International Journal of Contemporary Research in Computer Science and Technology (IJCRCST) e-ISSN: 2395-5325
Volume 2, Issue 12 (December '2016)

UP-TREE STRUCTURE FOR POTENTIALLY HIGH UTILITY ITEMSETS

S.Dhivya,
Assistant Professor,
Department of Computer Science & Engineering,
K.L.N. College of Information Technology,
Pottapalayam, Sivagangai Dist, Tamil Nadu.

T.T.Mathangi,
Assistant Professor,
Department of Computer Science & Engineering,
K.L.N. College of Information Technology,
Pottapalayam, Sivagangai Dist, Tamil Nadu.

S.Jeniba,
Assistant Professor,
Department of Computer Science & Engineering,
K.L.N. College of Information Technology,
Pottapalayam, Sivagangai Dist, Tamil Nadu.

Abstract: Association-rule mining is commonly used to discover useful and meaningful patterns from a very large database. It only considers the occurrence frequencies of items to reveal the relationships among itemsets. In recent years, the problem of high utility pattern mining become one of the most important research area in data mining. High-utility mining was designed to solve the limitations of association-rule mining by considering both the quantity and profit measures. The existing high utility mining algorithm generates large number of candidate itemsets, which takes much time to find utility value of all candidate itemsets, especially for dense datasets. In this paper we have proposed UP-tree structure to reduce number of PHUIs(Potentially High Utility Itemsets) and to reduce execution time in Incremental High Utility Pattern Mining(IHUP). This algorithm has two strategies which is compared with other existing algorithms in various aspects. The experimental results show that the proposed algorithms reduce the number of candidates effectively.

Keywords: Incremental High Utility Pattern Mining, Dense Database, Potentially High Utility Itemsets.

I. INTRODUCTION

Extensive studies have been proposed for mining frequent patterns. 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 afterward proposed. In the framework of frequent itemset mining, the importance of items to users is not considered. Thus, the topic called weighted association rule mining was brought to attention. Although weighted association rule mining considers the importance of items, quantities in transactions are not taken into considerations yet. Thus, the issue of high utility itemset mining is raised and many studies have addressed this problem. Liu et al. proposed an algorithm named Two- Phase which is mainly composed of two mining phases. It still generate many HTWUIs. Although two-phase algorithm reduces search space by using TWDC(Transaction Weighted Downward Closure) property, it still generates too many candidates to obtain HTWUIs and requires multiple database scans. In phase I, to efficiently create HTWUIs and several times avoid scanning database, Ahmed discovered a IHUP tree-based algorithm. An IHUP was one of the effective algorithm to create utility itemsets.

IHUP algorithm has three stages:

1) Construction of IHUP-Tree
2) Generation of HTWUIs, and
3) Identification of high utility itemsets.

In stage 1, items are rearranged (lexicographic order) in a fixed order, support descending order or Transaction Weighted Utility descending order. after rearrangement transactions are feed into an IHUP-Tree. In stage 2, HTWUIs are created from the IHUP-Tree by applying FP-Growth. For the performance result of algorithm, the number of generated HTWUIs is a major issue. Due to that our aim is to reduce itemsets by several strategies. The number of created candidates can be highly minimized in phase I and high utility itemsets can be identified more efficiently in phase II by applying the proposed strategies.

II. RELATED WORK

In some applications such as transaction databases, though weighted association rule mining considers the importance of items, items' quantities in transactions are not taken into considerations yet. Thus, the issue of high utility itemset mining is raised. Liu et al. proposed an algorithm named Two-Phase which is mainly composed of two mining phases. But it generates too many candidates to obtain high transaction weighted utility itemsets and requires multiple database scans.

IJCRCST © 2016 | All Rights Reserved

The utility mining model was defined to solve the limitations of frequent pattern mining by allowing the user to measure the importance of an itemset by its utility value. With utility mining, several important decisions in business like maximizing revenue, minimizing marketing or inventory costs can be made and more important knowledge about itemsets contributing to the majority of the profit can be discovered. These techniques can be applied to many other areas which includes network traffic measurements, telecom call records etc. Traditional Association Rule Mining problem is a special case of utility mining, where the utility of each item is always 1 and the quantity is either 0 or 1. There is no efficient strategy to find all the high utility itemsets due to the non-existence of downward closure property (anti-monotone property) in the utility mining model.

The Two-Phase algorithm was developed to find high utility itemsets. In Phase I, transaction-weighted utilization was defined and transaction-weighted utilization mining model was proposed. This model maintains a Transaction-weighted Downward Closure Property. Thus, only the combinations of high transaction weighted utilization itemsets are added into the candidate set at each level during the level-wise search. Phase I may overestimate some low utility itemsets, but it never underestimates any itemsets. In phase II, only one extra database scan is performed to filter the overestimated item sets.

The challenge of utility mining is to effectively reduce the number of candidates. The Isolated Items Discarding Strategy (IIDS), have been proposed which can be applied to each level-wise utility mining method to further reduce the number of redundant candidates. In each pass, a utility mining method with IIDS scans a database that is smaller than the original by skipping isolated items to efficiently improve performance. It mainly focuses on the task of efficiently discovering all high utility itemsets.

IHUP algorithms

Another tree based algorithm was proposed, named IHUP to efficiently generate HTWUIs and avoid multiple time database scanning. It uses a tree based structure IHUP-Tree to maintain the information about itemsets and their utilities. It first generate IHUP tree and then generate HTWUIs from tree and at last performs mining on that itemset. To perform this operation it uses two database scan. In first scan it generates tree and during second scan it uses FP-Growth algorithm. However this algorithm also generates too many candidates. Hence also require more execution time. Hence we include a new UP-Tree structure and applies various strategies on IHUP algorithm to reduce HTWUIs(High Transaction Weighted Utility Itemsets).

III. PROBLEM DEFINITION

We first give some definitions and define the problem of utility mining, and then introduce related work in utility mining.

TID	Transaction	TU
T1	(A,2) (C,1) (D,9)	12
T2	(A,1) (C,10) (E,1) (G,5)	26
T3	(A,6) (B,2) (D,2) (E,1) (F,2)	36
T4	(B,1) (C,3) (D,1) (E,3)	32
T5	(B,4) (C,1) (E,4) (G,2)	12
T6	(A,2) (B,1) (C,1) (D,1) (H,1)	13

Table 1 : Database Example

Item	A B C D E F G H
Profit	2 5 2 1 5 3 2 1

Table 2: Profit table

A finite set of items I ={ i_1, i_2, i_3,…., i_m},each item i_p($m > p > 1$) has a unit profit pr(i_p). An itemset X is a set of k distinct items {i_1, i_2, …., i_k }, where ij I; $1 \leq j \leq k$. k is the length of X. An itemset with length k is called a kitemset. A transaction database D ={T_1, T_2,…., T_n } contains a set of transactions, and each transaction Td($1 \leq d \leq n$)has a unique identifier d, called TID. Each item i_p in transaction T_d is associated with a quantity q(i_p, T_d), that is, the purchased quantity of i_p in T_d.

Definition 1: Utility of an item ip in a transaction T_d is denoted as u(i_p, T_d) and defined as pr(i_p) × q(i_p , T_d).

Definition 2: Utility of an itemset X in T_d is denoted as u(X, T_d) and defined as u(i_p, T_d).

Definition 3: Utility of an itemset X in D is denoted as u(X) and defined as u(X, T_d).

Definition 4: An itemset is called a high utility itemset if 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 itemset.

Definition 5: Transaction utility of a transaction Td is denoted as TU(T_d) and defined as u(T_d, T_d).

Definition 6: Transaction-weighted utility of an itemset X is the sum of the transaction utilities of all the transactions containing X, which is denoted as TWU(X) and defined as TU(T_d).

Definition 7: An itemset X is called a high transaction weighted utility itemset (HTWUI) if TWU(X) is no less than min_util.

Property 1: (Transaction-weighted downward closure.). For any itemset X, if X is not a HTWUI, any superset of X is a low utility itemset.

IV. THE PROPOSED SYSTEM

UP-TREE : For the performance of mining with avoiding again and again scanning original database, we prefer UP-Tree compact tree structure. So, transactional information and high utility itemsets are maintained. To minimize the overestimated utilities stored in the nodes of global UP-Tree, two stages are used. In following sections, the elements of UP-Tree are first defined. Next, the two strategies are introduced. Finally, how to construct an UP-Tree with the two strategies is illustrated in detail.

The Elements in UP-Tree : In an 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 UPTree. 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. In following sections, two strategies for decreasing the overestimated utility of each item during the construction of a global UP-Tree are introduced.

Strategy DGU: Discarding Global Unpromising Items during Constructing a Global UP-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 itemsets if its TWU is less than the minimum utility threshold. Such an item is called an unpromising item.

Property 2: (Antimonotonicity of unpromising items). If i_u is an unpromising item, i_u and all its supersets are not high utility itemsets.

Corollary 1: Only the supersets of promising items are possible to be high utility itemsets. During the second scan of database, transactions are inserted into a UP-Tree. When a transaction is retrieved, the unpromising items should be removed from the transaction and their utilities should also be eliminated from the transaction's TU according to Property 2 and Corollary 1.

This concept forms our first strategy.

Strategy 1 DGU: Discarding global unpromising items and their actual utilities from transactions and transaction utilities of the database. New TU after pruning unpromising items is called reorganized transaction utility (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 itemsets are generated. Strategy DGU uses RTU to overestimate the utilities of itemsets instead of TWU. Since the utilities of unpromising items are excluded, RTU must be no larger than TWU. Therefore, the number of PHUIs with DGU must be no more than that of HTWUIs generated with TWU. DGU is quite effective especially when transactions contain lots of unpromising items, such as those in sparse data sets. Besides, DGU can be easily integrated into TWU based algorithms. Moreover, before constructing an UP-Tree, DGU can be performed repeatedly till all reorganized transactions contain no global unpromising item. By performing DGU for several times, the number of PHUIs will be reduced; however, it needs several database scans.

Strategy DGN: Decreasing Global Node Utilities during Constructing a Global UP-Tree. It is shown that 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. The search space can be divided into the following subspaces:

1. {B}'s conditional tree (abbreviated as {B}-Tree),
2. {A}-Tree without containing {B},
3. {D}-Tree without containing {B} and {A},
4. {C}-Tree without containing {B}, {A}, and {D},

and

5. {E}-Tree without containing {B}, {A}, {D}, and{C}.

It can be observed that in the subspace {A}-Tree, all paths are not related to {B} since the nodes {B} are below the nodes {A} in global IHUP-Tree. In other words, the items that are descendant nodes of the item i_m will not appear in $\{i_m\}$- Tree; only the items that are ancestor nodes of i_m will appear in {im}-Tree. From this viewpoint, our 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.

Strategy 2. DGN: Decreasing global node utilities for the nodes of global UP-Tree by actual utilities of descendant nodes during the construction of global UP-Tree. By

applying strategy DGN, the utilities of the nodes that are closer are further reduced. DGN is basically suitable for the long transactional databases. Means, the more transactional items, the more utilities can be discarded by DGN. Our traditional TWU mining model is not suitable for such databases .since the more items a transaction contains, the higher TWU is. In following sections, we describe the process of constructing a global UP-Tree with strategies DGU and DGN.

Making a Global UP-Tree by using DGU and DGN

The construction of a global UP-Tree by using two database scans. In the first scan, each transaction's TU is computed; at the same time, each 1-item's TWU is also collected. Then, we can get promising items and unpromising items. After getting all promising items, DGU is applied. The transactions are reorganized by pruning the unpromising items and sorting the remaining promising items in a fixed order. Lexicographic, support, or TWU order can be used. Then above rearrangement is called a reorganized transaction.

V.CONCLUSION

The key contribution of this work is to reduce number of PHUIs(Potentially High Utility Itemsets) and to reduce execution time. A UP Tree structure has been proposed to mine High Utility Itemsets with two strategies DGU and DGN in IHUP Algorithm to improve mining performance. This algorithm with two strategies is compared with other existing algorithms in various aspects. The experimental results show that the proposed algorithms reduce the number of candidates effectively.

VI. REFERENCES

[1].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*,2009.

[2].T.-P. Hong, C.-W. Lin, and Y.-L. Wu, "Incrementally Fast Updated Frequent Pattern Trees," Expert Systems with Applications, 2008.

[3]**.** Y.-C. Li, J.-S. Yeh, and C.-C. Chang, "Isolated Items Discarding Strategy for Discovering High Utility Itemsets," Data and Knowledge Eng., 2008.

[4].Y.-S. Lee and S.-J. Yen, "Incremental and Interactive Mining of Web Traversal Patterns," Information Sciences, 2008.

[5].S. Zhang, J. Zhang, and C. Zhang, "EDUA: An Efficient Algorithm for Dynamic Database Mining," Information Sciences, 2007.

[6].A. Erwin, R.P. Gopalan, and N.R. Achuthan, "CTU-Mine: An Efficient High Utility Itemset Mining Algorithm Using the Pattern Growth Approach," *Proc. Seventh IEEE Int'l Conf. Computer and Information Technology (CIT '07)*, 2007.

[7].J. Hu and A. Mojsilovic, "High Utility Pattern Mining: A Method for Discovery of High Utility Itemsets," *Pattern Recognition*, 2007.

[8].C. Lucchese, S. Orlando, and R. Perego, "Fast and Memory Efficient Mining of Frequent Closed Itemsets," IEEE Trans. Knowledge and Data Eng.,Jan. 2006.

[9].U. Yun and J.J. Leggett, "WFIM: Weighted Frequent Itemset Mining with a Weight Range and a Minimum Weight," *Proc.Fifth SIAM Int'l Conf. Data Mining (SDM '05),* 2005.

[10].Y. Liu, W.-K. Liao, and A. Choudhary, "A Two Phase Algorithm for Fast Discovery of High Utility of Itemsets," *Proc. Ninth Pacific Asia Conf.Knowledge Discovery and Data Mining (PAKDD '05)*, 2005.

[11].Y. Liu, W.-K. Liao, and A. Choudhary, "A Fast High Utility Itemsets Mining Algorithm," *Proc. First Int'l Conf. Utility-Based Data Mining*, 2005.

[12].J. Wang, J. Han, Y. Lu, and P. Tzvetkov, "TFP: An Efficient Algorithm for Mining Top-K Frequent Closed Itemsets," IEEE Trans. Knowledge and Data Eng., 2005.

[13].G. Grahne and J. Zhu, "Fast Algorithms for Frequent Itemset Mining Using FP-Trees," IEEE Trans. Knowledge and Data Eng.,2005.