Review Paper - High Utility Itemsets Mining on Incremental Transactions using UP-Growth and UP-Growth+ Algorithm

Miss. A. A. Bhosale[#1], S. V. Patil[#2], Miss. P. M. Tare[#3], Miss. P. S. Kadam[#4]
[#1]Computer Sci. and Engg. Department, Jaywant College of Engineering & Management, Killemachindra Gad.
[#2]Computer Sci. and Engg. Department, Annasaheb Dange College of Engineering, Ashta, Sangli.
[#3]Computer Sci. and Engg. Department, Jaywant College of Engineering & Management, Killemachindra Gad.
[#4]Computer Sci. and Engg. Department, Malajiraje Bhosale Technical Campus, Islampur.
E-mail ID: [#1]bhosale.anita11@gmail.com, [#2]siddheshwar.patil@gmail.com, [#3]pallavitare123@gmail.com, [#4]pallavikadam14@gmail.com
[#1#2#3#4]Shivaji University, Kolhapur, Maharashtra, India.

Abstract - One of the important research area in data mining is high utility pattern mining. Discovering itemsets with high utility like profit from database is known as high utility itemset mining. There are number of existing algorithms have been work on this issue. Some of them incurs problem of generating large number of candidate itemsets. This leads to degrade the performance of mining in case of execution time and space. In this paper we have focus on UP-Growth and UP-Growth+ algorithm which overcomes this limitation. This technique uses tree based data structure, UP-Tree for generating candidate itemsets with two scan of database. In this paper we extend the functionality of these algorithms on incremental database. .

***Keywords** – Data mining, High utility mining, Utility mining*

*****

I. INTRODUCTION

Frequent itemset mining means finding items that occurs in a database above a user given frequency threshold. These techniques do not consider the quantity or profit of the purchased items. Therefore it is not efficient for the user who want find the importance of the items in database. However quantity and profit are basic terms for maximizing the profit of the organization. For this purpose new technique is introduced called as high utility mining. This technique refers to finding itemsets from database which gives high utility. Utility means importance or interestedness of items. Utility of items is calculated by multiplying internal utility and external utility. Itemset in a single transaction is called internal utility and itemset in different transaction database is called external utility.

High utility itemset is itemset which have utility no less than a user-specified minimum utility threshold; otherwise, it is called a low-utility itemset. In many applications like cross-marketing in retail stores mining such high utility itemsets from databases is an important task.

Existing techniques [2, 3, 4, 5, 6, 7] used for utility pattern mining. However, the existing methods often generate a large set of potential high utility itemsets and the mining performance is degraded consequently. If database contain long transactions or low threshold value is set situation is more complicated for utility mining. The large number of potential high utility itemsets forms a challenging problem to the mining performance. Two existing algorithms deal with these issues. In this paper these algorithms will work on incremental transaction database.

II. RELATED WORK

Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer, Byeong-Soo Jeong, and Young-Koo Lee presented three novel tree structures for efficiently perform incremental and interactive HUP mining[2]. The first tree structure is used to arrange the items according to their lexicographic order. It is known as Incremental HUP Lexicographic Tree (IHUPL-Tree). It can capture the incremental data without any restructuring operation. The next tree structure is the IHUP Transaction Frequency Tree (IHUPTF-Tree), which arranging items according to their transaction frequency in descending order. To reduce the mining time, the last tree, IHUP-Transaction-Weighted Utilization Tree (IHUPTWU-Tree) is designed. Structure of this tree is based on the TWU value of items in descending order.

Alva Erwin, Raj P. Gopalan, and N. R. Achuthan, proposed CTU-PROL algorithm for efficient mining of high utility itemsets from large datasets[3]. This algorithm finds the large TWU items in the transaction database. If data sets is too large to be held in main memory, the algorithm creates subdivisions using parallel projections and for each subdivision, a *Compressed Utility Pattern Tree (CUP-Tree)* is used to mine the complete set of high utility itemsets. If the dataset is small, it creates a single *CUP-Tree* for mining high utility itemsets.

Shankar S., Purusothaman T., Jayanthi, S., suggested a novel algorithm for mining high utility itemsets[4]. This fast utility mining (FUM) algorithm finds all high utility itemsets within the given utility constraint threshold. The proposed FUM algorithm scales well as the size of the transaction database increases with regard to the number of distinct items available.

R. Chan, Q. Yang, and Y. Shen, suggested mining high utility itemsets[5]. They proposed a novel idea of top-K objective-directed data mining, which focuses the top-K high utility closed patterns. They add the concept of utility to capture highly desirable statistical patterns and present a level wise itemset mining algorithm. They develop a new pruning strategy based on utilities that allow pruning of low utility itemsets to be done by means of a weaker but antimonotonic condition.

Ramaraju C., Savarimuthu N., proposed a conditional tree based novel algorithm for high utility itemset mining[6]. A novel conditional high utility tree (CHUT) compress the transactional databases in two stages to reduce search space and a new algorithm called HU-Mine is proposed to mine complete set of high utility item sets.

Y. Liu, W. Liao, and A. Choudhary, proposed a fast high utility itemsets mining algorithm[7]. They are present a Two-Phase algorithm to efficiently prune down the number of candidates and can precisely obtain the complete set of high utility itemsets. In the first phase, they propose a model that applies the “transaction-weighted downward closure property” on the search space to expedite the identification of candidates. Latter phase identifies the high utility itemsets

Adinarayanareddy B., O. Srinivasa Rao, MHM Krishna Prasad, suggested improved UP-Growth high utility itemset mining[8]. The compact tree structure, Utility Pattern Tree i.e. UP-Tree, maintains the information of transactions and their itemsets. It facilitates the mining performance and avoid scanning original database frequently. UP-Tree scans database only twice to obtain candidate items and manage them in an efficient data structured way. UP-Growth takes more execution time for Second Phase by using UP-Tree. Hence they presents modified algorithm aiming to reduce the execution time by effectively identifying high utility itemsets.

P. Asha, Dr. T. Jebarajan, G. Saranya, presents a survey on efficient incremental algorithm for mining high utility itemsets in distributed and dynamic database[9]. The proposed system employs one master node and two slave nodes. Database is partitioned for every slave node for computation. The slave node counts the occurrence of each item. These data’s are stored in their local table. Then each slave node sends these tables to master node. The Master Node maintain global table for storing these data. Based on the minimum utility threshold value it calculates the promising and unpromising itemsets.

III. PROBLEM DEFINATION

We have studied some proposed algorithms in related work. But all these algorithms incur the problem of producing a large number of candidate itemsets. Such a large number of candidate itemsets degrades the mining performance in terms of execution time and space. If algorithm generates huge number of candidate itemsets, then higher processing time it consumes. Utility pattern growth (UP-Growth) and UP-Growth+ algorithm[1] overcomes this limitation. These algorithms mine high utility itemsets by using effective strategies. The information of high utility itemsets is maintained in a tree-based data structure named *utility pattern tree (UP-Tree)* such that candidate itemsets can be generated efficiently with only two scans of database.

IV. PROPOSED SYSTEM

UP-growth and UP-Growth+[1] algorithm find high utility Itemsets efficiently. By applying the proposed strategies of these algorithms (Like DGU, DGN, DLU, and DLN), the number of generated candidate itemsets can be highly reduced in phase I and high utility itemsets can be identified more efficiently in phase II. This technique used on static datasets. It did not consider the modification of database.

Our proposed system will work on incremental database i.e. deletion or insertion of one or more records from database will consider. To achieve this it uses the existing techniques[2]. Proposed system can avoid unnecessary or repetition of calculations by using previous results when a database is updated, or when the threshold value is changed.

System Architecture of proposed system as follows:

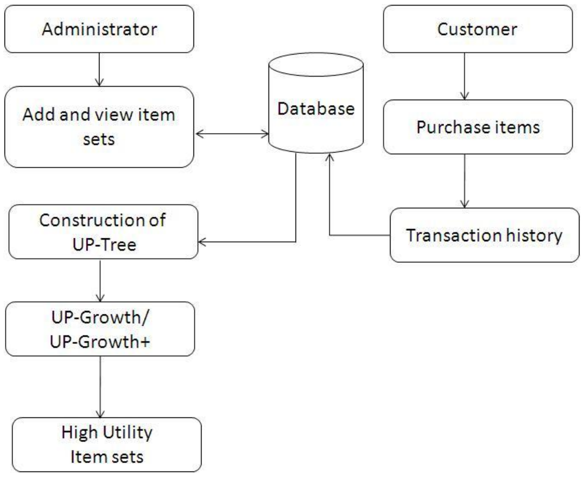

Fig. 1 System architecture

V. METHDOLOGY

Module 1: Administrator

The administrator maintain database of the transactions made by customers. In the daily market basis, each day a new product is released, so that the administrator would add the product, update the new product view the stock details.

Module 2: Customer

Customer can purchase the items. All the purchased items history are stored in the transaction database.

Module 3: Construction of UP-Tree[1]

1. First scan:-

- Initially Transaction Utiity(TU) of each transaction is computed. Then TWU of each single item is also accumulated.
- Discarding global unpromising items.
- Utilities of unpromising items are eliminated from the TU of the transaction.
- Then remaining promising items in the transaction are sorted according to the descending order of TWU.

2. Second scan:-

- UP-Tree is constructed by inserting transactions.

Module 4: UP-Growth Algorithm[1]

UP-Growth efficiently generates PHUIs from the global UP-Tree with two strategies, namely DLU (Discarding local unpromising items) and DLN (Decreasing local node utilities). For this Minimum Item Utility Table, abbreviated as MIUT, is used to maintain the minimum item utility for all global promising items.

In DLU(Discarding local unpromising items) strategy the minimum item utilities of unpromising items are discarded from path utilities of the paths during the construction of a local UP-Tree.

In DLN (Decreasing local node utilities) the minimum item utilities of descendant nodes for the node are decreased during the construction of a local UP-Tree. It is applied during the insertion of the reorganized paths.

Module 5: UP-Growth+ Algorithm[1]

Applying UP-Tree to the UP-Growth takes more execution time for Phase II. A modified algorithm i.e. UP-Growth+ reduce the execution time by effectively identifying high utility itemsets. It computes the Maximum transaction Weighted Utilization (MTWU) from all items and considering multiple of *min_sup* as a user specified threshold value.

Module 6: UP-growth and UP-growth+ for incremental Database

Proposed system will work, where continuous updating goes on appearing in a database. If the data is continuously added to the original transaction database, then the database size becomes larger and mining the entire lot would take high computation time, hence proposed system will mine only the updated portion of the database. It will use previous mining results to avoid unnecessary calculations.

VI. CONCLUSION

UP Growth and UP-growth+ algorithm are efficient for high utility itemsets mining. It also gives better performance on incremental database. This algorithm works better if one or more transactions are deleted or inserted in transaction database. It avoids unnecessary calculations by using previous mining results. This technique generates candidate itemsets with only two scans of the original database.

REFERENCE

[1] Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu, Fellow, IEEE "Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases" IEEE Trans. Knowledge and Data Engineering, vol. 25, no. 8, August 2013

[2] Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer, Byeong-Soo Jeong, and Young-Koo Lee, Member, IEEE "Efficient Tree Structures for High Utility Pattern Mining inIncremental Databases" IEEE Trans. Knowledge and Data Engineering, Vol. 21, Issue 12, pp. 1708-1721, December 2009.

[3] Alva Erwin, Raj P. Gopalan, and N. R. Achuthan, "Efficient Mining of High Utility Itemsets from Large Datasets", In Proc. of PAKDD 2008.

[4] Shankar, S.; Purusothaman, T.; Jayanthi, S. "Novel algorithm for mining high utility itemsets" International Conference on Computing, Communication and Networking, Dec. 2008.

[5] Raymond Chan; Qiang Yang; Yi-Dong Shen, "Mining high utility itemsets" In Proc. of Third IEEE Int'l Conf. on Data Mining ,November 2003.

[6] Ramaraju, C., Savarimuthu N. "A conditional tree based novel algorithm for high utility itemset mining", International Conference on Data mining, June 2011.

[7] Ying Liu, Wei-keng Liao, Alok Choudhary "A Fast High Utility Itemsets Mining Algorithm" In Proc. of the Utility-Based Data Mining Workshop, 2005.

[8] Adinarayanareddy B ,O Srinivasa Rao, MHM Krishna Prasad, "An Improved UP-GrowthHigh Utility Itemset Mining" International Journal of Computer Applications (0975-8887) Volume 58-No.2, November 2012.

[9] P. Asha, Dr. T. Jebarajan, G. Saranya, "A Survey on Efficient Incremental Algorithm for Mining High Utility Itemsets in Distributed and Dynamic Database" IJETAE Journal, Vol.4, Issue 1, January 2014.