Many algorithms have been proposed that have problem of producing a large number of candidate itemsets for high utility itemsets. Such a large number of candidate itemsets degrades the mining performance in terms of execution time and space requirement. This situation is difficult when the database contains lots of long transactions or long high utility itemsets. In this paper, we propose two algorithms, namely utility pattern growth (UP Growth) and UP-Growth+, for mining high utility itemsets with a set of effective strategies for pruning candidate itemsets. Information of high utility itemsets maintained in Up-tree, candidate itemsets can be generated efficiently with only two scans of database. Experimental results show that the proposed algorithms, especially UP Growth+, not only reduce the number of candidates effectively but also outperforms other algorithms substantially in terms of runtime, especially when databases contain lots of long transactions.

Keywords: Data Mining, high utility itemset, utility mining

I. INTRODUCTION

Data mining and knowledge discovery from data bases has received much attention in recent years. Data mining, 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. Knowledge Discovery in Databases (KDD) is the non-trivial process of identifying valid, previously unknown and potentially useful patterns in data. These patterns are used to make predictions or classifications about new data, explain existing data, summarize the contents of a large database to support decision making and provide graphical data visualization to aid humans in discovering deeper patterns. 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. 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 bioinformatics, Web click-stream analysis, and mobile environments. 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:

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

Utility of an itemset is defined as the product of its external utility and its internal utility. An itemset is called a high utility itemset. If its utility is no less than a user-specified minimum utility threshold; otherwise, it is called a low-utility itemset.

Here we are discussing some basic definitions about utility of an item, utility of itemset in transaction, utility of itemset indatabase and also related works and define the problem of utility mining and then we will introduce related strategies. Given a finite set of items I= {i1, i2, i3…ik}, each item ip(1≤p≤m) has a unit profit pr(ip). An itemset X is a set of k distinct items I= {i1, i2,i3…ik}, where 1≤i≤k. k is the length of X. An itemset with length k is called a k itemset. A transaction database D = {T1; T2; . . . ; Tn} contains a set of transactions, and each transaction Td (1≤d≤n) has a unique identifier d, called TID. Each item ip in transaction Td is associated with a quantity q (ip,Td), that is, the purchased quantity of ip in Td.

Definition 1: Utility of an item ip in a transaction Td is denoted as u (ip, Td) and defined as pr(ip)×q(ip,Td)
Definition 2: Utility of an itemset X in Td is denoted as U(x, Td) and defined as Σip∈Xu(ip,Td)
Definition 3: Utility of an itemset X in D is denoted as u(X) and ΣXu⊆Td⊆D u(X,Td)
Definition 4: An itemset is called a high utility itemset if its utility is no less than a user-specified minimum utility threshold or low-utility itemset represented by min-util.
Table 1: An Example Database

<table>
<thead>
<tr>
<th>TID</th>
<th>Transaction</th>
<th>TU</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>(A,1) (C,10) (D,1)</td>
<td>17</td>
</tr>
<tr>
<td>T2</td>
<td>(A,2) (C,6) (E,2) (G,5)</td>
<td>27</td>
</tr>
<tr>
<td>T3</td>
<td>(A,2) (B,2) (D,6) (E,2) (F,1)</td>
<td>37</td>
</tr>
<tr>
<td>T4</td>
<td>(B,4) (C,13) (D,3) (E,1)</td>
<td>30</td>
</tr>
<tr>
<td>T5</td>
<td>(B,2) (C,4) (E,1) (G,2)</td>
<td>13</td>
</tr>
<tr>
<td>T6</td>
<td>(A,1) (B,1) (C,1) (D,1) (H,2)</td>
<td>12</td>
</tr>
</tbody>
</table>

Table 2: A Profit Table

<table>
<thead>
<tr>
<th>Item</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

From table 1 and 2

\[ U(\{A,T1\}) = 5 \times 1 = 5 \]
\[ \sum U(\{A,T1\}) + U(\{D,T1\}) = 5 + 2 = 7 \]
\[ U(\{AD\}) = U(\{AD, T1\}) + U(\{AD, T3\}) = 7 + 17 = 24 \]
\[ U(\{BD\}) = U(\{BD, T3\}) + U(\{BD, T4\}) = 16 + 18 = 34 \]

II. MOTIVATION

Our motivation for this project was to application spectrum is wide in many real-life applications and is an important research issue in data mining area. Utility mining emerges as an important topic in data mining field. Here high utility item sets mining refers to importance or profitability of an item to users. Number of algorithms like Apriori (level – wise search) has been proposed in this area, they cause the problem of generating a large number of candidate itemsets. That will lead to high requirement of space and time and so that performance will be less. It is not at all good when the database contains transactions having long size or high utility itemsets which also having long size. Mining high utility item sets from databases refers to finding the itemsets with high profits. Here, the meaning of item set utility is interestingness, importance, or profitability of an item to users.

III. LITERATURE REVIEW

Agarwal et al developed an algorithm for mining association rules between sets of items in large databases. Association rule mining are if/then statements that helps to uncover relationships between seemingly unrelated data in a relational database or other information repository. Apriori Association rule mining technique uses a two-step process. The first step is to identify all the frequent itemsets based on the support count value of the itemsets. It uses the download closure property of itemsets to remove the infrequent itemsets. The second step is the generation of association rules from the frequent itemsets using the support and confidence [1].

Han j et al developed an algorithm for mining frequent patterns without candidate generation. In this framework of frequent itemset mining, the importance of items, profit and purchased quantities of items are not considered. Frequent itemset may only contribute a small portion to the overall profit, and non-frequent itemset may contribute a large portion to the profit. Fp-growth improves the efficiency of frequent mining as it does not generate candidate itemsets during the mining process. The drawback of this approach is that it considers only the important items in the frequent pattern [2].

Y. Liu, W-K. Liao, A. Choudhary proposed a two phase algorithm which was developed to find high utility itemsets, using the download closure property of apriori. The algorithms have defined the transaction weighted utilization (twu) while maintaining the download closure property. In this paper they defined two database scans. In the first database scan, the algorithm finds all the one element transaction weighted utilization itemsets and its results form the basis for two element transaction weighted utilization itemsets. In the second database scan, the algorithm finds all the two element transaction weighted utilization itemsets and it results form the basis for three element transaction weighted utilization itemsets. The drawback of this algorithm is that it suffers from level wise candidate generation and test methodology [3].

J Hu et al developed an algorithm for frequent item set mining that identify high utility item combinations. The goal of this algorithm is to find segments of data, defined through combinations of some items (rules), which satisfy certain conditions as a group and maximize a predefined objective function. The high utility pattern mining problem considered is different from former approaches, as it conducts rule discovery with respect to individual attributes as well as with respect to the overall criterion for the mined set, attempting to find groups of such patterns that together contributes to the most to a predefined objective function [4].

Y-C. Li, J-S. Yeh and C-C. Chang proposed an isolated item discarding strategy (IIDS). In this paper, they discovered high utility itemsets and also reduced the number of candidates in every database scan. They retrieved efficient high utility itemsets using the mining algorithm called FUM and DCG+. In this technique they showed a better performance than all the previous high utility pattern mining technique. However, their algorithms still suffer with the problem of level wise generation and test problem of apriori and it require multiple database scans [5].
Liu Jian-ping, Wang Ying, Yang Fan-ding et al proposed an algorithm called tree based incremental association rule mining algorithm (Pre-Fp). It is based on a FUFP (fast update frequent pattern) mining method. The major goal of FUFP is the re-use of previously mined frequent items while moving onto incremental mining. The advantage of FUFP is that it reduces the number of candidate set in the updating procedure. In FUFP, all links are bidirectional whereas in FP-tree, links are only unidirectional. The advantage of bidirectional is that it is easy to add, remove the child node without much reconstruction. The FUFP structure is used as an input to the pre-large tree which gives positive count difference whenever small data is added to original database. It deals with few changes in database in case of inserting new transaction. In this paper the algorithm classifies the items into three categories: frequent, infrequent and pre-large. Pre-large itemsets has two supports threshold value i.e. upper and lower threshold. The drawback of this approach is that it is time consuming [6].

Ahmed CF, Tanbeer SK, Jeong BS et al developed HUC-Prune. In the existing high utility pattern mining it generate a level wise candidate generation and test methodology to maintain the candidate pattern and they need several database scans which is directly dependent on the candidate length. To overcome this, they proposed a novel tree based candidate pruning technique called HUC-tree, (high utility candidate tree) which captures the important utility information of transaction database. HUC-Prune is entirely independent of high utility candidate pattern and it requires three database scans to calculate the result for utility pattern. The drawback of this approach is that it is very difficult to maintain the algorithm for larger database scan regions [7].

Shih-Sheng Chen et al (2011) proposed a method for frequent periodic pattern using multiple minimum supports. This is an efficient approach to find frequent pattern because it is based on multiple minimum threshold support based on real time event. All the items in transaction are arranged according to their minimum item support (MIS), and it does not hold download closure property, instead it uses sorted closure property based on ascending order. Then PFP (periodic frequent pattern) algorithm is applied which is same as that of FP-growth where conditional pattern base is used to discover frequent patterns. This algorithm is more efficient in terms of memory space, thereby reducing the number of database scans [8].

Chowdhury Farhan Ahmed, Syed KhairuzzamanTanbeer, Byeong-SooJeong, Young-Koo Lee, and Ho-Jin Choi et al proposed a Single-pass incremental and interactive mining for finding weighted frequent patterns. The existing weighted frequent pattern (WFP) mining cannot be applied for incremental and interactive WFP mining and also for stream data mining because they are based on a static database and its require multiple database scans. To overcome this, they proposed two novel tree structures IWFPPTWA (Incremental WFP tree based on weight ascending order) and IWFPPTFD (Incremental WFP tree based on descending order) and two new algorithms IWFPWA and IWFPFD for incremental and interactive mining using a single database scan. IWFPFD ensures that any non-candidate item cannot appear before candidate items in any branch of IWFPPTFD and thus speeds up the prefix tree. The drawback of this approach is that large memory space, time consuming and it is very difficult to support the algorithm for larger databases [9].

Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu proposed an efficient algorithm for mining high utility itemsets from transactional databases. In this paper, they discovered two algorithms named as UP-Growth and UP-Growth+ for mining high utility itemsets from transactional databases. In this technique they are totally dependent on the candidate length; it scans the database twice to construct the UP-Tree. They used efficient utility mining algorithm to generate huge number itemsets called potential high utility itemsets (PHUIs). In this technique they achieved a better performance than all previous high utility pattern mining techniques. However these algorithms still endure with the problem of search space, level wise candidate generation and wide memory usage [10].

IV. SUMMARY OF EXISTING SYSTEM

Utility pattern mining is based on a centralized database and the generation of frequent itemset is done locally. The frequent pattern mining is based on binary format of items in transactions. On the other hand incremental algorithm aims to discover the frequent pattern where the data is static. The drawback of high utility mining and incremental algorithm is that it imposes excessive communication overhead as the data is centralized and static. The static algorithms will not be effective when changes to original database occur continuously.

Proposed System:

The Proposed strategies can not only decrease the overestimated utilities of PHUIs but greatly reduce the number of candidates. Different types of both real and synthetic data sets are used in a series of experiments to the performance of the proposed algorithm with state-of-the-art utility mining algorithms. Experimental results show that UP-Growth and UP-Growth+ outperform other algorithms substantially in term of execution time, especially when databases contain lots of long transactions or low minimum utility thresholds are set.

Advantages:

1. Two algorithms, named Utility pattern growth(UP Growth)and UP-Growth+, and a compact tree structure, called utility pattern tree(UP-Tree),for discovering high utility item sets and maintaining important information related to utility patterns within databases are proposed.
2. High-Utility item sets can be generated from UP-Tree efficiently with only two scans of original databases. Several strategies are proposed for facilitating the mining process of UP-Growth+ by maintaining only essential information in UP-Tree.
3. By these Strategies, overestimated utilities of candidates can be well reduced by discarding utilities of the items that cannot be high utility or are not involved in search space.

Implementation

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

V. CONCLUSION

In this paper, a distributed and dynamic method is proposed to generate complete set of high utility itemsets from large databases. Mining high utility itemsets from databases refers to finding the itemsets with high profit. In distributed, it disposes the unpromising items based on the minimum utility itemsets from transactions database. This approach creates distributed environment with one master node and two slave nodes scans the database once and counts the occurrence of each item. The large database is distributed to all slave nodes. The global table has the final resultant. Incremental Mining Algorithm is used where continuous updating goes on appearing in a database. Finally incremental database is rearranged and the high utility itemsets is discovered. Hence, it provides faster execution, that is reduced time and cost.

REFERENCES


