A Novel Algorithm to Estimate High Utility Item Sets and to Achieve Privacy based on Transaction Splitting

R.Chitra Devi
PG Scholar
Department of Computer Science & Engineering
Christian College of Engineering and Technology,Oddanchatram, Tamilnadu-624619, India

S.Venkatesh Babu
Assistant Professor
Department of Computer Science & Engineering
Christian College of Engineering and Technology,Oddanchatram, Tamilnadu-624619, India

Abstract

Data mining methods are very useful in order to determine the hidden motivating and necessary information from the huge database. Extracting high utility item sets (HUI) and frequent item set mining are become a hectic task in data mining techniques. Frequent item set is a famous technique for identifying the items purchased together. But, frequent item set treats all the items in the same precedence. Even though it could identify the items, in some cases it fails to address the quantity of the rare product. HUI determines the item with high profit. Frequent item set mining is the major issue in data mining because it does not fulfill the requirement of users who desire to discover item sets with high utilities such as high profits. HUI determines the item with high profit. In many cases HUI will reduce the efficiency. The utility represents the importance of the product. In this paper, to overcome these issue we introduced the new concept of finding HUI with the closest item set. For the longer transactions new splitting technique was introduced rather than the truncation to maintain the privacy.

Keywords: Frequent item set mining, Utility Mining, Data Mining, closest item set mining

I. INTRODUCTION

FIM treats all items as having the same importance weight and it assumes that every item in a transaction appears in a same precedence, i.e., an item can be either present or absent in a transaction, which does not indicate its purchase quantity of the product. Hence, FIM doe not satisfy the requirement of users who desire to identify the item sets with high utilities such as high profits.

The utility of an item set represents its importance which can be measured in terms of weight, profit, cost, quantity depending on the user preference An itemset is called a high utility itemset (HUI) if its utility is no less than a user-specified minimum utility threshold; otherwise, it is called a low utility itemset. Utility mining is a wide range of applications such as website click stream analysis, cross marketing in retail stores, mobile commerce environment. However, HUI mining is not an easy task since the downward closure property in FIM does not hold in utility mining. HUI cannot be directly reduced as it is done in FIM because a superset of a low utility item set can be a high utility item . A very large number of high utility item sets makes it difficult for the users to comprehend the results. It cause the algorithms to become ineffective in terms of time and memory requirement, or even run out of memory. It is analyzed that the more high utility item sets the algorithms generate, the more processing they gain.

The result of the mining task decreases greatly for low minimum utility thresholds or when dealing with dense databases .In FIM, to reduce the computational cost of the mining task and present fewer but more important to users, many studies focused on developing comprehend representations for the results.

Frequent item set will find item set that occur in transaction more frequently than given threshold.FIM treat that every the item having the same profit. Differential privacy offers strong privacy of released data without making assumption about attacker background knowledge. Sequential pattern mining is defined to finding statistically relevant pattern. The customer buys a mobile phone, data cable and memory card if it occur frequently in a shopping history database is a sequential pattern. Association rule can be defined as{X,Y}=>{Z} the customer buys X,Y product and they like to buy Z. The traditional model of FIM may discover a large amount of frequent but low revenue item sets and lose some information on valuable item sets having some low frequencies. These problems are caused by the facts that the every item sets have same priority. For example, if a customer buys a very expensive product or just a piece of small product, it is viewed as being equally important. FIM cannot satisfy the requirement of users who desire to notice the item sets with high profit.

High Utility Item set is based on the profit of the product. While addressing the utility few redundancy may occur, so need to reduce the redundancy in the High Utility Itemset.After finding the High Utility Item sets find the closest itemset which consist of related itemset .The unit profits and purchased quantities of the rare products are not taken in frequent item set mining.

The reminder of this paper is organized as follows. Section II, describes the Related Works. Section III, describes the Proposed work, Section IV summarizes the conclusion.

II. RELATED WORK

A. *High Utility Item sets and Frequent Item Set*

1) CTU-PROL Algorithm

This algorithm has a enhancing performance by decreasing both the search space and the number of candidates. Especially UP-Growth, not only decrease the number of candidates keys effectively but also outperform other algorithms substantially in terms of runtime. They not provide comprehend result to the users and lots of unwanted candidates are pruned carries more computations and thus slower.

2) Sparse Vector Mechanism

Discovers frequent item sets using a small portion of the privacy budget. Algorithm use the remaining privacy budget to efficiently and effectively build a differentially private FP-tree. For longer transactions the computation may take for longer time so if truncation is done then information loss may occur .The user always expect the result should be in a concise manner without any information loss.

3) Projection Based Algorithm

Each candidate which appears in the continues manner could be thought of as a query condition, and the tuples in the database that contained these sequences were projected. The support count for each candidate sequence could also be easily obtained from the tuples.Effectively skip unpromising item sets and thus further save time for the execution. The indexing mechanism can imitate the traditional projection algorithms to achieve the aim of projecting the databases for the mining process. If Transaction is longer then the performance will be degraded. The user will not get a concise result and redundancy may occur. Some redundant information may annoy the user,so they should be pruned.

4) Privbasis Algorithm

It faces the challenge of high dimensionality problem by projecting the input dataset onto a small number of selected dimensions that one cares about. In fact, PrivBasis often uses several sets of dimensions for such projections, to avoid any one set containing number of dimensions. Each basis in B corresponds to one such set of dimensions for projection. Our techniques are able to select which set of dimensions are more helpful for the need of finding the k most frequent item sets. When the column M is larger than the truncated frequency approach is completely ineffective.

The reason why the Truncation Frequency(TF) method does not grow is that when one needs to select the top k item sets from a large set U of candidates, the large size of candidate may causes two difficulties. The first is regarding the running time. Even if every single low-frequency itemset in U is chosen with only a small probability, the sheer number of such low-frequency item sets means that the selected item sets likely include many infrequent ones. The TF technique tries to address the running time challenge by pruning the search space, but it does not address the accuracy challenge. This notice the problem caused by a larger candidate set, but not the root cause.

5) IUPG-Algorithm

These algorithms are experimented on synthetic datasets and real time datasets for different support threshold.IUPG algorithm performs well than UPG algorithm for different support values.IUPG algorithm scales well as the size of the transaction database increases. The goal of high utility mining is to discover all the high utility item sets whose utility values are more than the user specified threshold in a transaction database.

III. PROPOSED WORK

The goal of utility mining is to discover all patterns whose utility values are higher than the user specified threshold. It overcomes the problem of traditional frequent patterns mining, which ignores the different utility values of item set. It mainly focus on the concept of finding high utility item set with its closest item set. For longer transaction new splitting method is used to maintain the privacy.

A. *High Utility Item Set*

An item set is called a high utility item set (HUI) if its utility is no less than a user-specified minimum utility threshold; otherwise, it is called a low utility item set.

B. *Transaction Utility And Total Utility*

External utility is defined as the importance of distinct items and the Internal utility is the importance of the items in the transaction. Utility is calculated by multiplying the internal and external utility. Total utility is calculated by adding all transaction utility.

C. *Transaction Weight Utilization*

The transaction-weighted utilization (TWU) of an item set X is the sum of the transaction utilities of all the transactions containing X, which is denoted as TWU(X).

D. *The Transaction-Weighted Downward Closure(TWDC)*

For any item set X, if X is not a HTWUI, any superset of X is a low utility item set.

E. *Local Promising Item*

An item imp is called a local promising item in {air}-CPB if up(imp, {air}-CPB) is no smaller than minutil.

F. *Differential Privacy*

Differential privacy has gradually emerged as the de facto standard notion of privacy in data analysis. For two databases D and D', they are neighboring databases if they differ by at most one record.The amount of injected noise is carefully calibrated to the sensitivity. The sensitivity of count queries is used to measure the maximum possible change in the outputs over any two neighboring databases.

G. *Weighted Splitting Operation*

Consider a transaction t whose length exceeds the maximal length constraint Lm. A function f divides t into multiple subsets t1; . . . ;tk, where ti is assigned a weight wi and the length of ti is under the length constraint Lm.

H. *Proposed Work*

In the proposed scenario, we introduced the algorithm named as improved splitting algorithm to overcome the privacy issues in the industrial applications. This proposed algorithm is mainly focused on the achieving the higher privacy for high utility item sets. It also ensures the high instance efficiency by improving the service and privacy tradeoff in the transformed database. Apply the privacy conceptions in the important high utility information and reveal only the particular users.

I. *Architecture*

Fig.3.1 illustrates that it consist of two database one for storing the transactions and another for profit details.High Utility Items will be found by using AprioriHC-D(AprioriHC algorithm with Discarding unpromising and isolated items) remove the unfrequented item set. With the help of CHUD (Closed High Utility Item set Discovery) find the closest item set. An item set is closed if none of its immediate supersets has same support as item sets. An item set is maximal frequent if none of its immediate supersets is frequent.

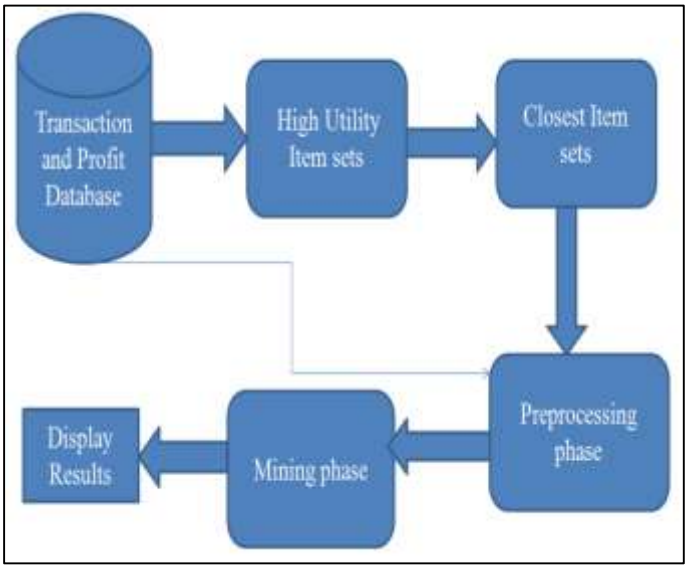

Fig. 1: Overall architecture

After finding the closest item set remove the unpromising items and reduce the redundancy. The AprioriHC and AprioriHC-D are based on Apriori algorithm. They use a horizontal database and explore the search space of CHUIs in a breadth-first search. The algorithm AprioriHC is regarded as a baseline algorithm in this work and AprioriHC-D is an improved version of AprioriHC.DAHU(Derive All High Utility Item sets)is used to derive all the all the high utility from the DAHU tree . For a

longer transaction, if truncation is done then information loss may occur.Instead,use splitting methods to split the longer transaction. Find the frequent item set and display the result.

J. Process Flow

Figure.1.5 illustrates that it have two database one for storing the transactions and another for profit details. High Utility Items will be found by using AprioriHC-D (AprioriHC algorithm with Discarding unpromising and isolated items) remove the unfrequented item set. With the help of CHUD (Closed High Utility Item set Discovery) find the closest itm set. An item set is closed if none of its immediate supersets has same support as item sets. An item set is maximal frequent if none of its immediate supersets is frequent. After finding the closest item set remove the unpromising items and reduce the redundancy.

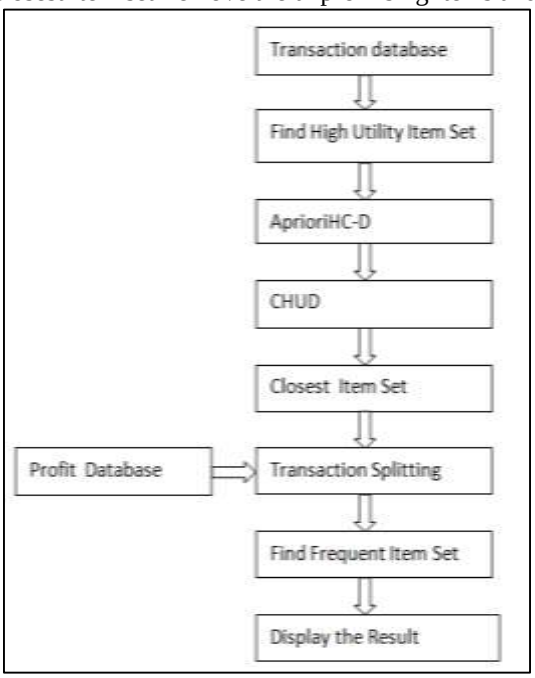

Fig. 2: Process Flow

The AprioriHC and AprioriHC-D are based on Apriori algorithm. They use a horizontal database and explore the search space of CHUIs in a breadth-first search. The algorithm AprioriHC is regarded as a baseline algorithm in this work and AprioriHC-D is an improved version of AprioriHC.DAHU (Derive All High Utility Item sets) is used to derive all the all the high utility from the DAHU tree. For a longer transaction, if truncation is done then information loss may occur.Instead, use splitting methods to split the longer transaction. Find the frequent item set and display the result.

IV. MODULES

A. Finding High Utility Item sets

From the transaction and profit database find the high utility item set. High utility item set: its utility is no less than a user-specified threshold or min_util the basic meaning of utility is the interestedness/ importance/profitability of items to the users. With the help of derive all high utility (DAHU) tree find HUI and prune the infrequent items. Then find the closest item set .An item set is closed if none of its immediate supersets has same support as item sets .An item set is maximal frequent if none of its immediate supersets is frequent. Utility Items will be found by using AprioriHC-D (AprioriHC algorithm with Discarding unpromising and isolated items) remove the unfrequented item set.

Algorithm:

Input: Plist the effective information list of item set P,initially empty; lists, the set of utility-lists of all P's 1- extensions; minutil, the minimum utility threshold.

Output: all the high utility itemsets with P

1) for each Plist X in EILs do
2) if SUM(X.iutils)≥minutil then
3) output the extension associated with X;
4) end if
5) if SUM(X.iutils)+SUM(X.rutils) ≥ minutility then
6) exEILs = NULL;
7) for each effective information list Y after X in EILs do

8) exEILs = exEILs+Build(Plist, X, Y);
9) end for
10) HUI(X, exEILs, minutillity);
11) end if
12) end for

B. Preprocessing Phase

With the help of CHUD (Closed High Utility Item set Discovery) find the closest item set. An item set is closed if none of its immediate supersets has same support as item sets. An item set is maximal frequent if none of its immediate supersets is frequent. After finding the closest item set remove the unpromising items and reduce the redundancy. The AprioriHC and AprioriHC-D are based on Apriori algorithm. They use a horizontal database and explore the search space of CHUIs in a breadth-first search. In preprocessing phase extract some statistical information from the original database .For a longer transaction, if truncation is done then information loss may occur. Instead, use splitting methods to split the longer transaction.

C. Mining Phase

In mining phase find the frequent item sets which have the high utility and display the result without redundancy.

V. EXPERIMENTAL WORK AND RESULT ANALYSIS

The process of CHUD and DAHU in Phase I is the same as that of CHUD. In Phase II, CHUD and DAHU first identifies CHUIs from candidates and uses CHUIs to derive all high utility.In the experiments, we do not combine AprioriHC/AprioriHC-D with DAHU because CHUD outperforms these algorithms, as it will be shown, and they produce the same output. The complete execution time of UPGrowth is less than CHUD, initially. But as the min_utility threshold became smaller, CHUD becomes faster (up to twice faster than UP-Growth). The reason why the performance gap between CHUD and UP-Growth is smaller for Food mart than for Mushroom is due to the fact that Food mart is a sparse dataset.

Number of Candidates in Phase I on Foodmart

Minimum Utility	#Cand. for Two-Phase	#Cand. for AprioriHC	#Cand. for AprioriHC-D
0.1%	1,728	1,641	1,627
0.05%	62,514	2,981	2,936
0.01%	232,505	6,345	6,345
0.005%	233,185	6,657	6,657

Fig. 3: Sample comparison on Foodmart

In private frequent mining, it achieves better performance. Compared with TT, we divide each long transaction into subsets and evenly assign weights among the subsets. It benefits the quantification of information loss and significantly improves the utility of the results. For PB, it privately samples items to construct a basis set and adds noise to the support of itemsets covered by bases. However, when the differences between the supports of items are small, it is very likely to sample infrequent items, which leads to poor performance in term of F-score.

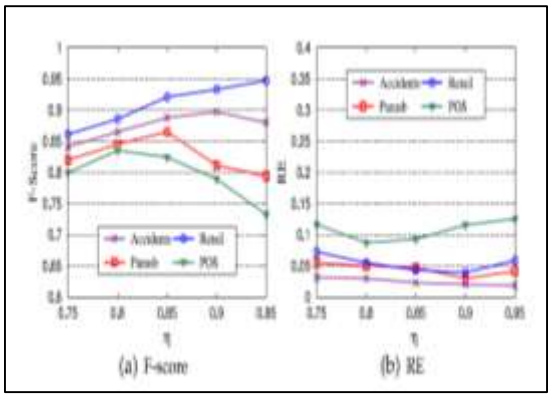

Fig. 4: Effect of maximal length constrain

VI. CONCLUSION

Frequent item set mining treats all the item in the same level so it is difficult to find the items which have the high utility. FIM mines the items which occur frequently but, HUI (high utility item set were used to identify the items with high profit or according to the user interestedness. Even in the HUI redundancy may occur which doesn't have any comprehend result. For longer transactions the truncation technique is used which may loss some information. Here, high utility item set will be found without any redundancy and for longer transaction splitting technique is used rather than truncation.

REFERENCES

[1] Sen su Shengzhi zu ; " Differentially Private Frequent Item Set Mining Via Transaction Splitting"IEEE Trans. Knowl. Data Eng", Volume:27, Issue 7 July 1 2015
[2] 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. Knowl. Data Eng., vol. 21, no. 12, pp.1708 -1721 2009
[3] Tseng V.S.,"Efficient Algorithm For Mining Concise And Lossless Representation For Mining High Utility Itemsets "IEEE Trans. Knowl. Data Eng", Volume:27, Issue 3 March 1 2015
[4] G.-C. , T.-P. , V. and S. Tseng "An efficient projection-based indexing approach for mining high utility itemsets", Knowl. Inf. Syst, vol. 38, no. 1, pp.85 -107 2014
[5] R. Chan, Q. Yang and Y. Shen "Mining high utility itemsets", Proc. IEEE Int. Conf. Data Min., pp.19 -26
[6] N. Li, W. Qardaji, D. Su and J. Cao "Privbasis: Frequent itemset mining with differential privacy,", Proc. VLDB Endowment, vol. 5, no. 11, pp.1340 -1351 2012
[7] L. Bonomi and L. Xiong "A two-phase algorithm for mining sequential patterns with differential privacy,", Proc. 22nd ACM Conf. Inf. Knowl. Manage., pp.269 -278 2013
[8] E. Shen and T. Yu "Mining frequent graph patterns with differential privacy", Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp.545 -553 2013.
[9] C.-W. , T.-P. and W.-H. "An effective tree structure for mining high utility itemsets", Expert Syst. Appl., vol. 38, no. 6, pp.7419 -7424 2011.
[10] R. Agrawal and R. Srikant "Fast algorithms for mining association rules", Proc. 20th Int. Conf. Very Large Data Bases, pp.487 -499
[11] A. Erwin, R. P. Gopalan and N. R. Achuthan "Efficient mining of high utility itemsets from large datasets", Proc. Int. Conf. Pacific-Asia Conf. Knowl. Discovery Data Mining, pp.554 -561
[12] Y. Liu, W. Liao and A. Choudhary "A fast high utility itemsets mining algorithm", Proc. Utility-Based Data Mining Workshop, pp.90 -99
[13] C.-W. , T.-P. and W.-H. "An effective tree structure for mining high utility itemsets", Expert Syst. Appl., vol. 38, no. 6, pp.7419 -7424 2011
[14] 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. Conf. Knowl. Discov. Data Mining, pp.253 -262
[15] C.-W. Wu, B.-E. Shie, V. S. Tseng and P. S. Yu "Mining top-k high utility itemsets", Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp.78 -86
[16] C.-W Wu, P. Fournier-Viger, P. S. Yu and V. S. Tseng "Efficient mining of a concise and lossless representation of high utility itemsets", Proc. IEEE Int. Conf. Data Mining, pp.824 -833