining high utility itemsets from transactional databases,*IEEE TKDE*, 25(8):1772-1786,

[8] CW Lin, TP Hong, WH Lu.[2011] An effective tree structure for mining high utility itemsets, *Expert Syst Appl*, 38(6): 7419-7424

[9] R.Jaya and Rajan. C,[2016]A Study On Data Mining Techniques, Methods, Tools And Applications In Various Industries, International Journal on Concurrent Applied Research in Engineering and Management,vol.1, No.4, pp. 14-24

[10] M Liu and J Qu.[2012] Mining high utility itemsets without candidate generation, in CIKM. ACM, , 5564.