A Survey on Efficient Incremental Algorithm for Mining High Utility Itemsets in Distributed and Dynamic Database

P. Asha¹, Dr. T. Jebarajan², G. Saranya³

¹Research Scholar, Computer Science and Engineering Department, Sathyabama University, Chennai.
²Head of the Department, Computer Science & Engineering, Rajalakshmi Engineering College, Chennai.
³M.E Student, Computer Science & Engineering, Sathyabama University, Chennai.

Abstract— Data Mining is the process of analyzing data from different perspectives and summarizing it into useful information. It can be defined as the activity that extracts information contained in very large database. That information can be used to increase the revenue or cut costs. Association Rule Mining (ARM) is finding out the frequent itemsets or patterns among the existing items from the given database. High Utility Pattern Mining has become the recent research with respect to data mining. The proposed work is to combine the High Utility Pattern Mining and Incremental Frequent Pattern Mining. The traditional method of mining frequent itemsets assumes that the data is centralized and static, which impose excessive communication overhead when the data is distributed, and they waste computational resources when the data is dynamic. To overcome this, Utility Pattern Mining Algorithm is proposed, in which itemsets are maintained in a tree based data structure, called as Utility Pattern Tree, and it generates the itemset without examining the entire database, and has minimal communication overhead when mining with respect to distributed and dynamic databases. A quick update incremental algorithm is used which scans only the incremental database as well as collects only the support count of newly generated frequent itemsets. Incremental Mining Algorithm not only includes new itemset into a tree but also remove the infrequent itemset from a utility pattern tree structure. Hence, it provides faster execution, that is reduced time and cost.

Keywords— Data Mining, Association Rule Mining, High Utility Mining, Incremental Mining.

I. INTRODUCTION

Data mining is the process of discovering useful knowledge from a collection of data. Association Rule Mining (ARM) is an important data mining technique which is used to discover the patterns/rules among items in a large database. The goal of ARM is to identify group of items that occur together, for example in a market basket analysis. A transaction database consists of two aspects such as internal and external utility. Quantity of a product in a particular transaction is called the internal utility and the profit value of a product is called external utility. The utility of itemset is defined as the product of external utility and internal utility.

The high utility itemset is that the itemset with a utility not less than a user specified minimum support threshold value; else that itemset is treated as a low utility itemset.

Incremental algorithm is a process where only the incremental part will be mined further for discovering frequent patterns. Main goal of incremental mining algorithm is that, the data is continuously added to the original transaction database, hence the database size becomes larger and mining the entire lot would take high computation time, hence it would be better if we mine only the updated portion. Hence it can provide faster execution when compared to the existing methods.

II. 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 in 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 contribute 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 a 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 Khairuzzaman Tanbeer, Byeong-So Jeong, Young-Koo Lee, 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 IWFPTWA (Incremental WFP tree based on weight ascending order) and IWFPTWD (Incremental WFP tree based on descending order) and two new algorithms IWFPTWA and IWFPTWD for incremental and interactive mining using a single database scan. IWFPTWD ensures that any non-candidate item cannot appear before candidate items in any branch of IWFPTWD 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, Bui-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].

A. 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.

III. PROPOSED SYSTEM

The proposed high utility incremental mining is a conceptual model built for distributed and dynamic database. Mining high utility itemsets from databases refers to finding the itemsets with high profit. The high utility itemset means that if its utility is no less than a user-specified minimum utility threshold, otherwise it is called a low-utility itemset.

Administrator can enter the Specified Minimum Utility Threshold value. The time taken to complete the entire computation is called the processing time and it is the difference between the starting time and ending time of a computation.

The System employs one master node and two slave nodes. Partition the database and send it to every slave node for computation. The slave node scans the database once and counts the occurrence of each item. These data’s are stored in their local table. The entire table from the slave nodes is transmitted to the master node. The Master Node prepares a global table. The global table has the number of occurrence of all data items in the given database. Based on the minimum utility threshold value it calculates the promising and unpromising itemsets. Promising items are those for which the threshold values are greater than the admin specified minimum utility. Unpromising Items are those items whose threshold values are lesser than the admin specified minimum utility and hence remove the infrequent items.

Incremental Mining Algorithm is used where continuous updating goes on appearing in a database. Incremental Mining Algorithm not only includes new itemset into a tree, but also disposes the infrequent itemset from Utility Pattern tree structure. Finally incremental database is rearranged and the high utility itemsets is discovered.

A. System Architecture

Fig. 1 shows the general block diagram of the proposed high utility incremental mining. First, the administrator creates the login page where the user name and password are given by the administrator and are validated with the database, once the validation is success then the login is provided.

In the daily market basis, each day a new product is released, so that the administrator would add the product, view the stock details, update the new product itemsets and the full control is only for the administrator. Customer can purchase the items and all the purchased items are stored in the transaction history and then transaction details are sends to original database which shows the list of items that are purchased by the customer. Then create an UP-tree.

UP-tree construction becomes an input for the incremental algorithm, and then the incremental database stores the copy of UP-tree constructed. Incremental algorithm requires only two scans.

![Fig. 1. Proposed Architecture Diagram](image-url)
In scan1 the incremental database alone is constructed and then it discovers the promising and unpromising itemsets as well as removes the unpromising items from the incremental database. In scan2, after the removal of unpromising items, the incremental database is rearranged and the high utility itemsets were discovered.

IV. 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


