Heuristics Rules for Mining High Utility Item Sets From Transactional Database

S. Manikandan[1], Mr. D. P. Devan[2]

[1, 2] *(PG scholar, Assistant Professor, Department of Computer Science and Engineering, Indra Ganesan College of Engineering, Trichy)*

Abstract: *Mining frequent item sets is an active area in data mining that aims at searching interesting relationships between items in databases. It can be used to address to a wide variety of problems such as discovering association rules, sequential patterns, correlations etc. A transactional database is a data set of transactions, each composed of a set of items, called an item sets (frequently occurring in a database). Existing methods often generate a huge set of potential high utility item sets and their mining performance is degraded consequently. There is a lacking of mining performance with these huge number of potential high utility item sets; higher processing time too. The novel algorithms as well as a compact data structure for efficiently discovering high utility item sets are proposed. High utility item sets is maintained in a tree-based data structure named UP-Tree (Utility Pattern Tree). Experimental results predict that not only reduce the number of candidates effectively and also performed well when databases contain lots of long transaction or a low minimum utility threshold is used.*

Keywords: *High utility item sets, Transaction Weight Utilization, Utility Mining, Frequent item set, Data mining Navigation.*

I. Introduction

Progress in digital data acquisition, distribution, and recovery and storage technology has resulted in the growth of huge databases. One of the most challenges facing organizations and individuals is how to turn their rapidly expanding data collections into available and actionable knowledge. The try to counter these challenges gathered researchers from areas such as data, machine learning, databases and probably several more, resulting in a new area of research, called Data Mining. Data mining is typically mentioned in the broader setting of Knowledge discovery in databases, or KDD, and is vision as a single step in a larger process called the KDD process this process includes: Developing an understanding of the application domain, the appropriate prior knowledge, and the goals of the end-user. Choose the target data set on which discovery is to be performed, and attack and transform this data if necessary.

The data mining step is concerned with the task of automated information extraction from data that might be valuable to the owner of the data store. A working definition of this discipline is the following. Data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner. In order to do this analysis, several different types of tasks have been identified, corresponding to the objectives of what needs to be analyzed and more importantly, what the intended outcome should describe. These tasks can be categorized as follows .Exploratory Data Analysis The goal is here to explore the data without any clear ideas of what is wanted to be found. Typical techniques include graphical present methods, projection performance and summarization methods.

Retrieval by Content the user has a specific pattern in mind in advance and is looking for similar patterns in the data set. This task is most commonly used for the retrieval of information from large collections of text or image data. The main challenge here is to define similarity and how to find all similar patterns according to this definition. Pattern Discovery aim is to find local patterns that occur frequently inside a database. A set of algorithms have been studied for several types of patterns, such as sets, tree structures graph structures or arbitrary relational structures and association rules.

Moreover when concerning on the expansion of the World Wide Web has resulted in a large amount of data that is now in general freely available for user access. The unusual types of data have to be handled and organized in such a way that they can be accessed by different users professionally. Therefore, the purpose of data mining techniques on the Web is now the focus of an increasing number of researchers. A number of data mining ways are used to discover the hidden information in the Web.

The focus of this concept is to provide an overview how to use frequent pattern mining techniques for discovering different types of patterns in a Web log. Web mining involves a wide range of applications that aims at discovering and extracting hidden information in data stored on the Web. Another important purpose of Web mining is to provide a mechanism to make the data access more efficiently and sufficiently. The third attractive approach is to discover the information which can be derived from the activities of users, which are stored in log files for example for predictive Web caching. Thus, Web mining can be categorized into three different classes based on which part of the Web is to be mined; these three categories are (i) Web content mining, (ii) Web structure mining and (iii) Web usage mining.

The paper is structured as follows: Section II briefly describes some related work. Section III describes the system overview of the proposed system. Section IV describes an efficient utility pattern growth algorithm. Experimental results of the proposed algorithm is shown in Section V. Section VI concludes the paper and hints some extensions of the work.

II. Related Work

Association rule mining is considered to be an interesting research area and studied widely by many researchers. In the recent years, some relevant methods have been proposed for mining high utility item sets from transaction databases.

[1] Reported a tree-based algorithm, named IHUP. A tree-based structure termed IHUP-Tree is used to maintain the information about item sets and their utilities. Each node of an IHUP-Tree consists of an item name, a TWU value and a support count. This approach frequently generates a huge set of potential high utility item sets and their mining performance is degraded consequently.

This situation may become worse when databases contain many long transactions or low thresholds are set.[2] Reported the several emerging applications data mining and discovering hidden frequent patterns in time series databases, e.g., sensor networks, environment monitoring and inventory stock monitoring .This concept address the problem of discovering frequent patterns in databases with multiple time series and propose an incremental technique for discovering the complete set of frequent patterns .The demerits are pruning search space for high utility item sets mining is difficult because a superset of a low utility item set may be a high utility item set. Little Time Spending Process

.[3] Suggested the high utility pattern (HUP) mining over data streams has become a challenging research issue in data mining. It saves a huge amount of memory space by keeping the recent information very efficiently in an HUS-tree. The Demerits are consisting multiple database scans.[4] implemented the knowledge discovery process; utility based measures can be used in three ways, which call the roles of the utility based measures.

First, measures can be used to prune uninteresting patterns during the data mining process to narrow the search space and thus improve the mining efficiency. Secondly, measures can be used to rank the patterns according to the order of their interestingness scores. Thirdly, measures can be used during post processing to select the interesting patterns. This approach meets the critical requirements on time and space efficiency for mining data streams.

[5] Reported about the increasing profit of a corporation is one of the most important goals of data mining. Traditional association rules methods only consider whether an item is bought in a transaction. However, customers can buy more than one of the same item in a transaction, and the unit profit for each item may vary.

Utility mining, a generalized form of share mining has been proposed to overcome the drawback of traditional association rule mining.[6] Suggested the expression-based axis, that gives more weight to gene expression profiles than the other two interpretation axis, is the most currently used. However, approaches in this axis present many well-known drawbacks.[7] Implemented the three variations of tree structure. Among them, the IHUPL Tree is very simple and easy to construct and handle, as it does not require any restructuring operation in spite of incremental updating of the databases

.A pattern growth approach is used to avoid the level-wise candidate generation-and-test methodology. The demerits is that not scalable for large number of transactions.[8] reported the weighted-support; a new measure of item sets in databases is introduced with only binary attributes. The basic idea behind w-support is that a frequent item set may not be as important as it appears, because the weights of transactions are different. These weights are completely derived from the internal structure of the database based on the assumption that good transactions consist of good items.[9] Suggested a new framework, namely Mobile Commerce Explorer (MCE), to mine and predict mobile users movements and transactions under the context of mobile commerce .Its limitations are adapting only Mobile commerce patterns. Its limitations are adapting only Mobile commerce patterns [10] implemented the derived algorithm named MF to convert the original sequence of log data into a set of maximal forward references. The backward references are being filtered out and concentrated on mining meaningful user access sequences. Tree construction time is little costlier.

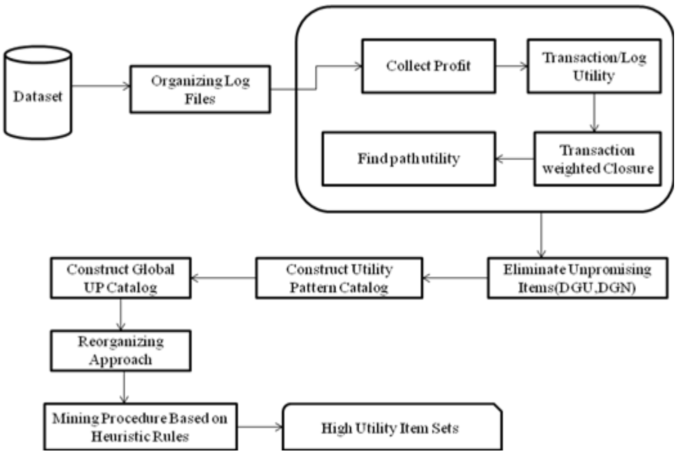
Dataset
Organizing Log Files
Collect Profit
Transaction/Log Utility
Transaction weighted Closure
Find path utility
Eliminate Unpromising Items(DGU,DGN)
Construct Utility Pattern Catalog
Construct Global UP Catalog
Reorganizing Approach
Mining Procedure Based on Heuristic Rules
High Utility Item Sets

Fig. 1 Work Flow model of the Proposed System

III. SYSTEM OVERVIEW

The various existing techniques and some of their problems are discussed in Section II. This section discusses about the system overview and work flow model of the proposed system. The fig.1shows the system architecture of the proposed method. The dataset contains group of items that are retrieved from the transactional databases. The Log Files are collected from the data catalogs. Organizing log files are used to preprocess the item sets, which are used to avoid the repeated items in the dataset. Based on the log files the patterns are generated as per the logic such as each user is initialized with their own id. Form the log files the quantity which determines the number of times user accessed the items. For each log, the profit table is initialized. Transaction Utility is calculated based on multiplication of both the quantity and profit value of each item. Based on the transaction utility, transaction weighted closure is calculated. After that minimum support count is calculated based on maximum transaction weighted utility and user specified threshold value.

Maximum transaction weighted utility is calculated from the transaction utility. Eliminating unpromising items from the item set based on DGU and DGN strategy. After eliminating unpromising items from the item set the utility pattern catalog is constructed. It is to maintain the information of transactions and high utility item sets. Based on the utility pattern catalog, global UP catalog is constructed. The conditional patterns are generated by tracing the paths in the original Catalog. Then Apply DLU to reduce path utilities of the paths. The path utility of an item is estimated.

The conditional Catalogs (also called local Catalogs).The reorganized path construction process is done too. This process is done after discarding the nodes. Now apply the mining procedure based on the Heuristics rules. The final step is to identify high utility item sets from the large data set.

IV. PROPOSED TECHNIQUE

The work flow model of the proposed system is explained in Section III. This section describes the details about the proposed techniques with modules and algorithm.

A. Organizing The Log Files

Suppose The Log Files are collected from the data catalogs. The patterns are generated as per the logic such as each user is initialized with their own id. The quantities which determine the number of times user accessed the item set. For each log, the profit table is initialized. However the transaction utility (TU) hereby called as log utility (LU) is estimated by multiplying the quantity and log Profit value. Given a finite set of items I = {i1, i2… im}, each item ip has a unit profit pr (ip). An item sets X is a set of k distinct items {i1, i2… ik}, 1<=j<=k. k is the length of X. An item sets with length k is called a k-item set, 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. Utility of an item ip in a transaction Td is denoted as u(ip, Td) and defined as profit and quantity. Utility of an item sets X in Td is denoted as u(X, Td) and defined as adding all utilities . An item sets is called a high utility item set sif its utility is no less than a user-specified minimum utility threshold which is denoted as min_util. Otherwise, it is called a low utility item set. After the dataset is extracted the data has to be reframed in this manner, it should contain the (transaction, quantity) and thus computing the utility patterns .The patterns are generated as per the logic such as each user is initialized with their own id. The quantities which determine the number of times user accessed the item set. For each log, the profit table is initialized. However the transaction utility (TU) hereby called as log utility (LU) is estimated by multiplying the quantity and log Profit value

B. Transaction-Weighted Downward Closure

Let Transaction-weighted utility of an item sets X is the sum of the transaction utilities of all the transactions containing X, which is denoted as TWU(X) and defined as by summing up the transaction utilities of all transactions containing that item. The minimized threshold is set and the transaction weighted utility is found. An item sets X is called a high transaction weighted utility item set if TWU(X) is no less than min_util. Following this the reorganization of the table is carried out. It is depended on ranking the high utility items.

For instance consider a transaction Log {A, B, D, E, F} The TWF is carried out for each item resulting in the manner such as Say for Item A if TWF is 93, for B if it is 92, C is 99, D is 96, E= 107, F=50, G=45, If the items minimum utility is set as 50 then the F, G are eliminated. It is ordered in descending order, {E, C, D, A, B}.Hence the original transaction is again re ordered as {E, C, A, B}, its respective Transaction utility is found by multiplying the quantity and Profit value. These steps are repeated for each transaction. An item ip is called a promising item if TWU (ip), min_util. Otherwise it is called an unpromising item. Without loss of generality, an item is also called a promising item if its overestimated utility (which is different from TWU in this paper) is no less than min_util. Otherwise it is called an unpromising item.

These are eliminated. New TU after pruning unpromising items is called reorganized transaction utility (abbreviated as RTU). RTU of a reorganized transaction Tr is denoted as RTU (Tr). By reorganizing the transactions, not only less information is needed to be recorded in UP-Tree, but also smaller overestimated utilities for item sets are generated. Compute the Transaction utility of a transaction Td based on minimum weighted utility. Compute the Transaction-weighted utility of an item set X is the sum of the transaction utilities of all the transactions containing X, which is denoted as TWU(X).Estimate the high transaction weighted utility item set . It is the one which is not less than min_util. Evaluate the Transaction Weighted Downward Closure by downward closure property which can be done by applying the transaction weighted utility.

C. Utility Pattern Tree

The construction of a global UP-Tree can be performed with two scans of the original database. In the first scan, TU of each transaction is computed. At the same time, TWU of each single item is also accumulated. By TWDC property, an item and its supersets are unpromising to be high utility item sets if its TWU is less than the minimum utility threshold. Such an item is called an unpromising item.

In a UP-Tree, each node N consists of N.name, N.count, N.nu, N.parent, N.hlink and a set of child nodes. N.name is the node's item name. N.count is the node's support count. N.nu is the node's node utility, i.e., overestimated utility of the node. N.parent records the parent node of N. N.hlink is a node link which points to a node whose item name is the same as N.name. A table named header table is employed to facilitate the traversal of UP-Tree. In header table, each entry records an item name, an overestimated utility, and a link. The link points to the last occurrence of the node which has the same item as the entry in the UP-Tree. By following the links in header table and the nodes in UP-Tree, the nodes having the same name can be traversed efficiently.

Hence the unpromising items are found and eliminated. {F}, {G} and {H} have been removed by DGU. After a transaction has been reorganized, it is inserted into the global UP-Tree. When T1' = {(C, 10) (D, 1) (A, 1)} is inserted, the first node NC is created with NC.item = {C} and NC.count = 1. NC.nu is increased by RTU (T1') minus the utilities of the rest items that are behind {C} in T1', that is, NC.nu = RTU (T1') – (u ({D}, T1') + u ({A}, T1')) = 17–(2+5) = 10. Note that it can also be calculated as the sum of utilities of the items that are before item {D} in T1', i.e., NC.nu = u ({C}, T1') = 10. The second node ND is created with ND.item = {D}, ND.count = 1 and ND.nu = RTU (T1') – u ({A}, T1') = 17–5 = 12. The third node NA is created with NA.item = {A}, NA.count = 1 and NA.nu = RTU (T1') = 17. After inserting all reorganized transactions by the same way, the global UP-Tree is constructed. Comparing with the IHUP-Tree, node utilities of the nodes in UP-Tree are less than those in IHUP-Tree since the node utilities are effectively decreased by the two strategies DGU and DGN.

D. Utility Pattern-Growth

Generate conditional pattern bases by tracing the paths in the original tree, (2) construct conditional trees (also called local trees in this paper) by the information in conditional pattern bases and (3) mine patterns from the conditional trees. However, strategies DGU and DGN cannot be applied into conditional UP-Trees since actual utilities of items in different transactions are not maintained in a global UP Tree. It cannot know actual utilities of unpromising items that need to be discarded in conditional pattern bases unless an additional database scan is performed. To overcome this problem, a naïve solution is to maintain items' actual utilities in each transaction into each node of global UP-Tree. In view of this module propose two strategies, named DLU and DLN, that are applied in the first two mining steps and introduced in this and next subsections, respectively.

For the two strategies, maintain a minimum item utility table to keep minimum item utilities for all global promising items in the database.

In UP-Growth, minimum item utility table is used to reduce the overestimated utilities. In UP-Growth+, minimal node utilities in each path are used to make the estimated pruning values closer to real utility values of the pruned items in database. Assume that Nx is the node which records the item x in the path p in a UP-Tree and Nx is composed of the items x from the set of transactions TIDSET (TX). The minimal node utility of x in p is denoted as min(x, p) and defined as Minimal node utility for each node can be acquired during the construction of a global UP-Tree.

The conditional pattern is generated by tracing the paths in the original Catalog. The Minimum item utility is evaluated by minimum utility threshold. Find the local promising items in the catalog. Then Apply DLU to reduce path utilities of the paths. The path utility of an item is estimated. The conditional Catalogs (also called local Catalogs are constructed).The reorganized path construction process is done too. This process is done after discarding the nodes.

E. Final Strategy

After introducing the modification of global UP-Tree, now address the processes and two improved strategies of UP-Growth+, named DNU and DNN. When a local UP-Tree is being constructed, minimal node utilities can also be acquired by the same steps of global UP-Tree. In the mining process, when a path is retrieved, minimal node utility of each node in the path is also retrieved. It can simply replace minimum item utility.

DNU: Discarding local unpromising items and their estimated Node Utilities from the paths and path utilities of conditional pattern bases.DNN: Decreasing local Node utilities for the nodes of local UP-Tree by estimated utilities of descendant Nodes during the construction of global UPTree. However the effective frequent logs are estimated at the end.

F. Proposed Algorithm

The algorithm for the proposed technique as follows:

```
Input: Transaction database D, user specified threshold.
Output: high utility item sets.
  1. Scan database of transactions Td ϵ  D.
  2. Determine transaction utility of Td in D and TWU of item set (X).
  3. Compute min_sup (Maximum Transaction Weight Utilization ×user specified Threshold).
  4. If (TWU(X) ≤ min_sup) then Remove Items from transaction database.
  5. Else insert into header table H and to keep the items in the descending order.
  6. Repeat step 4 & 5 until end of the D.
  7. Insert Td into global UP-Tree.
  8. Apply DGU and DGN strategies on global UP- tree.
  9. Re-construct the UP-Tree.
  10. For each item ai in H do.
  11. Generate a PHUI from utility global tree.
  12. Identify high utility item sets using PHUI.
```

V. EXPERIMENTAL RESULTS

In this section, experimental results on datasets are summarized on enhanced UP-Growth algorithm. These experiments were conducted on 2.53 Intel(R) Core(TM) i3 Processor with 2 GB of RAM, and running on Windows XB operating system. All algorithms were implemented in java language (JDK1.5) and applied on real datasets to evaluate the performance algorithms based on the datasets T10I6D10K. Where T is the average size of transactions; I is the average size of maximal potential frequent item sets; D is the total number of transactions and N is the number of distinct items. Fig-2 shows the execution times on various min_sup values from 60% to 90%. From the graph the proposed algorithm is performed well when the databases contain lots of long transactions or a low minimum utility threshold is used.

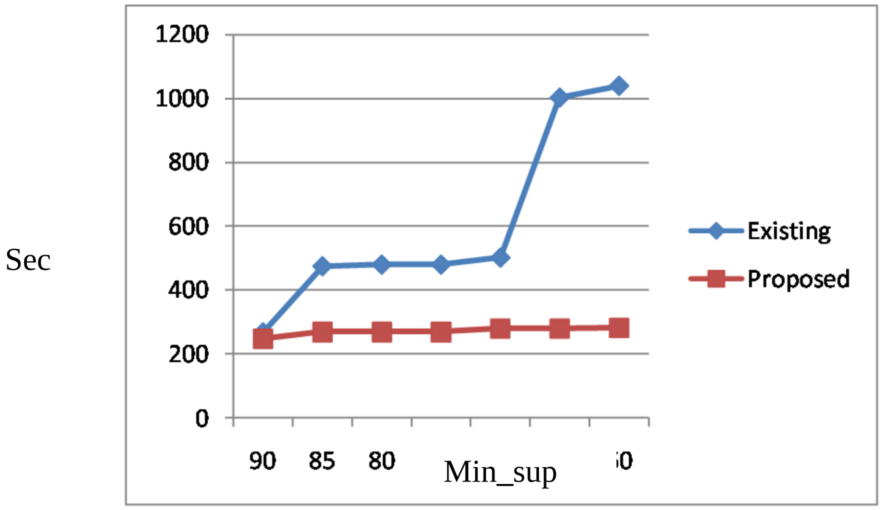

Fig. 2 Execution Time

VI. CONCLUSION

A data structure named utility pattern tree is proposed for upholding the information of high utility item sets. Potential high utility item sets are efficiently retrieved from utility pattern tree with only two database scans. Discarding Global Unpromising items and Discarding Global Node strategies are applied to decrease the overestimated utility and enhance the performance of utility mining. Therefore the proposed algorithm is efficient for mining high utility item sets from transaction databases. A data structure named utility pattern tree was proposed for maintaining the information of high utility item sets. Potential high utility item sets can be efficiently generated from utility pattern tree with only two database scans. The proposed algorithm is performed well when the databases contain lots of long transactions or a low minimum utility threshold is used. In future this approach can be efficient only when low quality threshold is applied. Hence this has to be revoked in Future. Moreover still there some scans performed and quite reorganization of tree requires processing time to some extent. Hence all these constraints will be considered in future.

REFERENCES

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

[2] Mohamed Y.Eltabakh, Mourad Ouzzani and Mohamed A.Khali "Incremental Mining for Frequent Patterns in Evolving Time Series Data bases" Technical Report of Purdue University, CSD TR#08-02, 2008.

[3] Syed Khairuzzaman Tanbeer, Chowdhury Farhan Ahmed and Byeong-Soo Seong "Efficient Mining of High Utility Patterns over Data Streams with a Sliding Window Method" in the Proceeding of PAKDD 2008, LNAI 5012.

[4] Yao,Howard J.Hamilton and Liqiang "A Unified Framework for Utility Based Measures for Mining Itemset", University of Regina Regina, SK, Canada S4S OA2.

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

[6] R. Martinez, N. Pasquier, and C. Pasquier, "GenMiner: Mining nonredundant Association Rules from Integrated Gene Expression Data and Annotations," Bioinformatics, vol. 24, pp. 2643-2644, 2008.

[7] Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer, Byeong-Soo Jeong, and Young-Koo Lee "Efficient Tree Structures for High Utility Pattern Mining in Incremental Databases", IEEE Transactions on Knowledge and Data Engineering, Vol. 21, No. 12, December 2009.

[8] Ke Sun and Fengshan Bai "Mining Weighted Association Rules without Preassigned Weights", IEEE Transactions on Knowledge and Data Engineering, Vol. 20, No. 4, December 2009.

[9] Eric Hsueh-Chan Lu, Wang-Chien Lee and Vincent S. Tseng "A Framework for Personal Mobile Commerce Pattern Mining and Prediction", IEEE Transactions on Knowledge and Data Engineering, Vol. 24, No. 5, May 2012.

[10] Ming-Syan Chen, Jung Soo Park and Philli S.Yu.Fellow "Efficient Data Mining for Path Traversal Patterns" IEEE Transactions on Knowledge and Data Engineering, Vol. 10, No. 2, March/April 1998.