International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

High Utility Itemsets Mining – A Brief Explanation with a Proposal

Anu Augustin[1], Dr. Vince Paul[2]

[1]Sahrdaya College of Engineering and Technology, Kodakara

[2]HOD of the Department, Sahrdaya College of Engineering and Technology, Kodakara

Abstract: High utility itemsets mining is relevant for business vendors. So that they can give more offers to high utility itemsets. To understand the above sentence we need to know what is high utility itemsets. High utility itemsets are those ones that yield high profit when sold together or alone that meets a user-specified minimum utility threshold from a transactional database. This high utility itemset mining is not a new topic, but it is an emerging area. The basis of high utility mining is frequent itemset mining. The various problems in frequent itemset mining are purchase quantity not taken into account, all items have same importance etc. So the number of items generated will be more. These limitations are overcomed by high utility itemset mining. For that in HUI mining a utility value (weight) is assigned to each item. Also a threshold applied to remove unwanted itemsets. Setting the threshold externally is a tedious work. Too low threshold will generate many HUI's and too high may cause no HUI's to found. In Top-K only top hui' s will be found. Here the minimum threshold is set internally. It is zero initially. Performance degrades when there are many hui's in the database. So the concept of closed itemset mining is introduced for memory and space efficiency. Also for proper utilization of resources.

Keywords: Frequent Pattern, High Utility Itemsets, Transaction Utility, Minimum Utility Threshold,Closed itemsets

1. Introduction

Data mining and knowledge discovery from databases has received much attention in recent years. The extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining is the process of revealing nontrivial, previously unknown and potentially useful information from large databases. Discovering useful patterns hidden in a database plays an essential role in several data mining tasks, such as frequent pattern mining, weighted frequent pattern mining, and high utility pattern mining etc. Among them, frequent pattern mining is a fundamental research topic that has been applied to different kinds of databases, such as transactional databases, streaming databases, and time series databases, and various application domains, such as biomedicine, stream analysis, market analysis and mobile computing[1]. In view of this, utility mining emerges as an important topic in data mining field. Mining high utility itemsets from databases refers to finding the itemsets with high profits. Here, the meaning of itemset utility is interestingness, importance or profitability of an item to users. Utility of items in a transaction database consists of two aspects:

1) The importance of distinct items, which is called external utility, and
2) The importance of items in transactions, which is called internal utility.

Frequent itemset mining (FIM) finds frequent items and may lose information on items that may have low selling frequencies. It makes use of Apriori Property. Patterns are those ones that are repeated. Mining frequent items, itemsets, subsequences, or other substructures is usually among the first steps to analyze a large-scale dataset, which has been an active research topic in data mining for years. The various databases in frequent itemset mining are transactional databases, streaming databases and time series databases. Explaining each one; transactional database means a dbms where write transactions on the database are able to be rolled back if they are not completed properly (e.g. due to power or connectivity loss). Most modern relational database management systems fall into the category of databases that support transactions. A Data stream management system (DSMS) is a computer program to manage continuous data streams. It is similar to a dbms, which is, however, designed for static data in conventional databases. Also a time series database (TSDB) is a software system that is optimized for handling time series data, arrays of numbers indexed by time (a datetime or a datetime range). In some fields these time series are called profiles, curves, or traces.

The older methods consider the utility of the items by its presence in the transaction set. The frequency of itemset is not sufficient to reflect the actual utility of an itemset. Recently, one of the most challenging data mining tasks is the mining of high utility itemsets efficiently. Identification of the itemsets with high utilities is called as utility mining. The utility can be measured in terms of cost, quantity, profit or other expressions of user preferences. For example, a computer system may be more profitable than a telephone in terms of profit. Utility mining model was proposed to define the utility of itemset. The utility is a measure of how useful or profitable an itemset X. The utility of an itemset X, i.e., u(X), which is the sum of the all utilities of itemset X in all the transactions containing X. An itemset X is called a high utility itemset if and only if u(X) greater than or equal to min-utility, where min-utility is a user defined minimum utility threshold. The main objective of high-utility itemset mining is to find all those itemsets having utility greater or equal to user- defined minimum utility threshold .

Table 1: An Example Transactional Database

TID	*Transaction*	*Transaction Utility(TU)*
T_1	(A,1)(C,1)(D,1)	7
T_2	(A,2)(C,6)(E,2)(G,5)	30
T_3	(A,1)(B,2)(C,1)(D,6)(E,1)(F,5)	29
T_4	(B,4)(C,3)(D,3)(E,1)	20
T_5	(B,2)(C,2)(E,1)(G,2)	13

Table 2: Profit Table

Item	*A*	*B*	*C*	*D*	*E*	*F*	*G*
Unit Profit	4	2	1	2	3	1	2

Transaction utility (TU) of an itemset in Table 1 is TU (T1) = Estimated utility ({A}, T_1) + Estimated utility({C}, T_1) + Estimated utility({D}, T_1) = 1×4 + 1×1 + 1×2 =7.

Transaction weighted utilization (TWU) of A = TU(T_1) + TU(T_2) + TU(T_3) = 7 + 30 +29 =66. TWU of F =TU(T_3) 29. TWU of G = TU(T_2) + TU(T_5) = 30 +13 =43.Suppose the minimum utility threshold is 50, by calculating the transaction weighted utilization of F and G, that have low threshold can be eliminated. Again setting threshold is very difficult so top-k can be used. In top-k , k high utility itemsets are retrieved. Here threshold is set internally, initially it is set to 0.Raised automatically as it progresses. Also the concept of closed itemsets can be incorporated into top-k mining algorithms. A closed itemsets means hui's which has no proper superset as same utility.

Explaining each application in detail - A clickstream is the recording of the parts of the screen a computer user clicks on while web browsing or using another software application. Clickstream analysis is useful for web activity analysis, software testing, market research and for analyzing employee productivity. It can be used to improve customer satisfaction with the website and with the company itself. Data mining is emerging as a promising topic in mobile computing environments. The wide availability and growing computing power of mobile devices has opened the way for data analysis and mining in mobile scenarios. Examples include smart phone-based systems for body-health monitoring, vehicle control, and wireless security systems. Market analysis studies the attractiveness and dynamics of a special market within a special industry. It is part of the industry analysis and thus in turn of the global environmental analysis. Through all of these analyses, the strengths, weakness, opportunities and threats of a company can be identified. 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 existing systems and literature survey.In Section3 gives a brief description problem definition and about proposed system and section4 the conclusion. Finally the references for the work in section5.

2. Existing Systems A Survey

2.1 Apriori Algorithm

It is also for mining frequent itemsets. Frequent itemset mining is an essential part of machine learning algorithms. We have to mine most frequent itemsets from a big list of transactions. Apriori[3][5] rule states that if an itemset is infrequent then all its super sets are also infrequent and may be pruned. Support count is the frequency of occurrence of an item in the transaction database. Ex:- Support Count (A) = 2.Support means fraction of transactions that contain an itemset.

Ex:-Support (AC) = (2/4)=(1/2).

An Example explained below:-

Table 3: An Example Transactional Database

Tid	*Transactions*
T1	A,C,D,F
T2	A,C,E,G
T3	B,C,D,E
T4	B,C,E,G

Suppose the minimum support is 50%, then there should be atleast 2 transaction itemsets. In first step we are calculating the count of each items and applying the threshold. The count of each items are A:2, B:2, C:4, D:2, E:3, F:1, G:2 respectively and F is discarded. Secondly the items are paired. AB:0,AC:2,AD:1, AE:1,AG:1,BC:2, BD:1,BE:1,BG:1,CD:2,CE:3,CG:2,DE:1,DG:0,EG:2.Among these values AC,BC,CD,CE,CG,EG are selected that meets the threshold. In each step filtered using threshold. Again these values are tripled. Then only CEG:2 satisfies minimum support. If there are more than one, the process continued to quadruples and so on until it becomes only one. Then association is formed by eliminating the subsets from the triplets. So CE, CG,EG will be removed and the final result will be {{CEG},{AC},{BC},{CD}}.In this method candidate generation is slower-pairs, triples etc. Huge candidates are generated and memory consumption is more.

2.2 Frequent Itemset Mining

The basic idea of high utility mining comes from frequent itemset mining. One of its popular applications is market basket analysis. It refers to the discovery of sets of items (item-sets) that are frequently purchased together by customers. FIM[1],[2],[3] may discover a large amount of frequent but low-value itemsets and lose the information on valuable itemsets having low selling frequencies. High Utility Itemset mining is an extension of the problem of frequent pattern mining. The most popular pattern mining algorithm is Apriori. In Apriori candidates generated will be more. In frequent pattern mining allows frequent itemset generation without itemset generation. There are many limitations for frequent itemset mining.

- Purchase quantity is not taken into account. Suppose we have bought 3, 7 or 9 packets of bread, all are considered as the same.
- Same importance for all items, For example if a customer buys a bottle of wine or a packet of bun, both considered as the same.

- Frequent Pattern Mining (FPM) finds many non-interesting patterns. Bread and milk is a frequent pattern. But from business perspective this pattern may be uninteresting because it doesn't generate much profit. Also FPM may miss rare patterns that generate a high profit such as wine and caviar.

There are various algorithms for frequent itemset mining like FP-Growth [3], Eclat, Relim etc.Compared to these FP-Growth has better performance. The variants of FP-Growth are DynFP-Growth, FP-Growth*, PPV, PrePost and FIN algorithm.It consists of two steps. First build FP-tree. Then extract frequent items directly from FP-tree. In an FP-Tree each node represents an item and its current count, and each branch represents a different association. Here only needs the database twice. Efficient in execution than apriori and saves space. In FP-tree insert sorted items by frequency into a pattern tree. Runtime increases and stores a compact version of the database. There is very much data inter-dependency that every node need a root. Pointers are maintained between the nodes having same item and linked using singly linked list.

2.3 High Utility Mining

All limitations of frequent pattern mining can be reduced using high utility mining. High utility itemsets generate profit when sold together. Here the number of occurrences is counted. Here no apriori property. Each item have different importance depending on the transaction weighted utilization[1][2][4].With respect to frequent pattern mining only patterns having high utility are discovered. In utility mining each item has a cost and there is purchase quantity. The utility can be measured in terms of profit, cost, quantity or other information depending on the user preference. An itemset is called a high utility itemset (HUI) if its utility is no less than a user specified minimum utility threshold; otherwise, it is called a low utility itemset. In HUI mining setting the threshold is a difficult. If threshold is very high there is a lot itemsets. Also the memory consumed will be more. Again if it is very low some itemsets may be missed causing the failure of expected result.

The various algorithms for high utility mining are IHUP, IIDS, Two-Phase, d^2-HUP, HUI-Miner, Up-Growth, UP-Growth$^+$. Among these UP-Growth performs well. So it is explained in detail.HUI-Miner, d^2-HUP are one phase algorithms. A compact tree structure called UP-Tree is used. UP-Growth is a two phase algorithm. In phase I, the framework of UP-Tree follows three steps:
1) Construction of UP -Tree.
2) Generate PHUIs (Potential HUI's) from UP -Tree.
3) Identify high utility itemsets using PHUI.

In phase II, high utility itemsets and their utilities are identified from HTWUI's by scanning original database once. How we can construct a Up-tree is explained as in the next paragraph.UP-tree form the basis for most mining algorithms.

Constructing UP-tree – The elements of UP-tree are each node contains name, count, node utility, parent, hlink. In addition to this an header table which consists of item name, an overestimated utility, a link. Link points to the last occurrence of the node which has the same item as the entry in the UP-tree. Consider table 1 and 2 and construct transaction weighted utilization table such that TWU(A) = 7+30+29 = 66.Since A appears in T1,T2,T3.Similarly TWU's of all item are calculated. The Table 4 shown below:-

Table 4: Items and their TWU's based on Table1 and Table2

Item	*A*	*B*	*C*	*D*	*E*	*F*	*G*
TWU	66	62	99	56	92	29	43

Suppose minimum threshold is 50, F and G is eliminated. Then the table of transactions is re-organized in the descending order of TWU. Then inserted into the UP-tree. Table 5 below shows re-organized transactions. Also the constructed Up-tree shown below in figure 1.In UP-tree first one is to find support count (SC) and the second is the node utility. The nodes which have the same item names are linked in a sequence by their node links.

Table 5: Reorganized Transactions and their RTU's

TID	*Reorganized Transactions*	*RTU*
T_1'	(C,1) (A,1)D,1)	7
T_2'	(C,6)(E,2)(A,2)	30
T_3'	(C,1)(E,1)(A,1)(B,2) (D,6)	29
T_4'	(C,3)(E,1)(B,4)(D,3)	20
T_5'	(C,2)(E,1)(B,2)	13

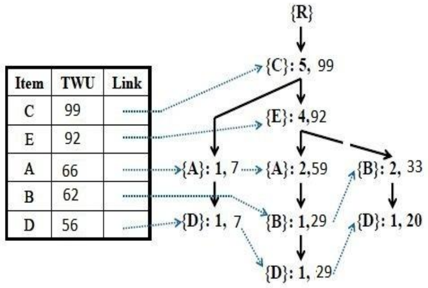

Figure 1: Constructed UP-Tree.

The UP-Growth[1],[2],[3],[4][5] algorithm consists of three parts:
1) Construction Of UP-Tree.
2) Generation of potential high utility itemsets (PHUI's) from Up-tree by UP-Growth.
3) Identification of hui's from the set of PHUI's. The term PHUI's to distinguish the discovered patterns from HTWUI's which are using in transaction-weighted utilization.

After constructing UP-tree, the algorithm start from the header of the table and it considers {D} first. By tracing the nodes to root there are three paths.(D→A→C: 1,7), (D→B→A→E→C: 1,29),(D→B→E→C: 1,20).The first number is count and the second one is the path utility. The paths are collected in the {D}'s conditional pattern base. After calculating local items and their path utilities in {D}-CPB the unpromising items are discarded. Unpromising items means that have threshold below the minimum one.

Again the re-organized paths are found out. Thus generating PHUI’s from {D}-tree by applying FP-Growth by scanning the original database. The generated PHUI’s are {{D}:56, {DE}:49,{DEB}:49,{DEC}:49,{DEBC}:49,{DB}:49, {DBC}: 49, {DC}:56}. In the same way consider each item in the header table and derive a set of PHUI’s.Finding all PHUI’s, high utility itemset and their utilities are identified from the set of PHUI by scanning original database once. The above strategy is DGU (Discarding global unpromising items) i.e., the unpromising items and their utilities are removed from the transaction utilities during the construction of a global UP-tree. To further reduce number of candidates DGN can be used.

DGN means discarding global node utilities. In this strategy the utilities of its descendants are discarded from the utility of the node during the construction of a global UP-tree. It is suitable for the database which contains lots of long transactions, so more items can be removed. The other two strategies are DLU (Discarding local unpromising items) and DGN (Decreasing local node utilities).We can efficiently generate PHUI’s from global Up-tree using these strategies.

They are used for the construction of the local Up-tree. Because individual items and their utilities are not maintained in the conditional pattern base. Here instead of maintaining exact utility values, minimum item utility table (MIUT) is kept.MIUT is same as the profit table (unit value). In DLU, the minimum item utilities unpromising items are eliminated from the path.DLN decrease minimum item utilities of descendant nodes. The purpose of DLN similar to DGN.

2.4 Top-K High Utility Mining

The burden of threshold setting arises a new area called top-k mining. In top-k no threshold is set externally. It is set internally by the algorithm itself. Initially the minimum utility threshold is set to zero. The threshold is set automatically as the work progresses. There are different methods for raising the threshold. Using k instead of minimum utility threshold helps in finding only top-k[1] sets of products that contribute highest profits to the company. Also an efficient way to find itemsets without setting minimum utility threshold. The challenges in top-k mining are:-

1) Neither monotone or anti-monotone
2) Incorporating the concept of top-k pattern mining with TWU model.
3) No pre-setting of minimum utility threshold.
4) Strategies for raising the minimum utility border threshold internally.

The main two algorithms using are TKU (Mining Top-K Utility Itemsets) and TKO(Mining Top-k Utility Itemsets in one phase)[1],[5],[6][8][9][10].Though TKU executes in two phases ,it is faster than TKO.Since more candidate elimination takes place in TKU.Also only need to check few of them.TKU uses global tree whereas TKO uses local tree. So TKU is ` better explained. In TKU UP-tree structure is used. It inherits properties from TWU model. In Phase I PKHUI’s are generated. Top-K HUI’s are identified from PKHUI’s in the Phase II. The UP-tree construction is explained earlier in this paper.

An example for top-k mining :– Let k =4 and absolute minimum utility(abs-min-util) = 0.From table 1 the absolute utility of each 1-itemsets is {{A}:16,{B}:16,{C}:13, {D}:20,{E}:15,{F}:5,{G}:14}.

Absolute utility (A) = Total occurrence (A) + Unit Profit.
Therefore absolute utility (A) = 4 × 4 = 16.
Absolute utility (D) = 10 × 2 = 20.

Similarly the above ones are calculated. The abs-min-util is raised to fourth highest value in top-k. Now abs-min-util become 14.So no top HUI’s will be missed. After raising abs-min-util TKU$_{Base}$ applies UP-growth algorithm with abs-min-util = minimum utility border(min-util$_{Border}$) to generate PKHUI’s.Potential top-k high utility itemset is an itemset if its estimated utility value(TWU) and MAU(Maximum Utility are no less than the min-util$_{Border}$ threshold. When four strategies are incorporated into TKU$_{Base}$ results in TKU algorithm. Two versions are developed for comparison purposes. The four strategies are:-

1) Pre-evaluation Step

PE matrix used to store lower bounds of the utilities of certain 2-itemsets.Initially PEM[x][y] is zero. After scanning all transactions, if k-th highest value in PEM is higher than min-util$_{Border}$,it can be raised to k-th highest value in PEM.This strategy can reduce the size of UP-tree and number of candidates produced in Phase I. Example pre-evaluation matrix for table 1 shown below:-

Table6: Pre-evaluation Matrix for Table1

Item	*B*	*C*	*D*	*E*	*F*	*G*
A	8	24	22	21	9	18
B		17	14	18	0	8
C			0	0	0	0
D				0	0	0
E					0	0
F						0

If min-util$_{Border}$ less than the k highest value in PEM, it is set to that. Here min-util$_{Border}$ becomes 18 if k=4.DGU can’t be applied, since min-util$_{Border}$ is set to 0 before UP-tree construction. If we apply PE strategy,DGU can be applied.

2) Raising threshold by Node utilities(NU)

The nodes should be not less than k in UP-tree construction and the k-th highest node utility in the UP-tree is higher than min-util$_{Border}$, then it is raised to that value. Example:-Consider first re-organized transaction; T$_1$’ = (C,1)(A,1)(D,1). The node utilities of C, AC,DAC in the path are respectively 1,6,8.Again in second transaction T$_2$’ = (C,6)(E,2)(A,2)(G,5).Now the node utilities for C, CE, CEA and CEAG are 6,12,22, 27.Since nodes not less than k,where k=4.The min-util$_{Border}$ is raised to 4th highest value i.e.,8.

3) Raising the threshold by MIU values of descendents (MD)

For each node under the root of Up-tree, algorithm traverses the subtree under the node to calculate the support count of of the itemset for every descendant node. For each itemset MIU value is calculated. If k-th highest MIU value is higher than min-util$_{\text{Border}}$,it can be raised to that one.

4) Raising the threshold during Phase II

Let C be candidates in Phase I which are sorted in descending order of estimated utilities. In Phase II utility of newly ones considered. If HUI(x) greater than min-util$_{\text{Border}}$,X and EU(X) inserted to a min-heap structure. HUI's in min-heap is in the decreasing order of utilities. Min-util$_{\text{Border}}$ is raised to k-th HUI from the min-heap.

Table6: Comparison of Different Strategies.

DGU	*DGN*	*DLU*	*DLN*
Applied before the construction of UP-tree	Applied with the construction of UP-tree	Applied after the construction of the UP-tree and before generation of PKHUI's	Applied during Phase II`

TKO uses utility list structure. Firstly utility list is calculated by scanning the original database twice. During first scan TWU and utility values are calculated. Items in each transaction are sorted in order of TWU values and utility-list of each item constructed in second scan. From this the itemsets generate utilities. In utility list structure each tuple has 3 fields:-Tid, iutil, Rutil.Tid is the identifier of transaction. Utility of the transaction is iutil. Rutil is the remaining utility. Here also there is TKO$_{\text{Base}}$ and TKO algorithms. Here also there is a heap structure for maintaining top-k HUI's in which itemset are sorted by descending order of utility. Initially the heap will be empty. Each itemset is processed such that if utility less than min-util$_{\text{Border}}$,it is added to the list(heap).If there are more than k-itemsets the min-util$_{\text{Border}}$ can be safely raised to that value. After that itemsets having min-util$_{\text{Border}}$ lower is eliminated from the list. The effective strategies for TKO are PE, DGU, RUZ and EPB.In this PE and DGU is already explained. Then RUZ means reducing estimated utility values using Z-elements (rutil = 0).EPB stands for exploring the most promising branches first.

3. Problem Definition and Proposed System

3.1 Problem Definition

Our idea is to incorporate the concept of closed itemset into top-k mining. In top-k mining when the transaction database is large we have to find space for storing the k values. This ideas will surely increase program efficiency an also saves space. Top-k concept have been already explained which include support of an item or an itemset, absolute utility, transaction utility, high utility itemset,TWU etc. An itemset is HTWUI[4] (High transaction weighted utilization itemset) iff its TWU greater than or equal to abs-min-utility.Comming to closed itemset mining; Tidset of an itemset is the set of Tids of transactions containing all itemsets.An itemset

I ε L is closed iff there exists no itemset J such that I subset or equal to J and SC(I) = SC(J).Mining frequent closed itemsets means itemsets having support no less than a user-specified threshold. Recovery of support is also possible. To incorporate the closed constraint into high utility itemset mining there are several strategies.First, define closure on the utility of itemets.Here a hui is said to be closed if it has no proper superset having the same utility. A second possibility is to define the closure on supports of itemsets either by mining all hui first, then apply the closed constraint or mine all the closed itemsets first and then apply the utility constraint.

3.2 Proposed System

After obtaining top-k hui's closed constraint applied to obtain a concise and compact representation. For top-k TKU algorithm used. Then passing the results to CHUD (Closed high utility itemset discovery) algorithm to obtain the final result. Recovery of support is possible using DAHU (Derive all high utility itemsets) from the set of closed hui's without accessing the original database. In CHUD vertical database is used. It will add more power to an existing system. One example for vertical database is MySql-Amazon RD (The cloud version of MySql).In vertical partitioning, divide a table into multiple tables that contain fewer columns. Two types of vertical partitioning are normalization and row splitting.DCI-closed is one of the best methods to mine closed itemets. CHUD outperforms it.

The CHUD[4] uses Itemset-Tidset pair tree(IT-tree) to find CHUI's.Itemset I; each node N(I),its Tidset g(X).Each node has two ordered sets of items PREV-SET(I) and POST-SET(I).Also there is estimated utility attached to each node. An itemset is maximal frequent if none of its immediate supersets is frequent. To make it lossless concept of utility unit array added. From Table 1; B is non-closed since $T_1 \cap T_2 \cap T_3 \cap T_4 \cap T_5 = \{AB\}$.The general procedure is to find hui's,then closed hui's and finally apply threshold to closed hui's.In CHUD also different strategies used to improve performance like REG(Removing the exact utilities of items from global TU-table),RML(Removing the mius of items from local TU-table), DCM(Discarding candidates with MAU that is less than the minimum utility threshold.

The CHUD first scans the database to convert to vertical database. Only promising items are kept, supersets of unpromising items are not CHUI's. Unpromising items can be removed from GTU table.CHUD creates a node and puts items upto the previous one into PREV-SET and items after it into POST-SET. Then Phase I executed. After that REG strategy applied. Finally the main procedure to obtain all CHUI's. The missed itemsets are also recovered. Sometimes there is a chance of missing itemsets.

4. Conclusion and Future Scope

The existing ones performs better but much improved performance using CHUD.Performance degrades when there are many hui's in the database. For large hui's execution time and memory used is more. Also when resources are limited, it is impractical to generate the entire set of hui's.So closed itemset mining is needed. In top-k more memory required for storing k-values. Implementing top-k high utility algorithms using CHUD algorithm improves the performance much better. Other than CHUD there are many other compact representations that are not combined with high utility mining. Although closed high utility itemset mining is essential to many research‘s and industrial applications, it is still a novel and challenging problem. The problem of top-k high utility itemset mining, where k is the desired number of high utility itemsets to be mined. The two efficient algorithms are TKO and TKU.The proposed algorithms have good scalability on large datasets and the performance of the proposed algorithms is close to the optimal case of the state-of-the-art two-phase and one-phase existing utility mining algorithms. Although we have proposed a new framework for 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 gives opportunity for exploration as future work.

References

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

[2] Vincent S. Teng, Cheng-Wei Wu, Bai-En Shie and Philip S.Yu, ”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, pp. 253-262, 2010.

[3] Jeff Heaton,” Comparing dataset characteristics that favor the Apriori, Eclat or FP-Growth frequent itemset mining algorithms”, in Proc. Of IEEE Trans. on Knowledge and Data Eng.,pp.200-213,2016.

[4] V. S. Tseng, C. Wu, P. Fournier-Viger, P. S. Yu, ”Efficient Algorithms for Mining the Concise and Lossless Representation of High Utility Itemsets”, in IEEE Transactions on Knowledge and Data Engineering, Vol 27,pp.726 - 739,2015.

[5] Vincent S.Tseng, Bai-En Shie,Cheng-Wei Wu and Philip S.Yu, ”Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases ", in IEEE Transactions on Knowledge and Data Engineering,Vol 25,pp.1772- 1786,2014.

[6] G. Pyun and U. Yun, “Mining Top-K Frequent Patterns with Combination Reducing Techniques ", Applied Intelligence,Vol.41,pp.76-98,2014.

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

[8] H. Ryang and U. Yun, “Top-K High Utility Pattern Mining with Effective Threshold Raising Strategies", Knowledge-Based Systems, Vol. 76, pp.109-126, 2015.

[9] C. Wu, B. Shie, V. S. Tseng and P. S. Yu, ”Mining Top-K High Utility Itemsets", in Proc. of the ACM SIGKDDInt'l Conf. on Knowledge Discovery and Data Mining, pp. 78-86, 2012.

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

[11] C. Lin, T. Hong, G. Lan, J. Wong and W. Lin, ”Efficient Updating of Discovered High utility Itemsets for Transaction Deletion in Dynamic Databases ",in Advanced Engineering Informatics,vol 29,pp.16-27, 2015.

Author Profile

Anu Augustin received the B.Tech degree from Caarmel Engineering College in 2006.Now an M.Tech student in Sahrdaya Engineering College, Kodakara. Doing research in data mining.

Dr.Vince Paul is the Head of the Department Sahrdaya Engineering College,Kodakara.He has PhD, M.E, PGDIT, MCSA.His area of interest is AI and neural network.