Mining High Utility Itemsets from Large Transactions using Efficient Tree Structure

T.Vinothini
Department of Computer Science and Engineering,
Knowledge Institute of Technology, Salem.

V.V.Ramya Shree
Department of Computer Science and Engineering,
Kathir College of Engineering, Covai.

Abstract- Mining high utility itemsets from a transactional database refers to the discovery of itemsets with high utility like profits. It is an extension of the frequent pattern mining. Although a number of relevant algorithms have been proposed in recent years, they incur the 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. The situation may become worse when the database contains lots of long transactions or long high utility itemsets. Association rule mining with item set frequencies are used to extract item set relationships. Frequent pattern mining algorithms are designed to find commonly occurring sets in databases. Memory and run time requirements are very high in frequent pattern mining algorithms. Systolic tree structure is a reconfigurable architecture used for utility pattern mining operations. High throughput and faster execution are the highlights of the systolic tree based reconfigurable architecture. The systolic tree mechanism is used in the frequent pattern extraction process for the transaction database. Systolic tree based rule mining scheme is enhanced for Weighted Association Rule Mining (WARM) process, used to fetch the frequently accessed transaction database with its weight values. The dynamic dataset is weight assignment scheme uses the item request count and span time values. The proposed system improves the weight estimation process with span time, request count and access sequence details.

Keywords - utility mining, pattern growth approach, high utility itemsets, systolic tree structure

I.INTRODUCTION

Identifying the frequent itemset is the main goal of Association Rule Mining (ARM).In order to obtain the utility of an itemset, decision-oriented (utility-based) ARM should be used. Utility is the measure of, how useful or interesting an itemset is. The utility can be measured in terms of profit, quantity, cost and other user preferences which are not considered in frequent pattern mining. In frequent pattern mining, only the frequency of occurrence of the itemset is considered, which may not yield the most profitable itemset. For finding the frequent itemset, only the binary occurrence (0/1) of the item is considered (i.e., either the item is present/not) [7] whereas, for the utility of an item, the considerations are the quantity and profit of each item in the itemset of transaction in a transactional database.

"Downward closure property" (Anti-monotone property) is the major challenge of utility pattern mining when compared to frequent pattern mining. According to this property, any subset of frequent itemset is also frequent which may not applicable to Utility mining. In utility mining, the subset of high utility itemset need not be of high utility.

II.RELATED WORK

1) LEVEL WISE CANDIDATE GENERATION AND TEST APPROACHES

Two-Phase

In this algorithm, Transaction-weighted utility (TWU), which is the sum of utilities of all transactions containing an item X (TWU(X)), is considered to prune the search space. Here transaction-weighted downward closure property is maintained. This property states that all subset of *high TWU* must also be high in TWU. [9] This shows that the *downward closure property* is maintained**.**

Phase I: In the first pass, it finds the entire one-element Transaction Weighted Utility Itemset (TWUI).With that it generates all the candidates for two-element TWUI. In the second pass, it finds all the two-element TWUI. The three-element TWUI is generated based on that and this continues till it finds all the candidates.

Phase II: In this phase, an extra database scan is required to filter out the low utility itemsets.Even though, this captures the complete set of high utility itemset, it is suitable only for sparse datasets with short patterns of transaction and not for large, growing databases and multiple-passes are needed in the phase-I. Rescanning of the whole database is needed when a new transaction arrives. Thus the execution time increases with each new transaction. These were considered as the disadvantages of the Two Phase algorithm.

UMining and UMining_H

A pruning strategy, known as UMining is based on utility upper bound property. UMining_H is a pruning strategy based on a heuristic method [15]. In UMining, set of all items are found from the Transactional database, then actual utility of each itemset is calculated. From that, select all high utility itemsets and generate all possible candidate itemsets and then remove the itemsets which have the utility upper bound less than the user-specified threshold

Heuristic pruning strategy is obtained by combining the Upper bound property with the Apriori property. In heuristic method, the size of the itemset is increased and the support is decreased due to fewer transactions. Some high utility itemset may be erroneously pruned here [14]. Moreover, the downward closure property is not satisfied in these methods and overestimation of too many patterns [3]. Thus, excessive candidate generations and poor test methodology and incomplete collection of high utility itemsets due to erroneous pruning [2] are considered as the disadvantages.

THUI-Mine

THUI-Mine is the first algorithm for mining high utility itemsets from data streams. This is based on Two-Phase algorithm which is extended with sliding-window-filtering technique.These mining steps are divided into two functions. Preprocessing function and Incremental function.The preprocessing step needs two scans of the original database. The database is partitioned into 'n' segments. Each partition is scanned sequentially for the generation of the candidate itemset. Consider the user-specified utility threshold as 'x'. This gets divided into 'n' (i.e.,x/n) for each partition of the data base. This partial threshold is called Filtering Threshold. The itemsets whose threshold is below the filtering threshold are dropped out.The incremental step deals with the update of the data stream. After the update activities, the old transactions are removed and the new transactions are added to the database. One extra scanning is required in this step to find all the temporal high utility itemsets[11].Even though it uses the data base reduction method to find the high utility itemset, it suffers from the performance problem such as false candidate itemsets and memory consumption

MHUI-BIT and MHUI-TID

MHUI-BIT (Mining High-Utility Itemsets based on BITvector)and MHUI-TID(Mining High-Utility Itemsets based on TIDlist) algorithms have been proposed for sliding window-based HUP mining over data streams .In this sliding window model , the data within a fixed time or fixed number of transactions by keeping a window which slides with time. The data kept in the window is used, if the mining process is requested [10].

A three-phase method, used by the algorithms MHUI-TID and MHUI-BIT is to mine high utility itemsets from data streams. These algorithms have found to outperform the existing data stream algorithm such as THUI-Mine. The calculation time and the number of candidate-patterns have been reduced by the bit-vector, TID-list. When the number of items of the transactions get increased in a window the lists become inefficient. These algorithms can capture only length-1 and length-2 candidate patterns by using a tree structure. Explicit level-wise candidate generation method is used for the pattern length greater than two [3].Hence this level-wise search and multiple database scans are consider as the disadvantages of these algorithms.

IIDS

The isolated items discarding strategy (IIDS) was proposed to reduce some candidates in every pass of databases. Here, Critical Function (CF) is considered in variant of the Transaction Weighted Downward Closure Property.CF calculation is an important step which is designing by IIDS. The CF of the candidates that are generated is to identify which candidates are useless and thereby reduce the number of candidates by pruning the isolated items (search space). Any level-wise utility mining methods can be combined with this IIDS for efficiency. Thus, two algorithms, FUM and DCG+ are evolved in such a manner. That is, DCG+ is a fast utility mining algorithm that is obtained by combining DCG and IIDS. However, the disadvantage is that it suffers from the problem of level-wise candidate generation-and-test methodology and need several database scans [9].

2) PATTERN GROWTH APPROACHES

IHUP

For mining high utility mining from an incremental database, this method had been implemented with three new structures with the “build once and mine many” property.HUP$_L$-Tree with lexicographical arrangement of items along with IHUP$_{TF}$-Tree and IHUPTWU-Tree structures are the data structures. To denote all the three tree structures, the term IHUP-Trees is used. The adjustments in the tree structure are required only for the last deleted/added/modified group of transactions.Every node in IHUP-Tree consists of an item name, a support count, and a TWU value. The framework of the algorithm consists of three steps: 1.IHUP-Tree construction. 2. HTWUIs generation. 3. high utility itemsets identification.

In step 1, the items in the rearranged transactions are inserted into the IHUP-Tree in TWU descending order (or any lexicographical order). In step 2, HTWUIs are generated by using the Frequent Pattern Growth algorithm. In step 3, by single scan of the original database, High Utility itemsets and their utilities are generated [4].Even though HTWUIs are produced without generating candidate itemset, it produces too many HTWUIs in Phase-I. This algorithm and Two-Phase produces same number of HTWUIs. The mining performance of Phase-I, in terms of memory consumption, gets degraded with such a large number of HTWUIs. This also affects the Phase-II by taking more execution time in Phase-II for finding high utility item set from those HTWUIs. Thus, this is the disadvantage of this method [13]

CTU-PRO

The algorithm CTU-PRO uses the data structure compressed utility pattern (CUP) tree which is an extension of the CFP-Tree. To prune the search space in CTU-PRO, the concept of TWU is used and also rescanning is avoided while finding the actual utility of high TWU itemset. While identifying each individual high TWU, a *GlobalCUP-Tree* is created by the algorithm. *LocalCUP-Tree* (a small projection tree), which is extended from the GlobalCUP-Tree, is used for mining all high utility patterns. A *pointer link*, which connects the same item identifier, is used by the CUP-Tree. This pointer link feature enables *bottom-up traversal* of the tree while mining [6]. The Global CUP-Tree is implemented entirely in the main memory. So this technique is not adequate for very large databases. Also, it cannot complete mine in a reasonable time, as it takes much time for the tree construction [15].

UMMI

Most of the above Pattern mining algorithms commonly have two steps. Potential itemsets are found in the first step and then the high utility itemsets are obtained from those potential itemsets. If the potential itemsets are larger in number, there occurs the bottleneck in mining. To avoid this problem, the number of potential itemset has to be reduced. The main difficulty is to speed up the first phase. Also, an efficient data structure is to be used to speed up in the second phase.UMMI (Utility Mining using Maximal Itemset property) uses the maximal itemset in order to reduce the number of potential itemset. All the HTWU itemsets are contained in the MTWU itemsets and the number of MTWUsets are less than the HTWU itemsets. Here, a *top-down traversal strategy* is implemented to minimize the traversal cost. So, only MTWU itemset is used in the *Maximal Phase* to reduce the search space. The data structure, *HTP* (High TWU Pattern) is used to capture all transaction utility itemset, from which MTWU itemsets are mined. Superset and subset checking optimizations are performed during this mining in Maximal phase.The second phase, known as Utility phase, implements the data structure MLexTree(Maximal Lexicographic Tree) to get the true high utility itemset after discovering all the MTWU itemsets. To generate all the HTWU itemsets, each MTWU itemset is inserted into the MLexTree and each node represents one HTWU itemset. After the tree construction, the database is scanned to store the utility values. The tree is traversed to get the high utility itemsets. Thus, this algorithm implements two-tree structure and two phases to mine the high utility itemsets. Although, the HTWU itemsets and MTWU itemsets are equal in number, the performance of MLexTree is faster in verifying the high utility itemsets. A fixed amount of memory is used in this algorithm with linear scaling with respect to number of transactions[15].

UP-Growth

Several pruning and counting strategies are implemented during the mining processes in UP-Growth. UP-Tree is the data structure used to maintain the information about transaction and high utility itemsets. This tree can be constructed in two scans of the database. In the first scan, promising items are found by using some strategies and in the second scan transactions are inserted into the database.The tree construction can be done in two phases[12]. In Phase-I, Global tree construction by DGU (Discarding global unpromising items) strategy and DGN (Discarding global node utilities) strategy. In phase-II, Local UP-Tree is constructed by using another two strategies such as DLU (Discarding local unpromising items) strategy and DLN (Discarding local node utilities) strategy. Some performance issues are found in the Phase- II. To overcome that, maximum transaction weighted utilizations (MTWU) are computed [13].The framework steps of this algorithm, is same as that of the IHUP, except the second step. Instead of HTWUI generation, PHUI (Potential High Utility Itemsets) are generated, as this approach does not follow the traditional framework (Transaction weighted utilization mining model). In this method the set of PHUIs are smaller than HTWUI set in phase-I [1]. Comparing to IHUP, UP-Growth is more efficient, as the promising patterns which cannot be pruned in IHUP are get reduced. Though the four strategies effectively reduced the PHUIs, the drawbacks of this algorithm are I/O cost and more execution time for Phase-II [1].

III. PROBLEM DEFINITION

For utility pattern mining, there are many extensive calculations, by considering the external utility and internal utility of the itemset. For instance, if the transaction database in a retail market is considered, the internal utility can be the *quantity* of each item which can be extracted from the same database (internal) and the external utility can be the *profit* of each item which can be extracted from the (external) Profit database. The product of the internal and external utility yields the utility of an itemset. An itemset is said to be high utility itemset, if it satisfies the user-specified minimum threshold; otherwise, it is a low utility itemset. As multiple database scans are required, for these calculations, the utility pattern mining is considered as tedious and challenging one.

IV. PROPOSED WORK

In systolic tree, the nodes are the processing elements. These set of processing nodes with regular and local connections which takes external inputs and process them in a predetermined manner. The architecture thus produced are not general and tied to a specific algorithm. A systolic tree algorithm is several times faster than the implementation of the FP-growth algorithm and UP-growth algorithm to implement in large dataset. We propose a system, designed to perform weighted rule mining for transaction dataset. Automatic weight estimation scheme is used in the system. Each item is assigned a weight value with reference to the request count and sequence. A systolic tree is an arrangement of pipelined processing elements (PE) in a multidimensional tree pattern. The role of the systolic tree as mapped in FPGA hardware is then similar to the FP-tree as used in software. The transaction items are updated into the systolic tree with candidate item matching and count update operations.

Systolic tree is used to arrange candidate sets with frequency values. Due to the limited size of the systolic tree, a transactional database must be projected into smaller ones each of which can be mined in efficiently. A high performance projection algorithm which fully utilizes the advantage of FP-growth is used. It reduces the mining time by partitioning the tree into dense and spare parts. Systolic tree based rule mining scheme is enhanced for weighted rule mining process.

LOCAL TRANSACTION UTILITY

Data should be preprocessed more easily and effectively by the user. The database is in the form of transaction table by removing all erroneous data. Inconsistencies and redundancies are to be removed. Utility of an item is obtained by the product of the local and external utility. Each transaction is considered individually. The quantity of the data items in each transaction is read. Thus considering the quantity is the local utility mining.

EXTERNAL UTILITY

To get the external utility, get the unit profit for each item from the Profit database. Considering profit for all itemset in the database is the external utility. The transaction utility is the product of the local and the external utilities. The itemsets should satisfy the user-specified minimum threshold to be the utility itemset.

QUANTITY MEASUREMENT OF UTILITY

The operation of quantity measurement is shown in Figure 5.6. Transaction Utility for each transaction is considered. From that, Transaction Weighted Utility of each item is considered. Threshold is set by the user and the items that are not satisfying the user-threshold are considered as unpromising items. Discard the item's transaction utilities from the database. New transaction utility is the Reorganized TU.

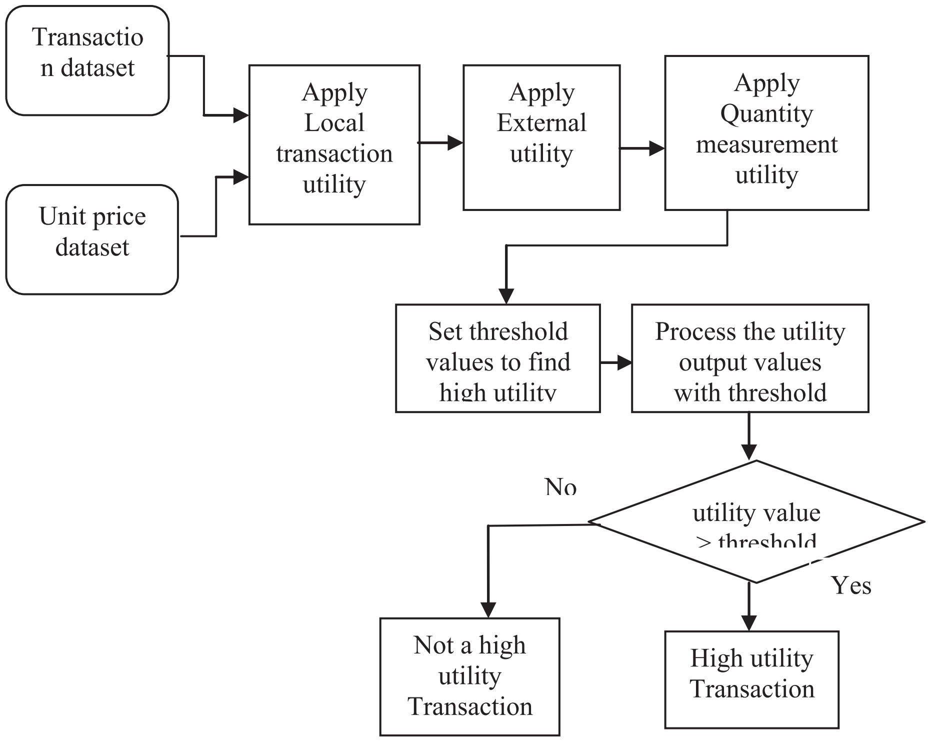
Transactio n dataset
Unit price dataset
Apply Local transaction utility
Apply External utility
Apply Quantity measurement utility
Set threshold values to find high utility
Process the utility output values with threshold
No
utility value > threshold
Yes
Not a high utility Transaction
High utility Transaction

Figure 1. System Architecture

HIGH UTILITY ITEMSETS

The Reorganized Transaction Unit is considered for the tree construction. The RTU are given as the input to the systolic tree. After processing, the utility value is compared with the user-specified threshold. If the value met the threshold, then it is high utility transaction.

V. CONCLUSION

The algorithm named UP-Growth for mining high utility itemsets from transaction databases has been used. A data structure named UP-Tree is used for maintaining the information of high utility itemsets. PHUIs can be efficiently generated from UP-Tree with only two database scans. Moreover, there are several strategies to decrease the overestimated utility and enhance the performance of utility mining. In the experiments, both real and synthetic data sets were used to perform a thorough performance evaluation. On the other hand, memory usage and the time consumption for processing nodes increase with the increase in the database size. So, an algorithm with data structure named Systolic tree structure can be used to implement parallelism in processing nodes. Systolic tree structure is a reconfigurable architecture used for utility pattern mining operations. High throughput and faster execution are the highlights of the systolic tree based reconfigurable architecture. It reduces the mining time by partitioning the tree into dense and spare parts. Systolic tree based rule mining scheme is enhanced for weighted rule mining process.

REFERENCES

[1] Adinarayanareddy.B.O, Srinivasa Rao,Krish na Prasad.M.H.M "An Improved UP-Growth High Utility Itemset Mining" International Journal of Computer pplications (0975 – 8887) Volume 58– No.2, November 2012.

[2] Ahmed. C.F, Tanbeer.S.K, Jeong.B.-S, and Choi.H.-J, "A framework for mining interesting high utility patterns with a strong frequency affinity," Information Sciences 181 (2011) 4878–4894.

[3] Ahmed. C.F, Tanbeer.S.K, Jeong.B.-S, and Choi.H.-J, " Interactive mining of high utility patterns over data streams" Expert Systems with Applications 39 (2012) 11979–11991.

[4] Ahmed. C.F, Tanbeer.S.K, Jeong.B.-S, and Lee.Y.-K, "Efficient Tree Structures for High Utility Pattern Mining in Incremental Databases," IEEE Trans. Knowledge and Data Eng., vol. 21, no. 12,pp. 1708-1721, Dec. 2009.

[5] Ahmed C.F, Tanbeer SK, Jeong B-S, Lee Y-K "HUC-Prune: an efficient candidate pruning technique to mine high utility patterns" Appl Intell 34(2):181–198(2011).

[6] Erwin. A, Gopalan. R.P, Achuthan. N.R, "A bottom-up projection based algorithm for mining high utility itemsets", in: Proceedings of 2nd International Workshop on Integration Artificial Intelligence and Data Mining, 2007, pp. 3–11.

[7] Guangzhu Yu, Shihuang Shao and Xianhui Zeng **, "**Mining Long High Utility Itemsets in Transaction Databases," Wseas Transactions on Information Science & Applications., Issue 2, Volume 5, Feb. 2008.

[8] Li. H.F, Huang. H.Y, Chen.Y.C, Liu.Y.J, and Lee.S.Y., "Fast and Memory Efficient Mining of High Utility Itemsets in Data Streams*,"*Proc. IEEE Eighth Int'l Conf. on Data Mining,pp. 881-886, 2008.

[9] Li. Y.-C, Yeh. J.-S, and Chang.C.-C, "Isolated Items Discarding Strategy for Discovering High Utility Itemsets,"Data and Knowledge Eng.,vol. 64, no. 1, pp. 198-217, Jan. 2008.

[10] Shie B.-E., Tseng.V.S., and Yu.P.S, "Online Mining of Temporal Maximal Utility Itemsets from Data Streams," Proc. 25th Ann.ACM Symp. Applied Computing,Mar. 2010.

[11] Tseng.V. S, Chu. C. J and Liang.T, "Efficient Mining of Temporal High Utility Itemsets from Data streams," in Proc. of ACM KDD Workshop on Utility-Based Data Mining Workshop (UBDM'06), pp. 18-27, USA, August 2006.

[12] Tseng.V.S, Shie.B.E, Wu.C. W and Yu. P. S, "Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases*,"* IEEE Trans. Knowledge and Data Eng., VOL. 25, NO. 8, August 2013.

[13] Tseng.V.S, Wu. C. W, Shie.B.E and Yu. P. S "UP-Growth: An Efficient Algorithm for High Utility Itemset Mining." In Proc. of ACM-KDD, Washington, DC, USA, pp. 253-262, July 25–28, 2010.

[14] Yao. H, Hamilton. H.J, "Mining itemset utilities from transaction databases," Data and Knowledge Engineering 59 (2006) 603–626.