ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

Novel Algorithm for High Utility Itemset Mining: A Review

Nilesh Vani, Yamini Jawale

Asst. Professor, Dept of Computer, GF's G.C.O.E Jalgaon, Maharashtra, India

M.E Student, Dept of Computer, GF's G.C.O.E. Jalgaon, Maharashtra, India

ABSTRACT: High utility mining means mining or finding the high utility itemsets like the itemsets having high profit from the database. Efficiently mining high utility itemsets has very much importance in various applications. Many algorithms are developed for mining high utility itemsets, but these algorithms generate large number of candidate high utility itemsets. In this paper we are implementing an algorithm named Pattern-Mining Using Utility-Tree for high utility itemsets with an efficient tree data structure and a pruning strategy. Our algorithm generates less candidate itemsets as compared to existing high utility mining algorithms.

KEYWORDS: Potential High Utility Itemset (PHUI), **T**rans_Weighted_Util (**TWE) , T**rans_Util (**T**U).

I. INTRODUCTION

Frequent itemset mining finds the frequent items from a transactional database depending on the user specified minimum support threshold. But frequent itemset mining does not consider the weight and quantity of the item in a transaction. The high utility itemset mining is to find all itemsets that have utility larger than a user specified value of minimum utility. Recently, a utility mining model was defined to discover more important or profitable items from database. We can measure the importance of an itemset by using the concept of utility [1]. Utility can be thought of as an importance or profit of an item in database. By utility mining, several important business area decisions such as maximizing revenue, minimizing marketing or inventory costs can be considered and knowledge about itemsets/customers contributing to the majority of the profit can be discovered.

Mining high utility item sets from databases is very difficult because the downward closure property [1] infrequent itemset mining does not hold in utility mining. This means in high utility mining, pruning search space for high utility itemset mining is difficult because a superset of a low-utility itemset may be a high utility itemset. This problem can be solved using the principle of exhaustion it is possible to enumerate all itemsets from databases. But this method suffers from the problems of a large search space, especially when databases contain lots of long transactions or a low minimum utility threshold is set. Hence, how to effectively capture all high utility itemsets sand efficiently prunes the search space is the challenging task in utility mining.

Existing studies use the concept of overestimated methods to facilitate the performance of utility mining. In these methods, firstly the potential high utility itemsets (PHUIs) are found, and then an additional database scan is performed for finding their utilities. However, these existing methods generate a huge set of PHUIs and because of this their mining performance is degraded consequently.

A. Item	*B.* A	*C.* B	*D.* C	*E.* D	*F.* E	*G.* F	*H.* G	*I.* H
J. Profit	*K.* 5	*L.* 2	*M.* 1	*N.* 2	*O.* 3	*P.* 5	*Q.* 1	*R.* 1

Table 1: Profit table

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

S.	T_{id}	*T.* Transaction		*U.* Tu	
V.	T_1	*W.* (A,1) (C,10) (D,1)		*X.* 17	
Y.	T_2	*Z.* (A,2) (C,6) (E,2) (G,5)		*AA.*	27
BB.	T_3	*CC.* (A,2) (B,2) (D,6) (E,2) (F,1)		*DD.*	37
EE.	T_4	*FF.* (B,4) (C,13) (D,3) (E,1)		*GG.*	30
HH.	T_5	*II.* (B,2) (C,4) (E,1) (G,2)		*JJ.* 13	
KK.	T_6	*LL.* (A,1) (B,1) (C,1) (D,1) (H,2)		*MM.*	12

Table 2: Database Example

To address this issue, we propose a novel algorithm which uses a compact data structure for efficiently discovering high utility itemsets from transactional databases. Various Standard and Synthetic datasets are used with real data set for educational feedback system.

II. BACKGROUND

A. Problem Definition

Let I= {i1, i2, i3, . . . , in} be a set of items. Each item i_p ($1 < p < m$) in the set has a unit profit pr (i_p). An itemset X is a set of j distinct items $\{i_1, i_2, \ldots, i_j\}$, where $i_k \in I$; $1 \leq k \leq j$. j is the length of X. A transaction database $D = \{T_1, T_2, \ldots, T_n\}$ contains a set of transactions, and each transaction T_{id} has a unique identifier d, called TID. Each item i_p in transaction T_{id} is associated with a quantity Tq (i_p, T_{id}), that is, the purchased quantity of an item in that transaction.

Utility of an item i_p in a transaction $T_{id}$is denoted as $u(i_p, T_{id})$ and defined as $pr(i_p) \times Tq(i_p, T_{id})$.

For example, in Table 2, $u(\{C\}, T1) = 1 \times 10 = 10$.

Let an itemset X be a subset of I. The utility of X in transaction T_{id}, denoted by $u(X, T_{id})$ is defined as: $u(X, T_{id}) = \sum_{ip \in X}(i_p, T_{id})$.

For example, $u(\{AC\}, T1) = u(\{A\}, T1) + u(\{C\}, T1) = 5 + 10 = 16$.

Utility of an itemset X in D is denoted as u(X) and defined as: $u(X) = \sum_{X \subseteq Tid \wedge Tid \in D} u(X, T_{id})$. $u(\{AD\}) = u(\{AD\}, T1) + u(\{AD\}, T3) + u(\{AD\}, T6) = 7 + 22 + 7 = 36$.

High utility itemset mining is to find all itemsets that have utility value above a user-specified *minutil* threshold. Since utility is not *anti-monotone*, Liu et al. [2] proposed the concepts of Transaction Utility i.e. trans_util(tu) and Transaction Weighted Utility i.e. trans_weighted_util (twu) toprune the search space of high utility itemsets.

Transaction Utility (tu) of a transaction, denoted *trans_util*(T_{id}) isthe sum of the utilities of all items in T_{id}. This is defined as: $tu(T_{id}) = \sum_{ip \in Tid} u(i_p, T_{id})$.

For example, $tu(T1) = u(\{ACD\}, T1) = 17$.

Transaction Weighted Utility (twu) of an itemset$X$, denotedas *twu(X)* is the sum of the transaction utilities of all the transactions containing the itemset X: $twu(X) = \sum_{X \subseteq Tid \wedge Tid \in D} tu(T_{id})$.

$twu(\{AD\}) = tu(T1) + tu(T3) + tu(T6) = 17 + 37 + 12 = 66$.

B. Related Work

In data mining, the association-rule mining techniques [1],[2]are frequently used to find useful rules or patterns in various applications, such as supermarket promotion, online e-commerce management. But, traditional association-rule mining only considers the occurrence relationship of items in transactions. The same significance is also assumed for all the items. Hence, the importance of items in a database cannot be easily found out. In reality, the importance of items should be different according to the factors, such as type, profits and costs of the items. Hence, some items with high-importance may not be discovered by using earlier association-rule mining techniques. To address this issue, high utility itemset mining technique was developed. This technique considers an itemset as an important itemset by considering their quantity and unit profit value. Many algorithms have been proposed for high utility itemset [2], [3], [5] mining but all they first produce large number of candidate itemset. By applying the proposed algorithm, the set of candidate itemsets generated during the generation of high utility itemsets are reduced effectively and the high utility itemsets are generated efficiently.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

III. PROPOSED METHOD

The aim of high utility itemset mining is to discover all the high utility itemsets from a transactional database D whose utility values are higher than user specified threshold. The framework of the proposed methods consists of three steps: 1) Perform two databases cans to construct the Utility-Tree 2) Generate Candidate High Utility Itemsets from Utility-Tree and 3) identify actual high utility itemsets from the set of Candidate High Utility Itemsets. We use a Header Table to store the items and their transaction weighted utilities in descending order of items trans_weighted_util values. We also use the Conditional Pattern Bases (CPB) to store path utilities of itemsets after removing the unpromising items.

A. The Proposed Algorithm

The algorithm Pattern Mining Using Uitlity-Tree is one of the efficient algorithms to generate high utility itemsets depending on construction of a Uitlity-Tree. Initially for each transaction in a database transaction utility (trans_util) is calculated. Also the transaction weighted utility (trans_weighted_util) of each itemset X is calculated. If the trans_weighted_util is less than the user specified minute value, then the itemset is considered as an unpromising itemset and removed from each transaction in the database. The transactions are reorganized according to descending order of trans_weighted_util in header table. Utility-Tree is formed from Dataset by taking input Dataset using Transaction table. PHUs (Potential High Utility Itemsets) are obtained from Utility-Tree and two strategies are applied on Utility-Tree to reduce unpromising items from obtained PHUs.

1. *Discarding global unpromising items (DGU):* The unpromising items and their actual utilities are eliminated from the trans_utils during the construction of a global Utility-Tree.
2. *Discarding global node utilities (DGN):* For any node in a global Utility-Tree, the utilities of its descendants are discarded from the utility of the node during the construction of a global Utility-Tree.

After applying the DGU and DGN strategies on the Utility-Tree, the tree contains only the global promising items. From this reorganized tree, High Utility Itemsets are found out. A basic method for generating PHUIs from the Utility-Tree is to mine the Utility-Tree by using FP-Growth method [1]. The generated candidate itemsets can be reduced by applying two strategies.

3. *Discarding local unpromising items (DLU):*

The minimum item utilities of unpromising items are discarded from path utilities of the paths during the construction of a local Utility-Tree.

The Conditional Pattern Bases (CPB) are generated during stage and the path utilities are inserted. Path utility of a path in the items CPB is utility of the item at the leaf node of that path. By using the DGU strategy, in a conditional pattern Utility-Tree, local unpromising items and their utilities can be discarded.

4. *Decreasing local node utilities (DLN):*

The utilities of descendant nodes for the each node are decreased during the construction of a local Utility-Tree. After applying these two strategies DLU and DLN the High Utility Itemsets can be found out from the Conditional Utility-Tree of each itemset.

B. Algorithm Pattern Mining Using Utility-Tree

Input: Transaction database D, user specified threshold *min util*.

Output: high utility itemsets.

Steps:

1. Scan the transactional database for each transaction T ϵ D
2. Read user defined threshold value
3. Determine transaction utility (trans_util) of T in D and Transaction weighted utility (trans_weighted_util) of itemset (X)
4. If (trans_weighted_util (X) $\leq$ minutil) then

i) Remove X from each transaction

ii) Reorganize transaction in descending order of their trans_weighted_util

5. Else insert item X into header table H and keep the items in descending order of trans_weighted_util.
6. Repeat step 4 & 5 until end of the D.
7. Insert each transaction T into global utility Tree
8. Apply DGU and DGN strategies on global utility tree
9. Re-construct Tree after eliminating unpromising items
10. For each item ai in Header Table H do

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

i) Generate a PHUI Y= X U ai
ii) Estimate utility of Y is set as ai's utility value in H
iii) Put local promising items in Y-CPB into H
iv) Apply strategy DLU and DLN locally on path to reduce path utilities
11. End for loop
12. End

IV. CONCLUSION

In this paper, I have proposed an efficient algorithm named Pattern Mining Using Utility-Tree for mining high utility itemsets from transactional databases. Utility-Tree structure is proposed for maintaining the information of the high utility itemsets. Hence, the potential high utility itemsets can be efficiently generated from the Utility-Tree with only two database scans. We have used four strategies to decrease the estimated utility value and enhance the mining performance in utility mining.

REFERENCES

1. R. Agrawal, T. Imielinksi, and A. Swami, "*Mining Association Rules between Sets of Items in Large Database*," in Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 207-216, 1993.
2. Y. Liu, W.-K. Liao, and A. Choudhary, "*A FastHigh Utility Itemsets Mining Algorithm*", Proc.UBDM'05, Chicago Illinois, 2005.
3. Alva Erwin, Raj P. Gopalan, N.R. Achuthan, "*CTU-Mine: An Efficient High Utility Itemset Mining Algorithm Using the Pattern Growth Approach*", 7thInternational Conference on Computer and Information Technology.
4. ChowdhuryFarhan Ahmed, Syed KhairuzzamanTanbeer, Byeong-SooJeong, and Young-Koo Lee, "*Efficient Tree Structures for High Utility Pattern Mining in Incremental Databases*", IEEE Transactions on Knowledge And Data Engineering, Vol. 21, No. 12, December 2009.
5. Vincent S. Tseng, Bai-En Shie, Cheng Wei Wu, and Philip S. Yu, Fellow, "*Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases*" IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 25, NO. 8, AUGUST 2013.