

Mining High Utility Item sets From Transaction Database

More Rani N.

*Department of Computer Engineering
S.V.P.M.COE Malegaon (B.K.), Baramati, Maharashtra, India*

Anbhule Reshma V.

*Department of Computer Engineering
S.V.P.M.COE Malegaon (B.K.), Baramati, Maharashtra, India*

Waghmare Archana B.

*Department of Computer Engineering
S.V.P.M.COE Malegaon (B.K.), Baramati, Maharashtra, India*

Abstract- Mining high utility item sets from a transactional database means to retrieve high utility item sets from database. Here, high utility item sets are the item sets which have highest profit. In existing system number of Algorithm's have been proposed but there is problem like it generate huge set of candidate Item sets for High Utility Item sets. If database contains large number of Transactions then it degrades the performance of mining in terms of execution time and space requirement. In Our proposed system, we propose Efficient Algorithm for Mining High Utility Item sets From Transactional Database i.e. UP-Growth Algorithm. For that algorithm information of high utility item sets is maintained in tree based data structure named Utility Pattern Tree. With the help of UP-Tree candidate item sets can be generated with only two scans of database. In first scan, Transaction Utility (TU) of each transaction is calculated. At the same time Transaction Weighted Utility (TWU) of each single item is also calculated. In second scan, transaction is inserted into UP-Tree. Proposed algorithm, not only reduce number of candidate item sets but also work efficiently when database contains lot's of long transactions..

Keywords – Data mining, candidate item sets, high utility item sets, UP-Tree

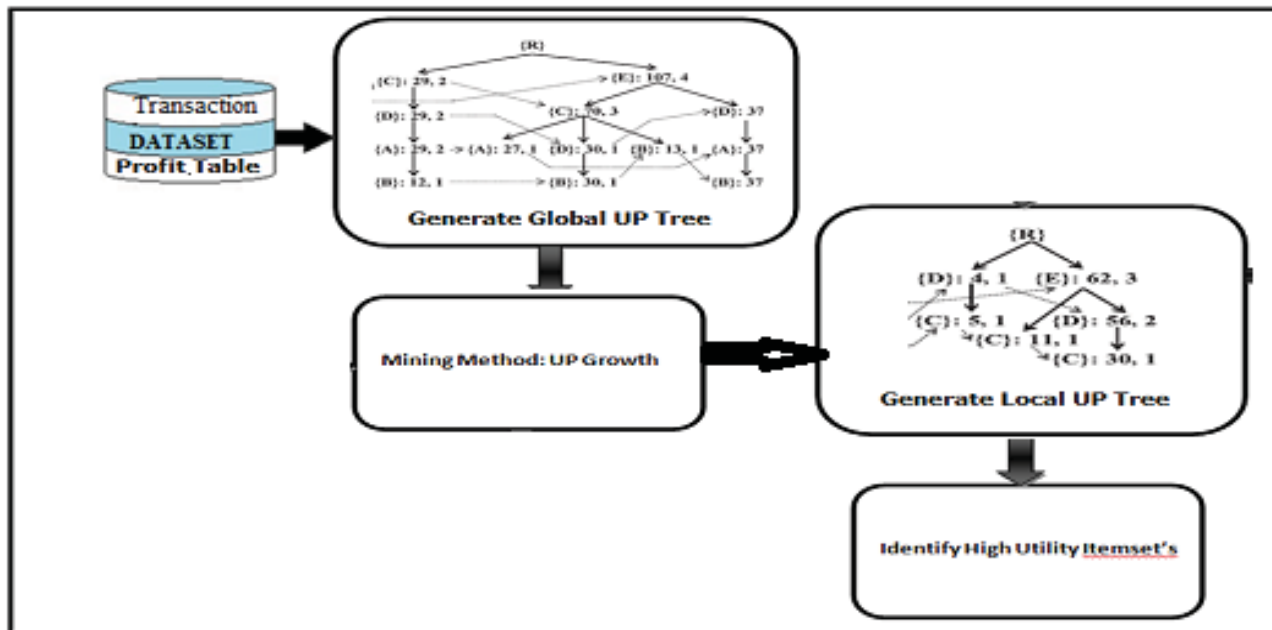
I. INTRODUCTION

The purpose of our proposed systems is towards finding high utility item set. Data mining is the process of retrieving item sets from database. Here, the meaning of item set utility is interestingness, importance, or profitability of an item to users. An item set is called a high utility item set if its utility is no less than a user-specified minimum utility threshold; otherwise, it is called a low-utility item set. The goal of frequent item set mining is to find items that co occur in a transaction database above a user given frequency threshold, without considering the quantity or weight such as profit of the items. However, quantity and weight are significant for addressing real world decision problems that require maximizing the utility in an organization. The high utility item set mining problem is to find all item sets that have utility larger than a user specified value of minimum utility. In existing System, HUP Algorithm is used to mining High Utility Item sets from database but there are some disadvantages like, it generates huge set of PHUIs. In our project we use UP-Growth Algorithm. Main advantages of this Algorithm are, it scan database only two times and it generates less set of PHUIs.

The rest of the paper is organized as follows. Proposed algorithms are explained in section II. Experimental results are presented in section III. Concluding remarks are given in section IV.

II. PROPOSED ALGORITHM

System Architecture:



- Firstly, store transaction dataset and profit table into database.
- Generate Global UP-Tree by using DGU and DGN.
- Use UP Growth Algorithm for mining HUIs.
- Generate Local UP Tree by using DLU and DLN.
- Identify the HUIs.

Precondition:

TID	Transaction
T_1	(A,1)(C,1)(D,1)
T_2	(A,2)(C,6)(E,2)(G,5)
T_3	(A,1)(B,2)(C,1)(D,6)(E,1)(F,5)
T_4	(B,4)(C,3)(D,3)(E,1)
T_5	(B,2)(C,2)(E,1)(G,2)

Items and their unit profits							
Item	A	B	C	D	E	F	G
Unit Profit	5	2	1	2	3	1	1

- Select Transaction dataset and profit table.
- Calculate Transaction Utility of Item set(TU) by using following formula

$$TU(T_d) = u(T_d, T_d)$$

Algorithms:

A. Discarding Global Unpromising Items algorithm –

- Compute Transaction Weighted Utility (TWU) of Item sets. e.g. $TWU(A) = u(T_1, T_1) + u(T_2, T_2) + u(T_3, T_3)$
- Remove unpromising items from each transaction, Rearrange items in a descending order of TWU and calculate Reorganized transaction utility.e.g. $RTU(T_2) = \text{Previous } TU(T_2) - [Q(G) * P(G)]$

B. Decreasing Global Node utilities algorithm –

- Select the items from which item our transaction is started.
- Insert these items into Up-Tree under root node.
- From these node select next items and insert it into UP-Tree

- Then recursively call above steps until all nodes inserted into Tree.
- In this way construct the Global UP-Tree.

C. *UP-Growth algorithm* –

- Trace each node related to specific item via link and calculate sum of node utility, start from leaf node.
- If sum is less than minimum threshold then remove that node from tree.
- If sum is greater than minimum threshold then calculate highest utility node, select that node and generate path from this node.
- Apply DLU to reduce path utilities of the paths.
- Insert this reorganized path into tree by using DLN.
- If node from local tree is not completed then recursively call this method.

D. *Discarding Local Unpromising items algorithm* –

- Compute local unpromising items.
- Remove local unpromising items from the path and recalculate path utility.

E. *Decreasing Local Node utilities algorithm* –

- Construct Local UP-Tree.
- Calculate node utility.
- Identify High Utility Item sets.

III. EXPERIMENT AND RESULT

The experiments were performed on a 2.60 GHz Intel Pentium D Processor with 3.4 GB memory. The operating system is Microsoft Windows 7. The algorithms are implemented in Java language.

The proposed system is tested using foodmart transaction dataset. Following are some reasons why our system outperforms the state of the art algorithms:-

1. Utilities of nodes in the global UP-Tree are much less than TWUs of nodes in IHUP-Tree because DGU and DGN algorithms effectively decrease overestimated utilities while constructing the global UP-Tree.
2. UP-Growth algorithm generates less number of candidate itemsets than FP-Growth algorithm because UP-Growth algorithm uses DLU and DLN methods for constructing local UP-Tree.

By the above mentioned reasons, the proposed algorithm UP-Growth achieves better performance than IHUP and FP algorithm.

IV. CONCLUSION

In this system, we propose a tree-based algorithm, called UP-Growth, for efficiently mining high utility item sets from transactional databases. We take Data Structure UP-Tree for maintaining the information of high utility item sets and four effective strategies, DGU, DGN, DLU and DLN, to reduce search space and the number of candidates for utility mining. PHUIs can be efficiently generated from UP-Tree with only two database scans. UP Growth Algorithm is faster than existing algorithms when database contains lots of long transactions.

REFERENCES

- [1] 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 Transaction on knowledge and data engineering, vol. 25, no. 8, Aug 2013.
- [2] C.F. Ahmed, S.K. Tanbeer, B.-S. Jeong, and Y.-K. Lee, "Efficient Tree Structures for High Utility Pattern Mining in Incremental Databases", IEEE Trans. Knowledge and Data Eng., vol. 21, no. 12, pp. 1708-1721, Dec. 2009.
- [3] S.K. Tanbeer, C.F. Ahmed, B.-S. Jeong, and Y.-K. Lee, "Efficient Frequent Pattern Mining over Data Streams", Proc. ACM 17th Conf. Information and Knowledge Management, 2008.
- [4] B.-E. Shie, V.S. Tseng, and P.S. Yu, "Online Mining of Temporal Maximal Utility Itemsets from Data Streams", Proc. 25th Ann. ACM Symp. Applied Computing, Mar. 2010.
- [5] J. Pisharath, Y. Liu, B. Ozisikyilmaz, R. Narayanan, W.K. Liao, A. Choudhary, and G. Memik NU-MineBench Version 2.0 Data Set and Technical Report, <http://cucis.ece.northwestern.edu/projects/DMS/MineBench.html>, 2012.