Establising Downclosure Property on Mining High Utility Frequent Itemsets From Large Databases

K.Raghavi, K.Anita Davamani, M.Krishnamurthy

Abstract— — In today's world a vast amount of knowledge is stored in the web and database. Due to the availability of huge knowledge repositories, getting the relevant information is a challenging task and hence it must be mined and extracted. Association Rule Mining is one approach for extracting useful knowledge from database which includes frequent patterns and association rules between the items or attributes of a dataset with varying levels of strength. An efficient algorithm has been proposed to effectively prune the search space and efficiently capture all high utility itemsets with no miss. The challenge of utility mining is in restricting the size of the candidate itemset and simplifying the computation for calculating the utility. The existing up-growth algorithm reduces the no of candidate's item set but having complex strategies and not holding the downward closure property in up-tree. Here we use new algorithm to maintain downward closure property which is used in every FP-Tree and this algorithm removes complex strategies as well as reduces no of candidate itemset effectively. The new algorithm efficiently finds the high utility items. The main focus of this work is that a new Association Rule Mining has been proposed to enhance the capabilities of the existing Association Rule Mining algorithms in terms of the number of scans and memory utilization.

Index Terms—Downward closure Property, Frequent itemset, high utility mining, utility mining.

1 Introduction

Data Mining (the analysis step of the "Knowledge Discovery in Databases" process, or KDD) is an analytic process designed to explore large amounts of data in search of consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data.

Data Mining tasks can be classified into two categories, Descriptive Mining and Predictive Mining. The Descriptive Mining techniques such as Clustering, Association Rule Discovery, Sequential Pattern Discovery, is used to find human-interpretable patterns that describe the data. The Predictive Mining techniques like Classification, Regression, Deviation Detection, use some variables to predict unknown or future values of other variables.

An emerging topic in the field of data mining is High Utility Mining. Frequent pattern mining does not consider the relative importance of each item and Weighted ARM does not quantify an item.

Utility of items in a transaction database consists of two aspects :(1) the importance of distinct items, which is called external utility, and (2) the importance of items in transactions, which is called internal utility. Utility of an itemset is defined as the product of its external utility and its internal utility. An itemset is a high utility itemset if its utility is no less than a user-specified minimum utility threshold.

Association rule mining is to find out association rules that satisfy the predefined minimum support and confidence from a given database. Mining association rules consists of two steps [1]:

1) Finding all the frequent itemsets having adequate support.

2) Producing association rules from these frequent itemsets.

The outcome depends upon the results of step1. The first algorithm proposed for association rule mining is apriori and it involves large number of candidate generation and multiple scans of the database. With the development of the information technology and application of database the collected data far exceed people's ability to analyze it. Thus new and efficient methods are needed to discover knowledge from large databases and data mining appeared. In the algorithms of the association rules mining, apriori is the ancestor.

The main idea involved in Frequent Pattern growth is finding frequent itemsets by FP Tree construction and removing the subsets that are not frequent, which based on the classical Apriori. Existing systems for mining high utility itemsets are applied over estimated methods to facilitate the performance of utility mining. In these methods potential high utility itemsets are found first and then an additional database scan is performed for identifying their utilities. It often generates a huge set of potential high utility itemsets which degrades the performance. To address the above issue, we propose UP-Growth algorithm as well as a compact data structure UP-Tree for efficiently discovering the high utility itemsets from transactional databases.

2 Related Work

Two main pruning strategies used in itemset mining based on the Apriori property for frequent itemset mining [2]. It states that if an itemset is frequent by support, then all its nonempty subsets must also be frequent by support. FP Growth which allows frequent itemsets discovery without candidate itemset generation by building FP tree and mines frequent patterns by traversal through FP tree[3].To efficiently generate HTWUIs in phase I and avoid scanning database many times. proposed a compact data structure tree-based algorithm, called Incremental High Utility Pattern [IHUP], for mining itemsets with high utility. They use an IHUP-Tree to maintain the information of high utility itemsets and transactions. Each and Every node in this tree having an item name, a support count, and a TWU value. The framework of the algorithm con-

sists of three steps: (i) The construction of IHUP-Tree, (ii) the generation of HTWUIs and (iii) identification of itemsets with high utility[4].FUP is based on the framework of Apriori and is designed to discover the new frequent itemsets iteratively. The idea is to store the counts of all the frequent itemsets found in a previous mining operation. Using these stored counts and examining the newly added transactions, the overall count of these candidate itemsets are then obtained by scanning the original database[5].Mining based on utilities where the usefulness of an itemset is characterized as a utility constraint and two new pruning strategies have been introduced. UMining, to find all itemsets with utility values higher than threshold from a database. We also develop a heuristic algorithm, called UMining_H, which typically finds most itemsets with utility values higher than min- utility based on a heuristic pruning strategy[6].Two-Phase algorithm [19] for finding high utility itemsets. In the first phase, a model that applies the "transaction-weighted downward closure property" on the search space to expedite the identification of candidates. In the second phase, one extra database scan is performed to identify the high utility itemsets[7]. A novel algorithm Fast Utility Mining (FUM) in [8]. The authors also suggest a novel method of generating different types of itemsets such as High Utility and High Frequency itemsets (HUHF), High Utility and Low Frequency itemsets (HULF), Low Utility and High Frequency itemsets (LUHF) and Low Utility and Low Frequency itemsets (LULF) using a combination of FUM and Fast Utility Frequent mining (FUFM) algorithms.

3 EXISTING WORK

In data mining, frequent pattern mining is a fundamental mining method that has been applied to different kinds of databases, such as transactional databases , streaming databases and time series databases, and various application domains, such as bioinformatics ,Web click-stream analysis, and mobile environments. Among the issues of frequent pattern mining, the most famous are association rule mining and sequential pattern mining. One of the well-known algorithms for mining association rules is Apriori, which is the pioneer for efficiently mining association rules from large databases. Pattern growth based association rule mining algorithms such as FP Growth were then proposed. It is widely recognized that FP-Growth achieves a better performance than Apriori-based algorithms since it finds frequent itemsets without generating any candidate itemset and scans database just twice. But relative importance of each item is not considered in frequent pattern mining. To address this problem, weighted association rule mining was proposed, where each item is assigned particular weight.

In this framework, weights of items, such as unit profits of items in transaction databases, are considered. With this concept, even if some items appear infrequently, they might still be found if they have high weights.

However, the quantities of items are not considered yet. Therefore, it cannot satisfy the requirements of users who are interested in discovering the itemsets with high sales profits, since the profits are composed of unit profits, i.e., weights, and purchased quantities.

In view of this, utility mining emerges as an important mining method in data mining field. Mining high utility itemsets from databases refers to finding the itemsets with high profits. Here, the meaning of itemset utility is interestingness, importance or profitability of an item to users. Utility of items in a transaction database consists of two aspects:

- The importance of distinct items called external utility.
- The importance of items in transactions called internal utility.

A tree-based structure called IHUP-Tree as shown in Figure 3.1 is used to maintain the information about itemsets and their utilities. Each node of an IHUP-Tree consists of an item name, a TWU value and a support count.

IHUP algorithm has three steps:

- Construction of IHUP-Tree,
- Generation of HTWUIs and
- Identification of high utility itemsets

4 PROPOSED SCHEME

Mining the dataset according to its utility, will involve Utility Pattern Growth and Utility Pattern Growth+.

The framework of the proposed methods consists of four steps:

- Scan the database twice to construct a global UP-Tree with the first two strategies DGU and DGN.
- Recursively generate potential high utility itemsets from the global UP-Tree and local UP-Trees by UP-Growth algorithm with the third and fourth strategies [DLU and DLN].
- Identify actual high utility itemsets which are smaller than the set of PHUIs identified after mining.
- The Association rules are then generated to specify the relationship between the patterns identified.

By our effective strategies, the set of PHUIs will become much smaller than the set of HTWUIs. Moreover, the association rules generated help in better analysis of mined patterns.

To facilitate the mining performance and avoid scanning original database repeatedly, we use a compact tree structure, named UP-Tree (Utility Pattern Tree), to maintain the information of transactions and high utility itemsets. Two strategies are applied to minimize the overestimated utilities stored in the nodes of global UP-Tree.

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. 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.

This mining process involves the following modules.

1. UP-Growth.
2. UP-Growth+.
3. Downward Closure Property.
4. Association rules.

UP-Growth:-

1.1 Strategy DGU

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 itemsets if its TWU is less than the minimum utility threshold. Such an item is called an unpromising item.

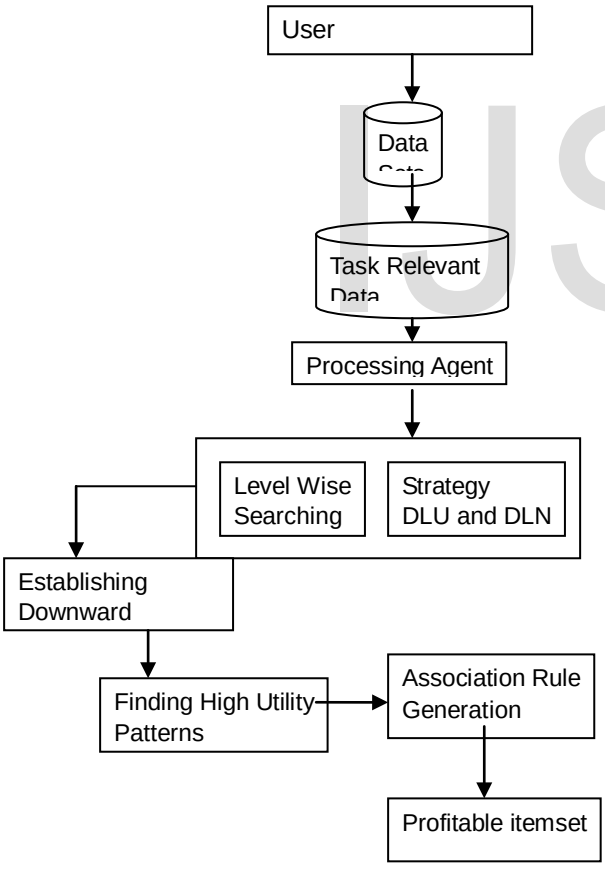

Fig. 1. System Architecture of Clustering Search Results

The overall system architecture will depict the way in which the association rule has been generated for the mined utility pattern.

1.2 Strategy DGN

The tree-based framework for high utility itemset mining applies the divide-and-conquer technique in mining processes. Thus, the search space can be divided into smaller subspaces. Insert_Reorganized_Transaction is said to be a function that is called to apply DGN during constructing a global UP-Tree. When organized transaction is is inserted into a global UP-Tree, Insert_Reorganized _ Transaction is called. At first the input are being taken for further process. Second proposed strategy for decreasing overestimated utilities is to remove the utilities of descendant nodes from their node utilities in global UP-Tree. The process is performed during the construction of the global UP-Tree.

UP-Growth+:-

2.1 Strategy DLU:

For this strategy, we maintain a minimum item utility table to keep minimum item utilities for all global promising items in the database. Minimum item utilities are utilized to reduce utilities of local unpromising items in conditional pattern bases instead of exact utilities. An estimated value for each local unpromising item is subtracted from the path utility of an extracted path. DLU can be recognized as local version of DGU. It provides a simple but useful schema to reduce overestimated utilities locally without an extra scan of original database.

2.2 Strategy DLN:

DLN is applied during inserting reorganized paths into a conditional UP-Tree. When an item is inserted into the conditional UP-Tree, the function Insert_Reorgnized_Path is called as detailed. Since the Tree must not contain the information about the items below i_m in the original UP-Tree, discard the utilities of descendant nodes related to i_m in the original UP-Tree while building $\{i_m\}$-Tree. Because actual utilities of the descendant nodes cannot be known, minimum item utilities to estimate the discarded utilities are used. Assume a reorganized path <DC> with support count 1 is inserted into a local UP-Tree. The node ND under root node is created or updated. The same as DLU, DLN can be recognized as local version of DGN. By the two strategies, overestimated utilities for itemsets can be locally reduced in a certain degree without losing any actual high utility itemset. After inserting all paths, the Tree is constructed completely with its updated path utilities.

The complete set of PHUIs is generated by recursively calling the procedure named UP-Growth given below. Initially, UPGrowth (TR, HR, null) is called, where TR is the global UP Tree and HR is the global header table. The algorithm starts from the bottom entry of header table and considers items first. By tracing all the first item hlinks, sum of its node utilities is calculated, that is, nusum(item). Thus a new PHUI (item): value is generated and {item}-CPB is constructed. By following {item}.hlink, the nodes related to {item} are found. By tracing these nodes to root, four paths are found. The num-

ber beside a path is path utility of the path, which equals to {item}.nu in each traversed path. These paths are collected into {item}-CPB.

UP-Growth achieves better performance than FP-Growth by using strategies DLU and DLN to decrease overestimated utilities of itemsets. The overestimated utilities can be closer to their actual utilities by eliminating the estimated utilities that are closer to actual utilities of unpromising items and descendant nodes.

2.3 Transaction weighted downward closure:

Downward closure property can be maintained in utility mining by applying the transaction weighted utility. A table will contain new term called total utility which is present along with transaction table.

Here the table contains new term called total utility for each transaction, total utility is calculated by multiplying the quantity of each item and the profit of the each item, then adding all item values in each transaction.

Therefore the transaction weighted utility of item G is 40. If the min_utility is set to 50 means item G is not high utility item. By using this approach weighted support cannot only reflect the importance of an itemset but also maintains the downward closure property during mining process. Although this approach considers the importance of items in some applications, items quantities in transaction are not taken into consideration yet and it still generates too many candidates to obtain high utility items.

2.4 Association Rules:

Association rule mining is to find out association rules that satisfy the predefined minimum support and confidence from a given database. Mining association rules consists of two steps [1]:

1. Finding all the frequent itemsets above predefined threshold utility.
2. Generating association rules from these frequent itemsets. The outcome depends upon the results of step1.

Steps to generate association rules are as follows.

- For each frequent itemset "l", generate all nonempty subsets of l.
- For every nonempty subset s of l, output the rule "s → (l-s)" if support_count(l) /support_count(s)>= min_conf where min_conf is minimum confidence threshold.

5 Implementation And Results

Transaction database is the input to the algorithm that is being used in this module.

1. Transaction table and profit table together will be the input. The rows in the transaction table will depict the item and quantity of that item purchased, specified alternately.
2. Profit file which is the parallel input to the algorithm. It has entries for all items and corresponding profit that is assigned for the particular item.

Two inputs, transaction dataset and the profit table, are retrieved and given as input for the algorithm UP-Growth.
The Initial strategy DGU is then applied to calculate TU, TWU of itemset. Output of the patterns calculated where transactions are sorted in descending order of TWU.

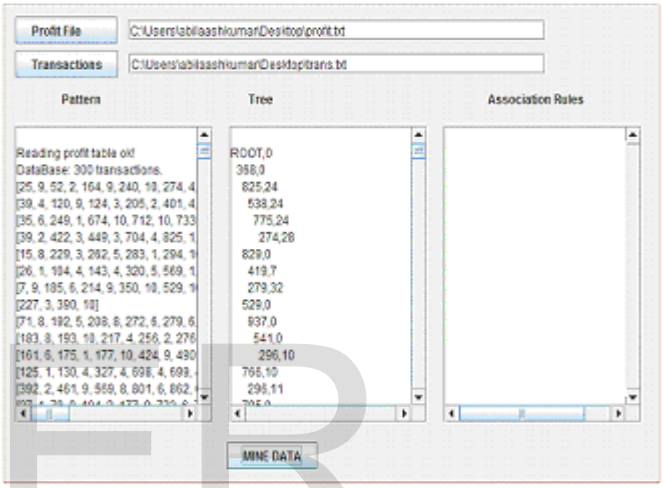

Fig. 2. Global UP-Tree

Then the strategy DLU is applied and CPB of each leaf is obtained. The CPB is used to construct the corresponding local trees of the leaf and the Potential High Utility Itemsets are mined from it. The PHUIs are displayed as output in Figure 3 for each leaf node that has the utility greater than threshold TWU. Association rules will be generated for the pattern or the tree that has been obtained for the transaction database. Utility of an itemset is defined as the product of its external utility and its internal utility. An itemset is called a high utility itemset if its utility is no less than a user-specified minimum utility threshold; otherwise, it is called a low utility itemset.

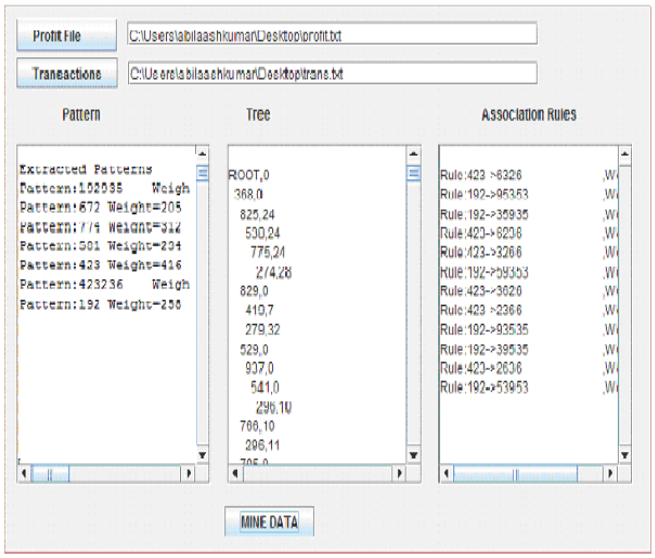

Fig. 3. Mined Patterns and Association Rules.

Mining high utility itemsets from databases is an important task has a wide range of applications such as website click stream analysis, business promotion in chain hypermarkets, cross-marketing in retail stores, online e-commerce management, and mobile commerce environment planning and even finding important patterns in biomedical applications.

6 Conclusion & Future Work

In our work the algorithm UP-Growth for mining high utility itemsets from transactional databases was studied. A data structure named UP-Tree was used for maintaining the information of high utility itemsets. Potential high utility itemsets were efficiently mined from UP-Tree with only two database scans. Association rules representing the relationship between mined patterns were also generated by the proposed method.

IBM synthetic datasets was used to perform a thorough performance evaluation. Results show that the strategies considerably improved performance by reducing both the search space and the number of candidates. Moreover, the algorithm outperforms the state-of-the-art algorithms substantially especially when databases contain lots of long transaction.

The association rules were extended to the mined patterns to better analyze the correlation between items. Thus the proposed methods effectively help in decision making of user by increasing the profit.

References

[1] Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, "Efficient Algorithm For Mining High Utility Itemsets From Transcational Database"IEEE Aug 2013.

[2] R. Agrawal and R. Srikant, "Mining Sequential Patterns, " in Proceedings of the 11th International Conference on Data Engineering, pp. 3-14, March, 1995.

[3] J. Han, J. Pei, Y. Yin, "Mining Frequent Patterns Without Candidate Generation," in Proceedings of the ACM-SIGMOD International Conference on Management of Data, pp. 1-12, 2000.

[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 Transactions on Knowledge and Data Engineering, Vol. 21, No.12, pp. 1708-1721, 2009.

[5] Yao. H and Hamilton. H. J, "Mining Itemset Utilities from Transaction Databases," IEEE Transactions on Knowledge and Data Engineering,Vol. 59, pp. 603–626, 2006.

[6] Y. Liu, W. Liao and A. Choudhary, "A Fast High Utility Itemsets Mining Algorithm," in Proceedings of the Utility-Based Data Mining Workshop, 2005.

[7] S. Shankar, T. P. Purusothoman, S. Jayanthi and N. Babu, "A Fast Algorithm for Mining High Utility Itemsets," Proceedings of IEEE International Advance Computing Conference (IACC), pp. 1459 - 1464, 2009.

[8] W. Wang, J. Yang, P. Yu, "Efficient Mining of Weighted Association Rules (WAR)," in Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), pp270-274, 2000.

[9] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," in Proceedings of the 20th VLDB Conference, pp. 487-499, 1994.

[10] R. Chan, Q. Yang and Y. Shen. "Mining High Utility Itemsets," in Proceedings of Third IEEE International Conference on Data Mining," pp. 19-26, November, 2003.

[11] A. Erwin, R. P. Gopalan and N. R. Achuthan, "Efficient Mining of High Utility Itemsets from Large Datasets," in Proceedings of 12th Pacific-Asia Conference, PAKDD, pp. 554-561, 2008.

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

[13] Frequent itemset mining implementations repository, http://fimi.cs.helsinki.fi/