AN EFFECTIVE WAY OF MINING HIGH UTILITY ITEMSETS FROM LARGE TRANSACTIONAL DATABASES

[1]Chadaram Prasad, [2]Dr. K..Amarendra
[1]M.Tech student, Dept of CSE, [2]Professor & Vice Principal,
DADI INSTITUTE OF INFORMATION & TECHNOLOGY
NH-5,Anakapalli,Visakhapatnam-531002,Anshrapradesh,India.

---***---

Abstract- *The main aim of this project is to generate the high utility (profitable) itemsets in a parallel environment, by considering the both profit and quantity of each item. The parallel mining of high utility itemsets will take very less time than mining with the single system over large number of transactions. A Utility-pattern tree(UP_tree) data structure (Memory-Resident) is used to store the information about the transactions. By UP_Growth algorithm the candidate itemsets are generated by scanning the database only twice. The above Utility-Pattern tree is duplicated at every node in the parallel system and at each node the conditional pattern base tree is constructed for the assigned high utility itemsets, from this tree mined the high utility itemsets according to the given minimum utility threshold. The performance of UP_Growth algorithm is observed by applying on the synthetic dataset.*

Keywords: Utility Mining, Quantitative Database.

I. INTRODUCTION

Data mining is the extraction of potentially useful information from the available data. Discovery of the knowledge is the data mining goal, which is used to predict the feature behavior for taking the structural decisions. Mining the frequent patterns is a former research topic in data mining. Frequent pattern mining is the finding of interesting patterns hidden in a database. Frequent pattern mining doesn't consider the profit for each item. To overcome this problem weighted frequent pattern mining [2] is proposed. Where as in weighted frequent pattern mining doesn't consider the quantity of each item. To overcome this problem utility mining is proposed. Utility mining consider the both profit and quantity of each item. Utility mining supports the quantitative database.

Utility mining finds the most valuable frequent itemsets. The Utility of an item refers to the users interestingness for item. The multiple of itemsets external and internal utilities defines it's utility. External utility is the profit of the item, internal utility is quantity of the item present in the transaction. For the given utility threshold if the determined itemset utility is greater than the given utility threshold then it is called promising itemset else it is called unpromising itemset. The process of mining high utility itemsets requires two inputs first one is transactional database and second one is profit for each item as given from table 1 and table 2.

II. BACKGROUND

Definition 1: The itemset X utility in

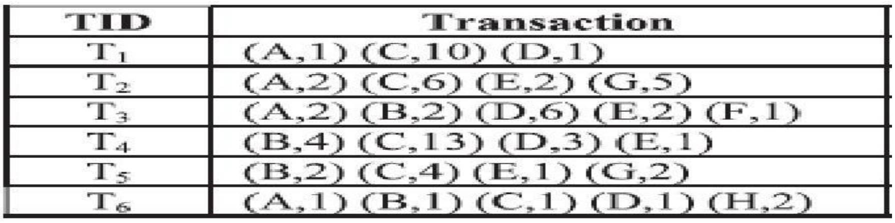
D DataBase is represented by

U(X) = $\sum_{X \subseteq T_d \wedge T_d \in D} u(X, T_d).$

Definition 2: For the D database the itemset X Transaction weighted utility is

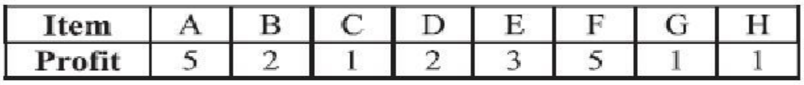
TWU(X)=$\sum_{X \subseteq T_d \wedge T_d \in D} \mathrm{TU}(T_d).$

TABLE 1
An Example Database



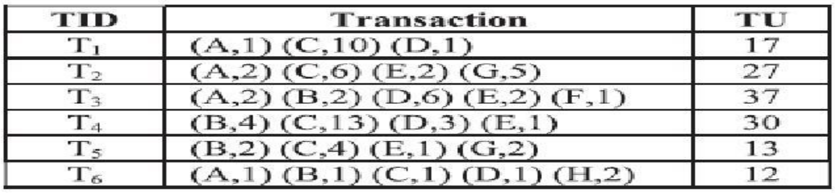
TID	Transaction
T_1	(A,1) (C,10) (D,1)
T_2	(A,2) (C,6) (E,2) (G,5)
T_3	(A,2) (B,2) (D,6) (E,2) (F,1)
T_4	(B,4) (C,13) (D,3) (E,1)
T_5	(B,2) (C,4) (E,1) (G,2)
T_6	(A,1) (B,1) (C,1) (D,1) (H,2)

TABLE 2
Profit Table



Item	A	B	C	D	E	F	G	H
Profit	5	2	1	2	3	5	1	1

TABLE 1
An Example Database



TID	Transaction	TU
T_1	(A,1) (C,10) (D,1)	17
T_2	(A,2) (C,6) (E,2) (G,5)	27
T_3	(A,2) (B,2) (D,6) (E,2) (F,1)	37
T_4	(B,4) (C,13) (D,3) (E,1)	30
T_5	(B,2) (C,4) (E,1) (G,2)	13
T_6	(A,1) (B,1) (C,1) (D,1) (H,2)	12

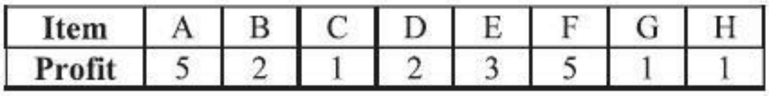


TABLE 2
Profit Table

Item	A	B	C	D	E	F	G	H
Profit	5	2	1	2	3	5	1	1

Transaction weighted downward closure property: Let k itemset is Ik, (k-1)-itemset is a Ik-1 then Ik-1 ⊂ Ik.

Since $I^{k-1} \subset I^k$, $T_{I^{k-1}}$ is a superset of T_{I^k} .

Since $I^{k-1} \subset I^k$, $T_{I^{k-1}}$ is a superset of T_{I^k} .

$$twu(I^{k-1}) = \sum_{I^{k-1} \subseteq T_q \in D} tu(T_q) \geq \sum_{I^k \subseteq T_p \in D} tu(T_p) = twu(I^k) \geq \varepsilon'$$

When mining the high utility itmsets from very large transactions takes much time, this mining time can be reduced by implementing in a parallel environment.

III. RELATED WORK

For mining the frequent itemsets basically Apriori Algorithm [1] is used. The disadvantage of this algorithm is it requires too many scans of the entire database and it generates test the each candidate itemset knowing for promising or unpromising itemset. To overcome this Frequent pattern growth (FP_Growth)[3] algorithm is proposed. High utility itemsets are mined by Up_Growth algorithm and it is a tree-based approach as same as FP-Growth[3] algorithm. Up_Growth algorithm requries only two scans of database[4] for creating the Up_tree from this the large utility itemsets are mined. For mining the interesting itmesets first Up_tree is created then the conditional pattern base (CPB) trees are derived for each high utility item assigned to the corresponding each parallel node. From each parallel node the corresponding high utility items are mined by using the Up_Growth algorithm.

For constricting the Up_Tree the node structure is

```
Class N
{
    String name;
    int count;
    int utility;
    String parent;
    Class N hlink;
}
```

here

N.name refers to item name of the node.

N.count refers to frequency count.

N.utility refers to the node utility.

N.parent refers to the parent node.

N.hlink refers to the node which is same as the N.name node.

Header table is useful to traverse the tree. The structure of the header table is

ITEMNAM	TWU	LINK

The link pointer is used to represent the last appearance of the node in UP_Tree which has same node name in the entry header table.

3.1 Steps for Construction of UP_Tree:

- Database is scanned only twice for Construction of UP_Tree.
- First scan
- For each transaction the transaction utility (Tu) is computed.
- At the same time the transaction weighted utility(TWU) of each item is also computed.
- The items with TWU's less than the given threshold then those items are called unpromising items.Discard these items.
- For the promising items compute the retrasacntion utilities.
- The promising items are arranged in the decreasing order of their TWU's.

- Second scan
- Each transaction is inserted as a branch into a UP_Tree.

After construction of UP_Tree the following strategies are used to reduce the time and the search space. The strategies are as follows

3.2 Strategy 1: Discarding global unpromising items (DGU).

Since the unpromising items does not involved in the high utility itemsets, these are not consider in the creation of Global UP_tree.

3.2 Strategy 2: Discarding global node utilities (DGN).

In the construction of a Global Utility pattern tree for a node the successor utilities are deleted from the current utility of the node.

3.2 Strategy 3: Discarding local unpromising items (DLU).

In the construction of a local Utility pattern tree the unpromising items minimum utilities are dropped from path utilities of paths.

3.2 Strategy 4: Decrease the local node utilities (DLN).

For construction of a local UP_tree for any node the item's minimum utilities of descendant's are decreased.

The pseudo code for UP-Grwoth is as follows:

```
Method: U-PGrow(Ax, Bx, Z)
Input: Ax as UP_Tree, Bx header table for Ax and Z itemset.
Output: PHUI's in given tree.
Method: U-PGrow (Ax, Bx, Z)
1. For every ci in Bx do
2. Create PHUI P = Z ∪
3. Assign P's utility= utility of ci's in Bx.
4. Create P's CPB;
5. Place local items which are having greater min_util in P-CPB into By.
6. Implement DLU.
7. Implement DLN, add branches to Ay;
8. If Ay! = { } call U-PGrow (Ay, By, P);
9. End of for.
```

Utility Pattern Tree is used for maintaining the information about the transactions. From the UP_Tree the PHUI's are generated. UP_Tree will be replicated at each parallel node and the corresponding the CPB (Conditional Pattern Base) trees are generated at each node for all the items in the header table. For each items CPB's the high utility itemssets are mined.

The UP_Tree constructed as follows:

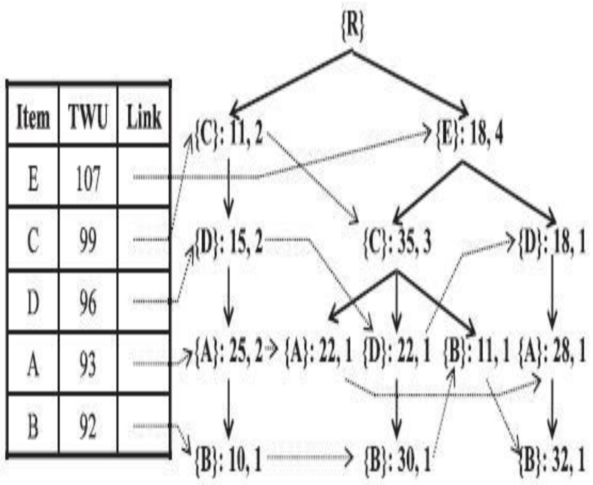


Fig 1: the UP_Tree for table 1

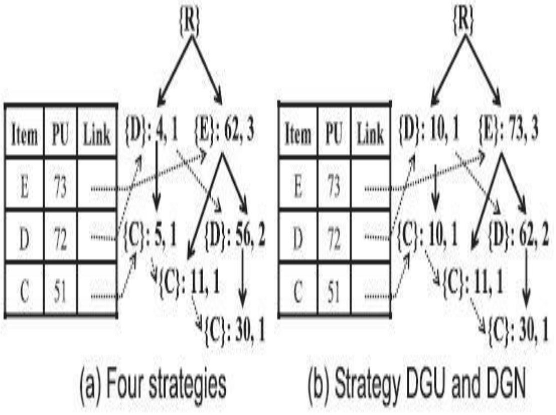


Fig: {B}conditional Pattern base tree

IV. EXISTING SYSTEM

- Utility mining considers the both quantity of items purchased along with it’s profit.

4.1 Disadvantages:

- Mining the high utility itemsets takes much time when the database is very large.
- Construction of Up_tree is also complex.

V. PROPOSED SYSTEM

In the proposed system the mining of high utility itemsets will done in parallel. Here the construction of Up_tree will done at server and the high utility itemsets are mined at each parallel node from its conditional pattern base tree’s.

5.1 Advantages

Parallel mining of high utility itemsets will take less time when compared to mining in a single system.

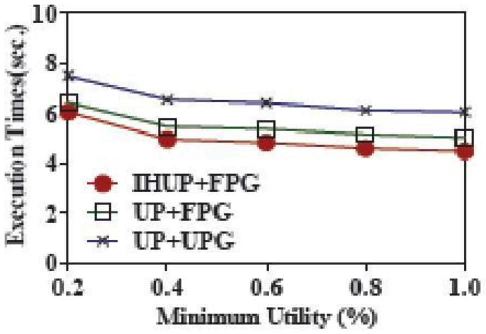
High utility itemsets are mined from each parallel node.

VI. EXPERIMENTAL EVOLUTIONS

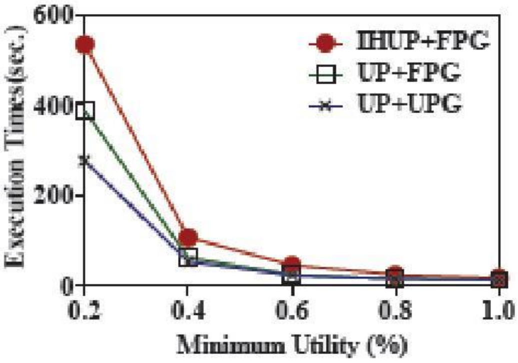
The new algorithms performance is examined in the parallel environment. Client-server environment is created by using the Apache Tomcat server. The performance will be checked by applying the algorithm in the Synthetic dataset [1].

. Number of candidates on T10I6D100K

Minimum utility	IHUP+FPG	UP+FPG	UP+UPG
0.2%	20,651	15,057	10,664
0.4%	4,003	2,347	1,990
0.6%	1,684	910	844
0.8%	873	527	521
1.0%	566	411	411



Execution time for phase I on T10I6D100K.



The performance of mining high utility itemsets is rapidly increased in the parallel environment by minimizing the memory space and the generation of unwanted candidate itemsets. The new methods will give a batter performance for the large Databases and for the low utility thresholds.

VII. CONCLUSIONS

This paper proposes a parallel system which takes very less turnaround time for mining large utility itemsets.Converting the transactions into UP_tree two scans are sufficient. Conditional pattern based tress for each high utility item set is replicated at each parallel node. From all the parallel nodes we get the high utility itemsets. with this we can also create a large synthetic datasets.

REFERENCES

[1]. Agrawal and Srikant, “Fast Algorithms for Mining Association Rules,”

[2]. C.H. Cai, A.W.C. Fu, C.H. Cheng, and W.W. Kwong, “Mining Association Rules with Weighted Items,”

[3]. C.F. Ahmed, S.K. Tanbeer, B.-S. Jeong, and Y.-K. Lee, “Efficient Tree Structures for High Utility Pattern Mining in Incremental Databases,”.

[4]. “A Two-Phase Algorithm for Fast Discovery of High Utility Itemsets” Ying Liu, Wei-keng Liao, and Alok Choudhary