An Improved Up-Growth Algorithm For Mining High Utility Itemsets From Transactional Databases

Ms. Jyoti Ashokkumar Aidale

Abstract- In the field of data mining it is very important to find interesting pattern from the transactional databases. To improving the efficiency of business applications, it is essential for producing itemsets that are frequently purchasing & association rules from a large amount of data sources. One of the fundamental mining methods is frequent pattern mining discovering useful pattern, and there is a need to find high utility itemsets to improve the performance of business management. In recent years, Utility mining becomes an important topic in the field of data mining. Finding itemsets with high utility from transactional databases. Even though we had a number of different algorithms, there are some problem with those existing mechanisms proposed in past. Those producing a large number of candidate itemsets, which minimizes the mining performance in terms of execution time and space requirement.

In this research I propose two algorithms, namely utility pattern growth (UP-Growth) and utility pattern Growth+ (UP-Growth+) for mining high utility itemsets with effective strategies for pruning candidate itemsets. UP-Tree structure is used to maintain the information of high utility itemsets. Experimental result shows that the performance of UP-Growth and UP-Growth+ outperforms state-of-art of other algorithms in terms of runtime and reduces the no.of candidates effectively especially, when there are long transactions in databases.

Keywords - Candidate Pruning, Data mining, Frequent pattern mining, High Utility Itemset, Utility mining.

1 INTRODUCTION

Data mining is the process of discovering useful information from large databases. Association rule mining is one of the most widely used techniques in data mining and knowledge discovery.one of the well-known and important technique in association rule mining is frequent pattern mining. In frequent pattern mining, only frequency of items is considered. But non frequent items are not considered which are also useful in a large profit. To address this limitation of association rule mining many types of mining were defined like weighted frequent pattern mining and Utility Mining. Utility mining is an important topic in the field of data mining. Mining high utility itemsets from databases means to find itemset with high profit. Utility of an itemset is the interestingness, an importance or profitability of an item to users.

Utility of an itemset in a transactional database consist of two aspects. First one is the importance of distinct item is called external utility. Second one is the importance of items in the transaction that is called internal utility. The transactional utility of an itemset is defined as the product of external utility and the internal utility. An itemset is called high utility if its utility is not less than a user specified minimum support threshold utility value; otherwise itemset is treated as low utility itemset. Finding high utility itemsets from databases is an important method which has a wide range of applications such as website click stream analysis, business promotion in chain hypermarkets, cross marketing in retail stores[4],[6] , online e-commerce management, and mobile commerce environment planning, and it also useful in finding important patterns in biomedical applications.

Finding useful patterns hidden in a database plays an important role in several data mining tasks, such as frequent pattern mining, weighted frequent pattern mining, and high utility pattern mining. Among them, frequent pattern mining is a fundamental method that has been applied to different kinds of databases. Nevertheless, relative importance of each item is not considered in frequent pattern mining. To address this problem, Cai et al proposed weighted association rule mining. In this framework, weights of items, such as profits of items in transaction databases are considered. With this concept, even if some items appear not frequently, they might still be found if they have high weights. However, in this framework, the quantities of items are not considered in weighted association rule mining. Therefore, it cannot satisfy the requirement of users who are interested in discovering of high profits.

There are no. of relevant algorithms have been proposed in recent years they incur problem of producing large no. of candidate itemsets for high utility itemsets and such large candidate itemset minimizes the mining performance in terms of execution time and space requirement. Mining high utility itemset is difficult when downward closure property does not hold. To address these issues, novel algorithms proposed with a compact data structure for efficiently finding high utility itemsets from

transactional databases. Some strategies are proposed with UP-Growth and UP-Growth+. UP-Tree data structure is used for maintaining information of High Utility itemsets. Experiments are performed on both real and synthetic data to compare the performance of the proposed algorithms with the state-of-the-art utility mining algorithms. From Experimental results we can conclude that UP-Growth and UP-Growth+ performs better than other algorithms in terms of execution time, when there is database contains lots of long transactions.

2 RELATED WORK

One of the fundamental algorithms for association rule is Apriori, that efficiently used for large databases .Pattern growth-based association rule mining algorithms , such as FP-Growth[5] were afterward proposed.FP-Growth achieves a better performance than Apriori-based algorithms. FP-growth finds frequent itemsets without generating any candidate itemset and scans database just twice. Then after in the framework of frequent mining Cai et al. first proposed the concept of weighted items and weighted association rules [2]. The weighted association rule does not have downward closure property that decreases mining performance. For that problem Tao et al. [3] proposed the concept of weighted downward closure property. By using transaction weight, weighted support can not only reflect the importance of an itemset but also maintain the downward closure property during the mining process. However weighted association rule mining considers importance of items but quantities of items are not considered. For that reason, high utility itemsets mining raised. Liu et al. proposed an algorithm named Two-Phase [9], which is mainly composed of two mining phases, in the phase I, HTWUIs is collected. In phase II, HTWUIs that are high utility itemsets are identified with an additional database scans. But still generates too many candidates and too many database scan Li et al. [8] proposed Isolated items discarding strategy (IIDS) to remove the number of candidates. During level-wise search, the isolated items are pruned the number of Candidate itemsets. This algorithm still scans database for several times. Ahmed et al. [1] proposed a tree-based algorithm, named IHUP. A tree based structure called IHUP-Tree is used to maintain the information about itemsets and their utilities. It uses a candidate generation-and-test scheme to find the high utility itemset which increases time complexity.

The two novel algorithms named, utility pattern growth (UP- Growth) and Utility pattern Growth +(UP-Growth+) and a compact tree structure called a utility pattern tree (UP-Tree) for discovering high utility itemsets and maintaining important information related to utility patterns within databases were proposed by Tseng et al.[7]. Several strategies are proposed for providing the mining processes of UP-Growth and UP-Growth+ by maintaining only important information on the UP - Tree. By these strategies, overestimated utilities of candidates can be well reduced by discarding utilities of the items that are unpromised. The Strategies were proposed cannot only decrease the overestimated utilities of the potential high utility itemsets but also effectively reduce the number of candidates.

3 PROBLEM DEFINITION

In this section, we define the preliminary work of utility mining. Given a finite set of items I = {i1, i2… im}. Each item ip (1 ≤ p ≤ m) has a unit profit p (ip).An itemset X is a set of k distinct items {i1, i2… ik}, where ij € I, 1≤ j ≤ k, and k is the length of X. An itemset with length k is called k-itemset. A 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 the transaction Td is associated with a quantity q (ip, Td), that is, the purchased number of ip in Td.

Definition 1: The utility of an item ip in the transaction Td is denoted as u(ip,Td) and defined as

$$p(ip) \times q(ip, Td)$$

Definition 2: The utility of an itemset X in Td is denoted as u(X, Td) and defined as

$$\Sigma_{ip \in X \wedge X \subseteq Td} \; u(ip, Td)$$

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

$$\Sigma_{X \subseteq Td \wedge Td \in D} \; u(X, T_d)$$

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

$$u(Td, Td)$$

Definition 5: 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

$$\Sigma_{X \subseteq Td \wedge Td \in D} \; TU(Td)$$

Definition 6: An itemset X is called a high transaction-weighted utility itemset (abbreviated as HTWUI) if TWU(X) is no less than min_util.

Property1(TWDC): Transaction weighted downward closure property states that any itemset Z, if Z is not HTWUI then any superset of Z is a lower utility itemset.

4 PROPOSED METHOD

UP-Growth and UP-Growth+ this gives a new way in analyzing the item sets. The goal of utility mining is to find all the high utility itemsets whose utility values are greater than a user specified threshold in a transaction database.

The itemsets that are both high frequent and high utility can be obtained using this method. Among all The State- of- art algorithms for high utility itemsets, UP-Growth and UP-growth+ performed well.

4.1 Proposed Data Structure Up-Tree

In UP-Tree each node N includes N.name, N.count, N.nu, N.parent, N.hlink and a set of child nodes. The details are introduced as follows. N.name is the item name of the node. N.count is the support count of the node. N.nu is called the node utility that is an estimate utility value of the node. N.parent records the parent node of the node. N.hlink is a node link which points to a node whose item name is the same as N.name. Header table is employed to facilitate the traversal of UP-Tree. In the header table, each entry is composed of an item name, an estimate utility value, and a link. The link points to the last occurrence of the node that has the same item as the entry in the UP-Tree.

Strategy 1: Discarding global unpromising items (DGU): The unpromising items and their utilities are removed from the transaction utilities during the construction of a global UP-Tree.

Strategy 2: Decreasing Global Node Utility (DGN): To remove the utilities of descendent nodes from their node utilities in global UP-Tree.

4.2 The Proposed Mining Method: Up-Growth

In this section, we describe how PHUIs are generated from global UP-Tree by using two strategies DLU (Discarding local unpromising items) and DLN (Decreasing local node utilities) in UP-Growth algorithm.

DGU and DGN strategies can effectively reduce the number of candidates in phase 1, and applied during construction of global UP-Tree. They are not applied during construction of local UP-Tree. Because we cannot know the utility values of unpromising items in conditional pattern base (CPB), we propose naïve approach to maintain the utilities of items in CPB, but it requires lots of memory usage. For that, we maintain minimum item utility table which contain minimum item utility for all unpromising items instead of exact utility values in CPB.

Strategy 3: Discarding local unpromising items (DLU) while constructing a local UP-Tree.

Strategy 4: Discarding local node utilities (DLN).The minimum item utilities of descendant nodes for a node are decreased during the construction of a local UP-Tree.

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

UP-Growth algorithm proposed with these two strategies for generating high utility itemsets.

4.3 An Improved Proposed Method: Up-Growth + Algorithm

UP-Growth+ algorithm used with improved strategies named, DNU and DNN.Overestimated utilities are reduced efficiently by this method. In UP-Growth+, minimal node utilities in each path are used to make the estimated removing values closer to real utility values of the removed items in the database.

During construction of Global UP-Tree, we add an element N.mnu to each node of UP-Tree. N.mnu is the minimum node utility of N. when N is retrieved it keeps track of minimal value of N.name's utility in different transactions. If N.mnu is larger than u(N.name,T.current) then N.mnu set to u(N.name,T. current).

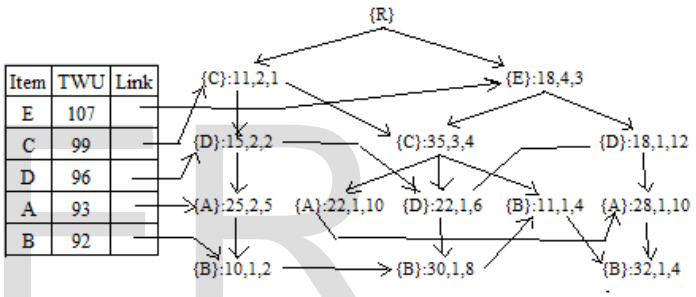

Fig1. Shows global UP-tree with the minimal node utility.

TABLE 1

{B}-CPB AFTER APPLYING DGU, DGN, AND DLU

Retrieved path: path Utility	Reorganized path: Path utility(after DLU	Support count
<ADC>:10	<DC>:5	1
<DCE>:30	<EDC>:30	1
<CE>:11	<EC>:11	1
<ADE>:32	<ED>:27	1

Strategy 5: (DNU) Discarding local unpromising items and their estimated Node Utilities from the paths and path utilities of conditional pattern base.

Strategy 6: (DNN) Decreasing local Node utilities for the nodes of local UP-Tree by estimated utilities of descendant Nodes.

TABLE 2

{B}-CPB BY APPLYING DGU, DGN, AND DNU

Retrieved path: Path Utility	Reorganized path: Path utility(after	Support count

	DNU)	
<A(5D(2)C(1)>:10)	<D(2)C(1)>:5	1
<D(6)C(4)E(3)>:30	<E(3)D(6)C(4)>:30	1
<C(4)E(3)>:11	<E(3)C(4)>:11	1
<A(10)D(12)E(3)>:32	<E(3)D(12)>:22	1

There are four retrieved paths, shown in above Table2 and the number in bracket denotes minimal node utility. After scanning {B}-CPB path utility of each local item is calculated {A}: 42, {C}: 51, {D}: 72 and {E}: 73. According to DNU, local unpromising item {A} and its minimal node utility are discarded from path utilities. And the items in each path are reorganized by descending order of path utility of local items. It is shown in Table2. As compared Table1 and Table 2 reorganized path reduced further and node utilities are further reduced.

Subroutine: UP-Growth.
Input: Transaction database D, user specified threshold.
Output: high utility itemsets.
Begin

1. Scan database of transactions Td ϵ D
2. Determine transaction utility of Td in D and TWU of itemset (X)
3. Compute min_sup (MTWU*)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 the 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 Y= X U ai
12. Estimate utility of Y is set as ai's utility value in H
13. Put local promising items in Y-CPB Into H.
14. Apply strategy DLU to reduce path utilities.
15. Apply strategy DLN and insert path into Td.
16. If Td ≠ null then call for loop
    End for
    End

5 EFFICIENTLY IDENTIFY HIGH UTILITY ITEMSETS

To identify high utility itemsets from Potential high utility itemsets (PHUIs) by scanning original database once in phase 2. There are many existing methods that they incur the problems, scanning original databases repeatedly so that it was time consuming, and a large number of high utility itemsets generated. Our proposed method overestimated utility of PHUIs is reduced. Smaller numbers of PHUIs are generated that is much smaller than HTWUIs in phase 2.

6 EXPERIMENTAL RESULT

Performance of an algorithm is in this section. In the experiment, both Real and Synthetic datasets are used. Synthetic data sets are generated from data generator. From FIMI repository, Real world datasets such as Accident and chess are obtained. Chain-store is obtained from NU-MineBench 2.0; Foodmart obtained from Microsoft foodmart 2000 database. Unit profits for items in utility tables are generated between 1 and 1,000 by using a log-normal distribution and quantities of items are generated randomly between 1 and 10.Performance of algorithm obtained by comparing different method. The algorithms, we design two methods UPT&UPG and UPT&UPG+ that are composed of the proposed methods UP-Tree and UP-Growth (with DGU, DGN, DLU, and DLN) and the proposed methods UP-Tree and UP-Growth+(with DGU, DGN, DNU, and DNN), respectively.

6.1 Performance Comparison on Different Data Sets

Dense data set Chess and sparse data sets Chain-store and Foodmart are used for performance comparison of proposed methods. When we compare both dataset, we can observe that the performance of proposed methods outperforms that of previous methods. The runtime of IHUPT&FPG is the worst, followed by UPT&FPG, UPT&UPG, and UPT&UPG+ is the best. Run time depends on candidate generation.

6.2 Scalability of proposed method

When Experiment performed on synthetic dataset T10.F6.|I|1,000.|D|,xk. We can observe that the performance of UPG&UPG+ is best than other algorithms. It generates least PHUIs in phase 1 as database size increases. Memory Usage of the Proposed Methods Memory usage increases as decreasing min_util and as increases database size. Performance of UPG&UPG+ is best among another method.

TABLE 3:
CHARACTERISTICS OF REAL DATA SETS

Dataset	|D|	T	|I|	Type
Accident	3,40100	33.8	460	Dense
Chess	3450	37.0	75	Dense
Chain-store	1,12,340	7.2	46,123	Sparse
Foodmart	4,140	4.2	1,255	Sparse

7 CONCLUSION

In this paper, two efficient algorithms such as UP-Growth and UP-Growth+ are proposed for the mining high utility itemset from transactional databases. A compact data structure is proposed for maintaining information of high utility itemsets and with only two databases scan PHUIs are generated from UP-Tree. The strategies used with UP-Growth and UP-Growth+ algorithm are for reducing overestimated utilities and improve the mining performance in terms of space and execution time. Experimental results on Real and Synthetic datasets show that strategies improved performance by reducing search space and number of candidates. Both proposed algorithms performed best comparison to other algorithms. The proposed algorithms, especially UP-Growth+ outperforms the state-of-the-art algorithms substantially especially when databases contain lots of long transactions.

REFERENCE

[1] C.F.Ahmed,S.K.Tanbeer,B.S.Jeong and Y.K .Lee,2009 "Efficient Tree structure for High Utility Pattern Mining in Incremental Databases", IEEE Transaction.1708-1721.

[2] C.H. Cai, A.W.C. Fu, C.H. Cheng, and W.W. Kwong, 1998 "Mining Association Rules with Weighted Items," Proc. Int'l Database Eng. and Applications Symp, pp. 68-77.

[3] F. Tao, F. Murtagh, and M. Farid, 2003 "Weighted Association Rule Mining Using Weighted Support and Significance Framework,"Proc. ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD '03), pp. 661-666.

[4] H. Yao, H.J. Hamilton, and L. Geng, 2006, "A Unified Framework for Utility-Based Measures for Mining Itemsets," Proc. ACM SIGKDD Second Workshop Utility-Based Data Mining, pp. 28-37.

[5] J.Han, J.Pei and Y.Yin, 2000"Mining Frequent pattern without candidate generation", Proc.International Database and Engg. 68-77

[6] S.J. Yen and Y.S. Lee, 2007, "Mining High Utility Quantitative Association Rules." Proc. Ninth Int'l Conf. Data Warehousing and Knowledge Discovery (DaWaK), pp. 283-292.

[7] V.S.Tseng, C.W.Wu, B.E.Shie and P.S.Yu,2010,"An Efficient Algorithm for mining high Utility Itemsets Mining", Proc. of ACM-KDD, Washington, DC, USA, 253-262

[8] Y.-C. Li, J.-S. Yeh, and C.-C. Chang, 2008, "Isolated Items Discarding Strategy for Discovering High Utility Itemsets," Data and Knowledge Eng., vol. 64, pp. 198-217.

[9] Y. Liu, W. Liao, and A. Choudhary, 2005 "A Fast High Utility Itemsets Mining Algorithm," Proc. Utility-Based Data Mining Workshop.