International Journal of Current Trends in Engineering & Research (IJCTER)
e-ISSN 2455–1392 Volume 2 Issue 3, March 2016 pp. 171 - 178
Scientific Journal Impact Factor : 3.468
http://www.ijcter.com

A Survey on Discovery of High Utility Itemset Mining from Transactional Databases

Shilpa K.K.[1], Syed Farook K.[2]

[1]PG Scholar, [2]Asst. Professor
[1,2]Department of Computer Science and Engineering MES College of Engineering, Kuttippuram Kerala, 679573, India

Abstract—Utility mining is a recent topic in the field of data mining. To find out the highest utility itemsets, in terms of profit, quantity, cost or other user preferences is the main goal of utility Mining. The high utility itemset (HUI) mining using a transactional database is to explore itemsets that have utility above a user specified threshold. High utility itemset mining is an addition to frequent itemset mining, which consists of itemsets discovery that are frequently bought together by customers. Rare items considered as high utility itemsets in many real-life applications. For several decision making domains like business transactions, medical, security, fraudulent transactions, retail communities etc, rare itemsets give helpful information. For example, in a supermarket, customers purchase microwave ovens or frying pans infrequently as compared to bread, washing powder, soap. But the later transactions yield less profit compared to former transactions for the supermarket. Thus the high profit rare itemsets are found to be very helpful in many application areas. The performance of utility mining can be further enhanced with compact and lossless representation of high utility itemsets. This study analyzes various methods for high utility itemset mining using transactional databases.

Keywords—Association rule mining, Frequent itemset, Utility mining, Closed high utility itemset, Data mining

I. INTRODUCTION

Currently there has been an aggressive growth in the generation and manipulation of electronic information as more and more activities are digitalized. Any organization or enterprise have started to figure out that the information acquired over years is an essential strategic asset and they also realize that there is potential business intelligence concealed in the huge amount of data. For that, what these enterprises want is a method that allows them to extract the most useful information from acquired data. The field of data mining suggests such methods which assessing the current data and deriving hidden information that would be helpful in future prediction, pattern recognition and decision making.

A. Data Mining

For the period of the last ten years, data mining, also known as Knowledge Discovery in Databases (KDD) has fixed its position as a leading and major research area. The purpose of data mining is to obtain higher level hidden information from plenty of raw data. Data mining has been utilized in various data domains. Data mining can be considered as an algorithmic process that carries data as input and yields pat-terns, such as classification rules, itemsets, association rules, or summaries, as output. Data Mining jobs can be categorized into two categories, descriptive mining and predictive mining. The descriptive mining methods such as clustering, association rule discovery, sequential pattern discovery, is used to discover human-interpretable patterns that illustrate the data. The predictive mining methods like classification, regression, deviation detection, utilize some variables to guess unidentified or future values of other variables.

B. Association Rule Mining

Association rule mining (ARM) is a well-liked method for finding co-occurrences, correlations, frequent patterns, associ-ations between items in a set of transactions or a database. Rules with confidence and support above user defined thresh-olds (minconf and minsup) were found. The newer data structures and algorithms are being developed as per data and its complexity continues to grow. Association rule mining [1][2] process can be divided into two steps. The first step is to find all frequent itemsets (or say large itemsets) in databases and the second step is to generate association rules using the discovered frequent itemsets. The rule evaluation metrics are:

Support(s) of an itemset is the fraction of transactions that include the itemset.

Confidence(c) measures how often items in Y appear in transactions that contain X.

ARM is commonly used in market basket analysis. For exam-ple, frequent itemsets can be originated by analyzing market basket data and then association rules can be produced by guessing the purchase of other items by conditional probability.

C. Frequent Itemset Mining

Frequent itemsets are the itemsets that come about fre-quently in the transaction dataset. The goal of frequent itemset mining (FIM) is to discover all the itemsets that are frequently bought together by customers in a transaction dataset. The frequency of an itemset X is the chance of X appearing in a transaction T. A frequent itemset is the itemset having frequency support larger a minimum user specified threshold. It satisfies the downward closure property or anti monotone property. It states that if an itemset is frequent, then all its subsets must also be frequent. Support of an itemset never exceeds the support of its subsets. The Support value of an itemset is the percentage of transactions that contain the itemset. FIM may discover a large amount of frequent but low revenue itemsets. So the major limitations caused by FIM are:

1) Treats all items with same price.
2) Assumes that each item can’t appear more than once in each transaction.

II. HIGH UTILITY ITEMSET MINING

The restrictions of frequent or rare itemset mining motivated researchers to introduce a utility based mining approach, which enables a user to conveniently express his or her views regarding the usefulness of itemsets as utility values and then identify itemsets with high utility values higher than a threshold. Mining high utility itemsets from databases means to finding the itemsets with high profits. Identification of the itemsets with high utilities is called as high utility itemset mining (HUIM). Let I = f a_1, a_2,..., a_M g be a finite set of distinct items. A transactional database D = f T_1,T_2,...,T_N g is a set of transactions, where each transaction is a subset of I and has a unique identifier R, called Tid. Each item in transaction is associated with a positive real number called external utility and every item in the transaction has a real number called its internal utility. Cost, quantity, profit or any other user terms of preference can be used to compute the utility. Utility of items in a transaction database comprises of two aspects:

The unit profit for the item, which is called external utility.

The quantity of the item sold in a transaction, which is called internal utility.

TABLE I AN EXAMPLE TRANSACTIONAL DATABASE

TID	Transaction	TU
T1	A(2), C(1), D(2)	25
T2	B(2), C(20)	90
T3	B(2), C(1), D(10)	60

TABLE II PROFIT TABLE

Item	A	B	C	D
Unit Profit	4	15	3	7

Utility of an itemset is described as the product of its external utility and its internal utility. An itemset is called a low utility itemset, if its utility is fewer than a user spec-ified minimum utility threshold. Let TABLE I be a database containing three transactions. There are four items in the transactions, respectively denoted A to D. Each row in TABLE I shows a transaction, in which every letter represents an item and has a purchase quantity (internal utility). The unit profit of each item is shown in TABLE II (external utility). Assume the profits of the items are 4, 15, 3 and 7 respectively. HUIM is not an simple task as the downward closure property in frequent itemset mining doesn't hold in utility mining. In other words, the search space for mining high utility itemsets cannot be directly reduced because a superset of a low utility itemset can be a high utility itemset. To facilitate the mining task, Liu et al. introduced the concept of transaction-weighted downward closure (TWDC) property [3], which is based on following definitions.

Definition 1. Transaction Utility ($TU(T_R)$): It is the sum of all the utilities of all items in that transaction.

Definition 2. Transaction-Weighted Utilization of an itemset (TWU(X)): It is the sum of the transaction utilities of all the transactions containing X.

Definition 3. High Transaction-Weighted Utilization Itemset (HTWUI): An itemset X is a HTWUI iff TWU(X) greater than or equal to its user specified minimum utility threshold. Definition 4. Transaction-Weighted Downward Closure (TWDC) property: It states that for any itemset X that is not a HTWUI, all its supersets are low utility itemsets.

The other definitions related to HUIM are:

Definition 5. Actual utility of an itemset in a database (au(X)): It is defined as summation of utility values of X in all transactions containing X.

Definition 6. High Utility Itemset (HUI): An itemset X is a HUI iff au(X) greater than or equal to user specified minimum utility threshold.

A generic model of HUIM is illustrated in Fig. 1. The data is given first and then identifies the high utility itemsets from that by comparing each itemset with the minimum utility threshold value and finally the reduced high utility itemset is produced as output.

A. Improving the efficiency of HUIM

The main purpose of association rule mining (ARM) is to find out the interesting associations or relations among the different itemsets in database. Interestingness actions can play a key role in

knowledge discovery. These actions are planned for selecting and ranking patterns corresponding to their potential interest to the user. Hence the factors to be examined while improving the performance of high utility itemset mining are as follows:

- Reducing the number of scans in the original database.
- Minimize memory utilization by reducing the search space.
- Reducing the total execution and computation time.
- Reducing the resource utilization.
- Increase the performance in terms of time and space complexity.

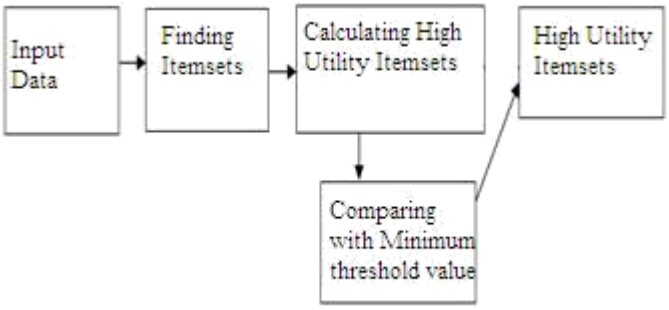

Fig. 1. Process Diagram of HUIM

B. Closed High Utility Itemset Mining

The concept of closed itemset is incorporated with high utility itemset mining to develop a representation named Closed High Utility Itemset (CHUI)[8]. CHUIs mining may discover only the high utility itemsets that are closed. An itemset X is a closed itemset such that there does not exist an itemset Y strictly included in X that has the same support. The number of transactions that contain the itemset is termed as support of an itemset.

A HUI is said to be closed iff for any itemset X it should be equals to its closure and also its absolute utility should be no less than user specified minimum utility threshold. The itemset X is also annotated with utility unit array so that its actual utility can be inferred from the utility unit array directly without scanning the original database. The utility unit array helps to represent the set of closed HUIs as lossless, which is defined as follows.

Definition 7. Utility unit array (V(X)): Let V(X)=[v_1, v_2,...,v_k]. It contains a list of K utility values. The ith utility value v_i is denoted as V(X,a_i) represents the sum of actual utilities of a_i in transactions containing X.

III. LITERATURE REVIEW

Recently, one of most challenging data mining tasks is mining of high utility itemsets efficiently. HUIM is helpful in decision making method of various applications such as retail marketing, website click stream analysis, mobile commerce environment and biomedical applications. This review includes several high utility itemset mining methods using transactional database.

Liu et al. in [3] proposed the Two-Phase algorithm for fast discovering all high utility itemsets using the downward closure property. The property states that any superset of a non frequent itemset is also non frequent. By using this property the search space is reduced early by pruning non frequent itemsets. The two phase algorithm generates high utility itemsets candidates in a level wise way. It explain transaction weighted utilization in Phase I. So during the level wise search only the combinations of high transaction weighted utilization itemsets are suppliment into the candidate set

at each level. In phase II, only one extra database scan is done to sort out the overestimated itemsets. In terms of speed and memory cost it performs very well both on synthetic and real databases, even on large databases. The database scanning time is, however, a bottleneck of the approach.

To overcome the problem of two phase algorithm Yu-Chiang Li et al. in [4] proposed an isolated items discarding strategy, abbreviated as IIDS, to minimize the number of candidates. By pruning isolated items during the level wise search, the number of candidate itemsets for HTWUIs in phase I can be reduced successfully. But this approach also scans database multiple times and needs a candidate generation and test scheme to find high utility itemsets.

Chun-Wei Lin et al. in [5] proposed a new utility mining approach with the help of a tree structure. A new tree structure called the high utility pattern tree (HUP tree) is first considered to keep related information for utility mining. To mine high utility itemsets a mining method named HUP growth based on the projected HUP tree structure, is next considered. The whole process of exploring the high utility itemsets from a database therefore consists of the construction process of the HUP tree and the mining process of the HUP-Growth. It is also clear that the proposed approach for mining high utility itemsets has a better performance than the two phase approach in execution time. Moreover, different item ordering ways may change the numbers of nodes in the HUP tree.

To provide high utility itemsets mining capably for handling incremental databases, while allowing for many insertions, deletions, and modifications with the currently existing mem-ory size and to keep away from multiple database scans, C. F. Ahmed et al. in [6] proposed incremental high utility pattern (IHUP) mining. They use an IHUP-Tree to keep up the data of high utility itemsets and transactions. Every node in IHUP-Tree contained of an item name, a support count, and a TWU value. The structure of the algorithm consists of three steps:

1) The construction of IHUP-Tree.
2) The generation of HTWUIs.
3) Identification of high utility itemsets.

In step 1, items in the transaction are rearranged in a fixed order like lexicographic order, support descending order or TWU descending order. The rearranged transactions are then inserted into the IHUP-Tree. In Step 2 Candidate itemsets are taken out from IHUP-Tree by FP growth algorithm. And in Step 3 actual high utility itemsets are recognized with an additional database scan. Although IHUP can create a tree and explore high utility itemsets with two database scans, it produces a large number of candidates by applying the TWU model.

To address problem of creating a large number of candi-dates, V. S. Tseng et al. in [7] proposed UP-Growth method and it uses PHU (Potential High Utility) model. For dropping the number of candidate itemsets, the UP growth utilizes four strategies, DGU (Discarding Global Unpromising items), DGN (Decreasing Global Node utilities), DLU (Discarding Local Unpromising items), and DLN (Decreasing Local Node utilities). Moreover, it constructs a tree structure, named UP tree, with two database scans and carry out mining high utility itemsets.

In other words, it needs three database scans for identifying high utility itemsets. In the first database scan, TWU values of each item are gathered. In the second database scan, items having fewer TWU values than the user specified minimum utility threshold are detached from each transaction. Further, items in transactions are sorted according to TWU descending order and the transactions are placed into the UP-Tree. In this stage, DGU and DGN are applied for minimizing overesti-mated utilities. After that, high utility itemsets are produced from the UP tree with DLU and DLN.

A limitation of many high utility itemset mining algo-rithms is that they produce too many

itemsets as output. To achieve high efficiency for the mining task and to provide a concise mining result to users, Cheng-Wei Wu et al. in [8] proposed a novel framework for mining closed high utility itemsets (CHUIs), which serves as a compact and lossless representation of HUIs by using combination of Closed High Utility Itemset Discovery (CHUD) and Derive All High Utility Itemsets (DAHU) methods. The concept of closed itemset is incorporated with HUI mining in this work. A HUI is said to be closed iff for any itemset X it should be equals to its closure and also its absolute utility should be no less than user specified utility threshold.

CHUD (Closed High Utility itemset Discovery) is an ef-ficient depth-first search algorithm and it considers vertical database to discover CHUIs. It is treated as one of the presently best methods to mine closed itemsets. CHUD is adapted for mining CHUIs and include several effective strategies for reducing the number of candidates generated in Phase-I. CHUD adopts an IT-Tree (Itemset-Tidset pair Tree) to find CHUIs.

In an IT-Tree, each node N(X) consists of an itemset X, its Tidset g(X), and two ordered sets of items named PREVSET(X) and POST-SET(X). The IT-Tree is recursively explored by the CHUD algorithm just before all closed item-sets that are HTWUIs are generated. Different from the other algorithms, each node N(X) of the IT-Tree is connected with an estimated utility value EstU(X).

A data structure called TU-Table (Transaction Utility Table) is adopted for storing the transaction utilities of transactions. It is a list of pairs hR,TU(T_R)i sorted according to a increasing order of support, where the first value is a TID R and the second value is the transaction utility of T_R. The utilities of unpromising items can be removed from TU-Table. So for each TID R, the value TU(T_R) can be efficiently retrieved from the TU-Table.

IV. PERFORMANCE ANALYSIS

For the purpose of performance analysis, the different mining methods for high utility itemsets discussed includes:

Two-Phase IIDS

HUP-Growth IHUP

UP-Growth

CHUD+DAHU

The pros and cons of the above mentioned methods are shown in TABLE III. The problem of repetition in high utility itemset mining is solved by proposing a lossless and compact representation called closed high utility itemsets.

TABLE III COMPARISON TABLE

METHOD	PROS	CONS
Two-Phase	• Reduces search space of utility mining.	• Generates too many candidates. • Demands multiple database scans. • Suffers from problem of level-wise candidate generation and test.
IIDS	• Reduced the number of candidates by pruning isolated items.	• Still scans database multiple times. • Suffers from problem of level-wise candidate generation and test.
HUP-Growth	• Better performance than Two-Phase method. • Without level-wise generation of candidate itemsets, high utility itemset can be derived effectively from HUP tree.	• Efficiency may varies with predefined minimum high utility value.
IHUP	• Avoided multiple database scans. • Maintain the information about itemsets and their utilities in the form of tree structure.	• Generates large number of candidates.
UP-Growth	• Reduced the number of candidate itemsets. • No unpromising item in reorganized transactions thus execution time for phase-2 can be further reduced.	• Complex for evaluation due to tree structure. • Will work well especially when original database contains lots of unpromising items. • Occupies lots of memory.
CHUD+DAHU	• Achieved high efficiency for mining task. • Provided a concise and lossless mining results to users.	• Fail to recover all HUIs due to memory limitation when there are too many HUIs in database. • Mayn't achieve a massive reduction on very sparse datasets.

The combination of CHUD and DAHU method for mining high utility itemsets results in improved performance interms of execution time and memory consumption when compared with other methods. It requires only less memory usage as well as less time for the processing. Although this method provide better results, it has certain restrictions too. The major difficulty is the memory consumption due to traversal and sorting procedure and so further enhancements has to be done to reduce the memory utilization.

V. CONCLUSION

The frequent itemset mining is based on the belief that the itemsets which appear more often in the transaction databases are of more significance to the user. However the practical value of mining the frequent itemset by considering only the regularity of occurence of the itemsets is challenged in many application domains such as retail research. It has been that in many valid applications that the itemsets that give the most in terms of some user defined utility function (for e.g. profit) are not essentially frequent itemsets. Utility mining attempts to solve this by using item utilities as an indicative quantity of the importance of that item in the user's point of view. Utility mining is a comparatively new area of research and most of methods are focused towards reducing the search space while searching for the high utility itemsets. This paper presents a survey on various methods for mining of high utility itemsets.

REFERENCES

[1] R. Agrawal and R. Srikant, ”Fast algorithm for mining association rules,” The International Conference on Very Large Data Bases, pp. 487499, 1994.
[2] R. Agrawal, T. Imielinksi, and A. Swami, ”Mining association rules between sets of items in large database,” The ACM SIGMOD International Conference on Management of Data, pp. 207216, 1993.
[3] Y. Liu, W. Liao, and A. Choudhary, ”A fast high utility itemsets mining algorithm,” Proc. Utility-Based Data Mining Workshop, pp. 9099, 2005.
[4] Y.-C. Li, J.-S. Yeh, and C.-C. Chang, ”Isolated items discarding strategy for discovering high utility itemsets,” Data Knowl. Eng., vol. 64, no. 1, pp. 198217, 2008.
[5] C.-W. Lin, T.-P. Hong, and W.-H. Lu, ”An effective tree structure for mining high utility itemsets,” Expert Syst. with Appl., vol. 38, no. 6, pp. 74197424, 2011.
[6] 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. on Knowl. Data Eng., vol. 21, no. 12, pp. 17081721, Dec. 2009.
[7] V. S. Tseng, C.-W. Wu, B.-E. Shie, and P. S. Yu, ”UP-Growth: An efficient algorithm for high utility itemset mining,” Proc. ACM SIGKDD Int’l Conf. on Knowl. Discov. and Data Mining, pp. 253262, 2010.
[8] V. S. Tseng, C.-W Wu, Philippe F.-Viger, and Philip S. Yu, ”Efficient Algorithms for Mining the Concise and Lossless Representation of High Utility Itemsets,” IEEE Trans. on Knowl. Data Eng., Vol. 27, no. 3, March 2015.

