International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-1, Issue-4, July 2015
ISSN: 2395-3470
www.ijseas.com

High Utility Itemsets Mining from Incremental Databases

Mr. Dewarde D.H.
[1]M.E. Second Year Student, Department of Comp. Engineering, SPCOE, Otur, Pune, India

Mr. Kahate S.A.
[2]Professor, Department of Computer Engineering, SPCOE, Otur, Pune, India

Abstract*— Data Mining defined as an activity that extracts some non-trivial information which included in large database. Mainly, data mining techniques have focused on detecting the stastical correlation between the items that are more frequent in transaction databases. One of the important research areas in data mining is high utility pattern mining from transactional database. Discovering itemsets with high utility like profitable items from database is known as high utility itemset mining. There are many number of existing algorithms have been work on this issue. But some of them incurs problem of generating large number of candidate itemsets. This affects 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 will overcome this limitation. This technique uses tree based data structure finding itemsets, UP-Tree for generating candidate itemsets with two scan of database. In this paper we extend the functionality of UP-Growth and UP-Growth+ algorithms on transactional database. In High utility itemsets mining the objective is to identify itemsets that have utility value above a given utility threshold to generate tree. However, two main problems determined in relevant studies: 1) the utilities (e.g., importance of item or profits of item) are not considered. Actual utilities of patterns cannot be reflected in frequent itemsets. 2) Existing utility mining methods outcomes too many patterns and this makes it difficult for the users to filter useful patterns among the huge set of patterns. In view of this propose a novel framework, named GUIDE (Generation of maximal high Utility Itemsets from Data strEams), to find maximum high utility itemsets from data streams with different no of models, i.e. landmark, sliding window and time fading models. The advanced structure, named MUI-Tree (Maximal high Utility Itemset Tree), it requisites essential information for the mining processes and the proposed strategies further facilitates the performance of GUIDE. Main contributions of this paper are as follows: 1) To the best of our knowledge, this is the first work on mining the compact form of high utility itemsets from data streams; 2) GUIDE is an progressive one-pass framework which meets the requirements of data stream mining; 3) GUIDE produces novel patterns which are not only high utility but also maximal, which offers compact and insightful hidden information in the data streams. Experimental results which show that our approach outperforms the state-of-the-art algorithms under various conditions in data stream environments on different models.

***Keywords*— Data mining; High utility mining; Utility mining; maximal pattern; data stream mining**

I. INTRODUCTION

Frequent itemset mining means finding items that presents in a database above a user given frequency threshold value. These techniques do not consider the quantity of items or profit of the purchased items. Accordingly it is not efficient for the end-user who want find the importance of the items in database. After all quantity of items and profit of items are basic terms for maximizing the profit of the organization. For this purpose new technique in data mining is introduced called as high utility mining. This technique is useful for finding itemsets from database which gives high utility. Utility means influence or usefulness of items. Utility of items is mainly calculated by multiplying internal utility and external utility. Itemset in a single transaction is called utility or internal utility and itemset in different transaction database is called external utility.It allows users to identify the usefulness or importance of items using different values. Thus, it emulates the impact of different items. High utility itemsets mining is useful in decision making activity of many applications, such as retail marketing and Web service, since items are actually different in many ways in real applications. High utility itemset is itemset which having utility no less than a user-specified minimum utility threshold value; otherwise, it is called a low-utility itemset. In many applications such as cross-marketing in retail stores mining such high utility itemsets from databases is an important process. Existing techniques which [2, 3, 4, 5, 6, 7] used for utility pattern mining in large database. However, the existing methods usually generate a large set of potential high utility itemsets and the mining performance is degraded consequently. If database accommodate long

transactions or low threshold value is set situation is more complicated for utility mining. The large number of potential high utility itemsets designs a challenging problem to the mining act. Two existing algorithms deal with these issues to solving some kind of problem as well as a compact data structure for efficiently discovering high utility itemsets from transactional databases.

In this paper these algorithms will work on transactional database. Augmentation of this work is summarized as follows:

1.Two algorithms, titled as utility pattern growth (UP-Growth) and (UP-Growth)+, and a compact tree Structure, called utility pattern tree (UP-Tree), for Discovering high utility itemsets and maintaining Necessary information related to utility patterns Within databases are proposed. High-utility itemsets Generated from UP-Tree efficiently with only two scans of original databases.
2. Several approach are proposed for expedite the Mining processes of UP-Growth and UP-Growth+ by maintaining only essential information in UP-Tree. By these approaches, overestimated utilities of candidates can be well reduced by discarding utilities of the items that cannot be high utility or are not included in the search space. Not only proposed strategies can decrease the overestimated utilities of PHUIs but also greatly reduce the number of candidates.
3. Different types of both real and synthetic data sets are worn in a series of experiments to compare the performance of the proposed algorithms with the State-of-the-art utility mining algorithms. Experimental results show that UP-Growth and UP Growth+[1].outperform other algorithms substantially in terms of execution time, especially when databases contain lots of long transactions or low minimum utility threshold.

In view of these, we investigate the topic of finding *maximal high utility itemsets*, which are not only high utility but also maximal items, from data streams. A novel framework is known as GUIDE (*Generation of maximal high Utility Itemsets from Data strEams*) is produced for finding maximal high utility itemsets from data streams. Based on the proposed framework, three algorithms, namely GUIDE$_{LM}$, GUIDE$_{SW}$ and GUIDE$_{TF}$, are produced for landmark, sliding window and time fading models, respectively. The basic idea of these algorithms is to effectively pick up the essential information, i.e., the utilities of appeared itemsets, and store them into tree structures, namely *MUI-Trees* (***M**aximal high **U**tility **I**temset Trees*). To facilitate the mining process, two strategies are produced for efficient tracing and pruning the MUI-Trees.

MUI-Trees maintain essential information for the mining processes and the two proposed strategies further facilitate the performance of GUIDE. It generates the unique patterns which are not only high utility but also maximal itemsets. The patterns provide compact and insightful hidden information in datasets and data in the data streams. Through experimental evaluation of Data Mining, the proposed algorithms are shown to substantially outperform the state-of-the-art algorithms for mining high utility itemsets from data streams .

II. LITERATURE SURVEY

In the literature survey we will go to discuss various existing methods which allow user to access the services from multiple service providers in High Utility Itemsets Mining. Below we are discussing some of them.

Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer, Byeong-Soo Jeong, and Young-Koo Lee granted three novel tree structures for efficiently perform transactional and interactive HUP mining [2]. The first tree structure is used to organize the items according to their lexicographic order. It admitted as Transactional HUP Lexicographic Tree (IHUPLTree). It captures the transactional data without any restructuring operation. The next tree structure is the IHUP Transaction Frequency Tree (IHUPTF-Tree), which is useful arranging items according to their transaction frequency in descending order. To curtail the mining time, the last tree, IHUP-Transaction-Weighted Utilization Tree (IHUPTWUTree) is designed. The structure of this tree is based on the Transactional Weighted Utility(TWU) value of items in descending order.

Alva Erwin, Raj P. Gopalan, and N. R. Achuthan, Advised CTU-PROL algorithm for efficient mining of high utility itemsets from large datasets[3]. These algorithms search the large TWU items in the transaction database. If data sets is too large to be held in main memory, the algorithm generates 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 Limited, it built a single CUP-Tree for mining high utility itemsets.

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

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

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

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

Adinarayanareddy B., O. Srinivasa Rao, MHM Krishna Prasad, implied improved UP-Growth high utility itemset mining[8]. The compact tree structure, Utility Pattern Tree i.e. UP-Tree, maintains the history of transactions and their itemsets. It expedites the mining performance and avert scanning original database frequently. UP-Tree scans database only two times to achieve candidate items and manage them in an efficient data structured way. UP-Growth gates more execution time for Second Phase by prating UP-Tree. Hence they proposed modified algorithm aiming to reduce the execution time by effectively identifying high utility itemsets.

P. Asha, Dr. T. Jebarajan, G. Saranya, implied a survey on efficient transactional algorithm for mining high utility itemsets in distributed and dynamic database [9]. The proposed system applies one master node and two slave nodes. Database is subdivided for every slave node for computation. The slave node measures the existence of each item. These datas are gathered in their local table. Then each slave node forwards these tables to master node. The Master Node maintain global table for gathering these data. Based on the minimum utility threshold value it measures the promising and unpromising itemsets.

Ahmed et al. [10] implied a structure named IHUP-Tree for maintaining essential information about utility mining. It avoids scanning of database for multiple times and generating candidates or patterns during the mining process. However, although IHUP-Tree produces better performance than Two-Phase and IIDS, it still provides too many HTWUIs.

Tseng et al. proposed a novel algorithm named UP-Growth [11], which applies several pruning and counting strategies during the data mining processes. By the proposed strategies, the estimated utilities are effectively decreased in UP-Trees during the data mining processes and the number of HTWUIs is further reduced. Therefore, the system performance of utility mining can be improved significantly.

III. EXISTING SYSTEM

We have prepared some proposed algorithms in related work. But all these algorithms obtain the problem of generating a large number of candidate itemsets. Such a large number of candidate itemsets downgrades the mining performance in terms of execution time and space. If algorithm produces huge number of candidate itemsets, then higher processing time it consumes. Utility pattern growth (UP-Growth) and UP-Growth+ algorithm[1] conquer this limitation. These algorithms fount high utility itemsets by using adequate strategies. The information of high utility itemsets is managed in a tree-based data structure named utility pattern tree (UP-Tree) such that candidate itemsets can be produced efficiently with only two scans of database.

IV. PRELIMINARY AND DEFINITION

In this section, The definitions of utility mining from the transaction databases to the data stream. Firstly define the notations for utility mining in streaming in data environments and then address the problem statement of this work.

A *data stream DS* is composed of a continuous set of transactions denoted by $\{Tid_1, Tid2... Tid_n\}$. A *transaction* Tid_k is denoted as $\{itemset_k, t_k\}$, where $itemset_k$ $(k = 1)$ is the items appeared in the transaction Tid_k and t_k $(k = 1)$ is the time when $itemset_k$ appeared in *DS*. $Itemset_k$ is composed of a set of items and their purchased numbers, which is denoted by $\{(x_1, q_1), (x_2, q_2), ..., (x_p, q_p)\}$, where x_r (1 □

r (p, x_r I) is the *item* and q_r (1 r p) is the *purchased number* of x_r in Tid_k. $I = \{i_1, i_2, ..., i_m\}$ is the set of items. An *itemset* is a subset of I. Every item in the data stream has its own unit profit.

Definition 1 (Valid transaction.): For a transaction $Tid_k = \{itemset_k, t_k\}$, Tid_k is *valid* in the data stream DS if it is *captured* by following conditions:

1. For landmark and time fading models, assume the landmark time point is set as t_{lm}.

   Tid_k is valid if t_{now} $t_k > t_{lm}$.

2. For sliding window model, assume window size is set as t_{sw}. Tid_k is valid if t_{now}

   $t_k > t_{now} - t_{sw}$, where t_{now} is the current time of DS.

Definition 2 (Utilities of the elements in a data stream.) : The utilities of items, itemsets and transactions in the data stream DS are defined as follows.

1. The utility of an item x_r in a transaction Tid_k is denoted as $u(x_r, Tid_k)$ and defined as

   $p(x_r)$ q_r.

2. The utility of an itemset X in Tid_k is the summarization of the utilities of the items those are belonged to X in Tid_k. It is denoted and defined as $u(X, Tid_k)$

   $\sum_{x \in X \wedge Tid_k} u(x, Tid)$.

3. The utility of X in DS, which is denoted as $u(X)$, is defined as

   $\sum_{Tid \in X \wedge Tid \in DS_k \wedge k} u(X, Tid)$

4. The transaction utility of Tid_k is denoted and defined as $u(Tid_k)$

   $\sum_{x \in Tid_k \wedge k} u(x, Tid)$.

5. Total utility of DS is the summation of the utility of all valid transactions in DS. Assume the set of valid transactions in DS is VT. It is denoted and defined as

   $TotalU = \sum_{Tid \in VT_k \wedge k} u(Tid)$. By Definition 1, for landmark and time fading models, it is accumulated progressively since the landmark was set; for sliding window model, only the utilities of the transactions in items in the window are accumulated.

V. PROPOSED WORK

UP-growth and UP-Growth+[1] algorithm discovering high utility itemsets efficiently. By applying the proposed strategies of these algorithms the number of generated candidate itemsets can be highly decreased in phase I and high utility itemsets can be identified more efficiently in phase II. This technique is useful on static datasets. It did not consider the adaption of database. Our proposed system will perform on transactional database which is related to data mining. i.e. deletion or insertion of one or more records from database will consider on database history. To achieve this it uses the existing approach[2]. Proposed system can avert unnecessary or repetition of calculations by using previous results when a database is updated, or when the threshold value is replaced.

1. UP-Growth Algorithm: The UP-Growth is one of the productive algorithms to generate high utility itemsets depending on construction of a global UPTree. The framework of UP-Tree includes three steps:

(i). UP-Tree Construction according to items.
(ii). From UP-Tree generate PHUIs .
(iii). Using PHUI identify high utility itemsets.

The construction of global UP-Tree is follows, (i). Discarding global unpromising items in transactional database (i.e., DGU strategy) is mainly focus to eliminate the low utility items and their utilities from the transaction utilities. (ii). Discarding global node utilities in transactional database (i.e.DGN strategy) during construction in global UP-Tree. In DGN strategy, node utilities which are nearest to UP-Tree root node are effectively reduced items. The PHUI is equal to TWU, which compute all itemsets utility with the help of estimated utility. Finally, analyze high utility itemsets in transactional database (not less than min-

sup)from PHUIs values. The global UP-Tree which contains many sub paths. Each path studied from bottom node of header table. This path is known as conditional pattern base (CPB).

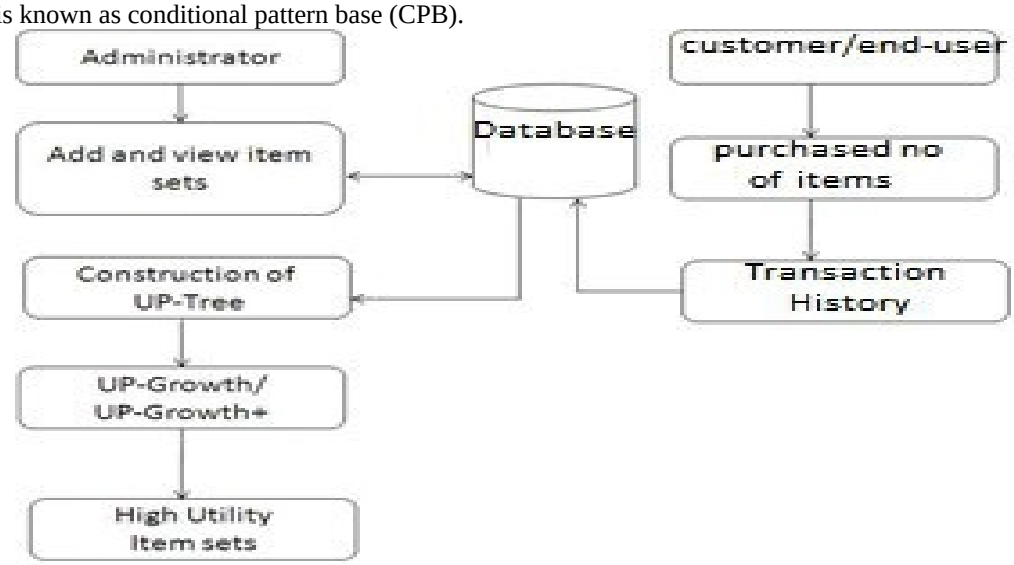

Fig. 1. System Architecture

2. Improved UP-Growth Algorithm

Although DGU and DGN strategies are very efficiently reduce the number of candidates in Phase 1(i.e., global UP-Tree). But they cannot be applied during the construction of the local UPTree in transactional database(Phase-2). Rather use, DLU strategy is (Discarding local unpromising items) to discarding utilities of low utility items from path utilities of the paths and DLN strategy (Discarding local node utilities) to discarding item utilities of descendant nodes during the local UP-Tree construction. Even though,the algorithm facing some performance issues in phase-2. To overcome this issues, maximum transaction weight utilizations (MTWU) are computed from all the items and considering multiple of min-sup value of items as a user specified threshold value as shown in algorithm. By this modification, performance of system will increase compare with existing UP-Tree construction also increases the performance of UP-growth algorithm. An improved utility pattern growth is compressed as similar to IUPG.

VI. PROPOSED WORK

High Utility Itemsets mining from transactional database constructed in 4 modules.
Module 1: Administrator
The administrator preserve database of the transactions made by customers. In the daily market basis, each day a new product is let go, so that the administrator would add the product or items, and update the new product view the stock details.
Module 2: Customer
Customer can purchase the number of items. All the purchased items history is stored in the transaction database.
Module 3: Construction of UP-Tree [1]
1. First scan:-
    a) Initially Transaction Utility (TU) of each transaction is counted. Then TWU of each single item is also assembled.
    b) Removing global unpromising items.
    c) Utilities of unpromising items are excluded from the TU of the transaction.
    d) Then remaining promising items in the transaction are arranged according to the descending order of TWU.
2. Second scan:-
    a) UP-Tree is generated by inserting transactions.
Module 4: UP-Growth Algorithm [1]

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

In DLN (Decreasing local node utilities) the minimum item utilities of descendant nodes for the node are decreased during the constitution of a local UP-Tree. It is tested during the insertion of the reorganized paths. In DLU (Discarding local unpromising items) strategy the minimum item utilities of unpromising items are removed from path utilities of the paths during the construction of a local UPTree.

Module 5: UP-Growth+ Algorithm [1]
Applying UP-Tree to the UP-Growth takes further execution time for Phase II. A modified algorithm i.e. UP-Growth+ curtail the execution time by effectively identifying high utility itemsets. It measures 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 transactional Database
Proposed system will work, where continuous updating goes on emerging in a database. If the data is continuously inserted to the original transaction database, then the database size becomes increase 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 earlier mining results to avoid unnecessary calculations.

CONCLUSION

In this paper, frequent itemset mining is stationed on the rationale that the itemsets which appear more frequently in the transaction databases are of more importance to the user. However the practical benefits of mining the frequent itemset by considering only the frequency of appearance of the itemsets is imposed in many application domains such as retail research. It has been that in many real applications that the itemsets that devote the most in terms of some user defined utility function (for e.g. profit) are not necessarily frequent itemsets.

The UP Growth and UP-growth+ algorithm are more efficient for high utility itemsets mining. It gives better performance on transactional database to find out high utility itemsets. This algorithm works well if one or more transactions are deleted or inserted in transaction database. It abstains unnecessary calculations by using previous mining results. This technique provokes candidate itemsets with only two scans of the original

REFERENCES

[1] Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu, “Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases,” IEEE Trans. Knowledge and Data Engineering, vol. 25, no. 8, August 2013
[2] Adinarayanareddy B ,O. Srinivasa Rao, MHM Krishna Prasad, “An Improved UP-Growth High Utility Itemset Mining,” International Journal of Computer Applications ,November 2012.
[3] Ramaraju, C.; Savarimuthu, N; “A conditional tree based novel algorithm for high utility itemset mining,” International Conference on Data mining, June 2011.
[4] 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. IEEE Trans. Knowledge and Data Engineering, Vol. 21, Issue 12, pp. 1708-1721, December 2009.
[5] Shankar, S.; Purusothaman, T.; Jayanthi, S.; “Novel algorithm for mining high utility itemsets,” International Conference on Computing, Communication and Networking, Dec. 2008
[6] A. Erwin, R. P. Gopalan, and N. R. Achuthan, “Efficient Mining of High Utility Itemsets from Large Datasets,” In Proc. of PAKDD 2008.
[7] Y. Liu, W. Liao, and A. Choudhary, “A Fast High Utility Itemsets Mining Algorithm,” In Proc. of the Utility-Based Data Mining Workshop, 2005.
[8] R. Chan, Q. Yang, and Y. Shen , “Mining high utility itemsets,” In Proc. of Third IEEE Int’l Conf. on Data Mining ,November 2003.