An Efficient Discovery of High Utility Item Sets from Large Database

Santhamani.V
PPG Institute of Technology
Coimbatore, India

Premkumar.M
Department of CSE
PPG Institute of Technology
Coimbatore, India

Gayathri.A
PPG Institute of Technology
Coimbatore, India

Gokulavani.M
PPG Institute of Technology
Coimbatore, India

Abstract - Identifying frequent items from database and treating each item in a database as equal. However, items are actually differs in many aspects like, profit in real application, such as retail marketing. The difference between items makes a strong impact on the decision making applications, where the values of each items are considered as utilities. Utility mining focuses on identifying the itemsets with high utility like profit, aesthetic value. High utility itemsets mining extends frequent pattern mining to discover itemsets in a large database with utility values above a given threshold. Here we use two algorithms UP-Growth and FP-Growth for mining high utility itemsets and frequent users with a set of effective strategies. The information of high utility itemsets is maintained in a UP-Tree. Candidate itemsets are generated efficiently. Customer Relationship Management (CRM) is incorporated into the system by tracking the customers who are frequent buyers of the different kinds of item sets.

Keywords: Candidate itemsets; Frequent itemset; High utility itemset; Utility mining; data mining.

1. INTRODUCTION

Data mining is the process of showing nontrivial, previously unknown and potentially useful information from large databases. It enables the companies to focus on important information in data warehouses. It can be implemented rapidly on the existing software and hardware platforms and also enhances the value of the existing information resources. Discovering useful patterns hidden in a database plays an essential role in several data mining tasks, such as frequent pattern mining, and high utility pattern mining.

Association rules mining (ARM) is one of the most widely used techniques in data mining and has tremendous applications like business, science etc. Make the decisions about marketing activities such as, e.g., promotional pricing or product placements.

Relative importance of each item in frequent pattern mining is not considered. To address this problem, weighted association rule mining was proposed. In this, weights of items, such as unit profits of items in transaction databases, are considered. In this view, utility mining emerges as an important topic in data mining field. Mining high utility itemsets from the large databases refers to finding the itemsets with high profits. The meaning of itemset utility is interestingness, importance, or profitability of an item to user. Utility of items in a transaction database consists of two aspects:1) the importance of distinct items, and 2) the importance of items in transactions.

A high utility itemset is an itemset if its utility is no less than a user-specified utility threshold; otherwise, it is a low-utility itemset. Mining high utility itemsets from the large databases is not an easy task since downward closure property in frequent itemset mining does not hold. In different way, pruning search space for high utility itemset mining is difficult because a superset of a low-utility itemset may be a high utility itemset. Existing methods often generate a huge set of PHUIs and their mining performance is degraded consequently. The situation becomes worse when database_contain many long transactions or low threshold value are set. The huge amount of PHUIs forms a challenging problem for mining performance since the more PHUIs the algorithm generates, the time consuming process. Major contributions of this work are summarized as follows:

1. Two algorithms, namely Utility Pattern growth (UP-Growth) and FP-Growth, and a compact tree structure, called utility pattern tree (UP-Tree), for discovering high utility itemsets and maintaining important information related to utility patterns within databases are proposed. Efficiently High-utility itemsets can be generated from UP-Tree by two scans of original databases.
2. Several strategies are proposed for facilitating the mining processes of UP-Growth. By maintaining only essential information in UP-Tree, overestimated utilities of candidates can be well reduced by discarding utilities of the items that cannot be high utility.
3. UP-Growth and FP-Growth outperform other algorithms substantially in terms of execution time, especially when database contain lots of long transactions or low minimum utility thresholds are set.

1.1 Preliminary

Given a finite set of items L = {l_1,l_2,….l_n}, each item l_p (1≤ p ≤n)has a unit profit pr(l_p). An itemset X is a set of k distinct items {l_1, l_2,….l_k}, where l_j £ L,1≤j≤k.k is the length of X. An itemset with length k is called a k-itemset. A transaction database D = {T_1, T_2,…,T_m} contains a set of transactions, and each transaction Td(1 ≤ d ≤m) has a unique identifier d, called TId. Each item l_p in transaction Td is associated with a quantity q (l_p,T_d), that is, the purchased quantity of l_p in T_d.

TABLE 1

An example Database

TID	Transaction	TU
T1	(A,2)(B,3)	12
T2	(B,2)(C,2)(D,1)	15
T3	(C,1)(D,2)	7
T4	(A,1)(B,1)(C,3)	20

TABLE 2

Profit Table

Item	A	B	C	D
Profit	3	2	5	1

Definition 1

An itemset is no less than user- specified minimum utility threshold which is called high utility itemset, Otherwise it low-utility itemset.

For example, in Tables 1 and 2,

u({A},T1) = 3×2=6;

u({AB},T_1) = u({A},T_1)+ u({B},T_1)
=6+6 = 12;

u({AB})= u({AB},T1)+u({AB},T4)
=12+5 =17;

If min_util is set to 17, {AB} is a high utility itemset.

Definition 2

Transaction-weighted utility(TWU) of an itemset X is the sum of the transaction utilities of all the transaction containing X,wich is denoted as TWU(X).

Property 1: (Transaction-weighted downward closure)

Any subset of a high transaction-weighted utilization itemset must also be high in transaction-weighted utilization it is called transaction weighted downward closure (TWDC).

Downward closure property can be maintained in utility mining by applying the transaction weighted utility. For example, TU(T1) = u({ABC},T1)=17; TWU ({A})=

TU(T1)+TU(T4)=17+28= 45; If min_util is set to 30, {A} is a HTWUI.

The challenge of utility mining is restricting the size of the candidate set and simplifying the computation for calculating the utility.

Problem statement- The problem is producing a large number of candidate itemsets for high utility itemsets. Apriori based algorithms prune candidate itemsets, however algorithms need to test all candidates. Moreover, they must repeatedly scan a large amount of the original database in order to check if a candidate item is frequent or not. It is inefficient and ineffective.

2. RELATED WORK

One of the well-known algorithms for mining association rules is Apriori [1], which is used for efficiently mining association rules from large database. Pattern growth-based association rule mining algorithms [3], [5] such as FP-Growth [3] were afterward proposed. It achieves a better performance than Apriori-based algorithms since it finds frequent itemsets without generating any candidate itemset and scans database just twice.

In a frequent itemset mining, the importance of items to users is not considered. So, the topic called weighted association rule mining was introduced. The weighted association rule mining (WARM) considers the importance of items, in some applications such as transaction databases, items' quantities in transactions are not_yet consider. So, the issue of high utility itemset mining is raised. And many studies [2], [4], [6], [7], [8] have addressed this problem.

Two phase algorithm has been proposed [4] which is composed of two mining phases. In phase 1, an Apriori-based level wise method is used and the complete set of HTWUIs is collected. In phase 2, an additional database scan is computed for identify HTWUIs. The two phase algorithm also reduces search space by using TWDC property, but it still produces too many candidate items to obtain HTWUIs and requires multiple database scans. To overcome this, an isolated items discarding strategy (IIDS) was introduced to reduce the number of candidate items. This algorithm uses a candidate generation-and-test scheme for finding high utility items but it still scans database for several times.

To avoid scanning database too many times and to generate HTWUIs, a tree based algorithm has been proposed, namely IHUP. In IHUP-Tree the information about itemsets and their utilities are maintained. This algorithm has three steps: 1) Create IHUP-Tree, consists of an item name, TWU value and support count, 2) HTWUIs are generated with the help of FP-Growth, 3) The original database has been scanned once, in which High utility itemsets are identified. From the above steps, there are many HTWUIs, thus the performance of an algorithm became a critical issue.

Hence the overestimated utilities of itemsets have to be reduced by applying proposed method.

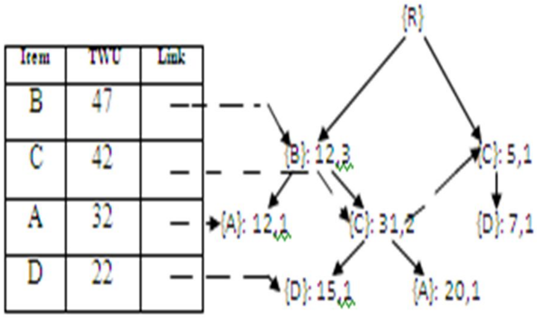
3. PROPOSED METHODS

The framework of proposed methods consists of four steps: 1) Scan the database twice to construct a UP-Tree; 2) Generate PHUIs (Potential High Utility Itemsets) from UP-Tree by using UP-Growth; 3) Identify actual high utility items from the set of PHUIs; 4) Identify frequent users by using FP-Growth and incorporating CRM;

3.1 Data structure: UP-Tree

To improve the mining performance and avoid repeated scanning of original database, we use a UP-Tree structure. It is used to maintain the information of transactions and high utility itemsets.

In a UP-Tree each node N consists of N.name (node's item name), N.count node's support count), N.nu (node's node utility, i.e., overestimated utility of the node), N.parent (records the parent node of N), N.hlink (node link points to a node which name is the same as N.name) and a set of child nodes.



Item	TWU	Link
B	47	—
C	42	—
A	32	—
D	22	—

Figure 1. UP-Tree

3.2 Mining Method: UP-Growth

After constructing a UP-Tree, a basic method for generating PHUIs is to mine UP-Tree by UP-Growth by pushing two more strategies into the framework of FP-Growth. From the strategies, overestimated utilities of itemsets can be decreased and thus the number of PHUIs can be further reduced.

3.3 Identify High Utility Itemsets

After finding all PHUIs, the third step is to identify high utility itemsets and their utilities from the set of PHUIs by scanning original database once. This step is called phase II. However, in previous studies, two problems in this phase occur: 1) number of HTWUIs is too large; and (2) scanning original database is very time consuming. In our work, overestimated utilities of PHUIs are smaller than or equal to TWUs of HTWUIs since they are reduced by the proposed strategies. So, the number of PHUIs is much smaller than that of HTWUIs. Hence, in phase II, our method is much efficient than the previous methods._Although our methods generate fewer candidates in phase I, scanning original database is still time consuming since the original database is large and it contains lots of unpromising items. In our framework, high utility itemsets can be identified by scanning reorganized transactions. Since there is no unpromising item in the reorganized transactions, I/O cost and execution time for phase II can be further minimized. This technique works well especially when the original database contains lots of unpromising items.

3.4 Identify frequent users and CRM

The item sets that are both high frequent and high utility can be obtained using FP-Growth. From the basic framework of this algorithm the different kinds of item sets namely high utility high frequent, high utility low frequent, low utility high frequent and low utility low frequent are generated. Then Customer Relationship Management (CRM) is incorporated into the system by tracking the customers who are frequent buyers of the different kinds of item sets.

4. CONCLUSION

Generate potentially high utility itemsets using utility pattern with two database scans. This combines with the frequent pattern to provide better performance and gives best solution for time consumption. Apriori algorithm requires multiple time databases scanning. To find long patterns it may need too many database scanning that is quite time consuming. Meantime, while processing data sets that contain long patterns, it generates too many candidates and subsequences of frequent patterns. To solve these problems we are using high utility pattern, which avoids the costly candidate generation and requires only two times database scanning. The first pass finds all frequent items, and the second pass constructs compact data structure using the high utility items which are used for storing compressed, crucial information about high utility patterns.

Customer Relationship Management (CRM) is incorporated into the system by tracking the customers who are frequent buyers of the different kinds of item sets.

5. REFERENCES

[1] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules,” Proc. 20th Int’l Conf. Very Large Data Bases(VLDB), pp. 487-499, 1994.

[2] A. Erwin, R.P. Gopalan, and N.R. Achuthan, “Efficient Mining of High Utility Itemsets from Large Data Sets,” Proc. 12th Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining (PAKDD), pp. 554-561, 2008.

[3] J. Han, J. Pei, and Y. Yin, “Mining Frequent Patterns without Candidate Generation,” Proc. ACM-SIGMOD Int’l Conf. Management of Data, pp. 1-12, 2000.

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

[5] J. Pei, J. Han, H. Lu, S. Nishio, S. Tang, and D. Yang, “H-Mine: Fast and Space-Preserving Frequent Pattern Mining in Large Databases,” IIE Trans. Inst. of Industrial Engineers, vol. 39, no. 6,pp. 593-605, June 2007.

[6] B.-E. Shie, V.S. Tseng, and P.S. Yu, “Online Mining of Temporal Maximal Utility Itemsets from Data Streams,” Proc. 25th Ann. ACM Symp. Applied Computing, Mar. 2010.

[7] V.S. Tseng, C.J. Chu, and T. Liang, “Efficient Mining of Temporal High Utility Itemsets from Data Streams,” Proc. ACM KDD Workshop Utility-Based Data Mining Workshop (UBDM ’06), Aug. 2006.

[8] V.S. Tseng, C.-W. Wu, B.-E. Shie, and P.S. Yu, “UP-Growth: An Efficient lgorithm for High Utility Itemsets Mining,” Proc. 16th ACM SIGKDD Conf. Knowledge iscovery and Data Mining (KDD ’10), pp. 253-262, 2010.

[9] 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. Knowledge and Data Eng., vol. 21, no. 12,pp. 1708-1721,Dec.2009.

[10] C. Creighton and S. Hanash, “Mining Gene Expression Databases for Association Rules,” Bioinformatics, vol. 19,no.1,pp.79-86,2003.

[11] M.Y. Eltabakh, M. Ouzzani, M.A. Khalil, W.G. Aref, and A.K.Elmagarmid, “Incremental Mining for Frequent Patterns in Evolving Time Series Databases,” Technical Report CSD TR#08-02, Purdue Univ., 2008.

[12] C.H. Lin, D.Y. Chiu, Y.H. Wu, and A.L.P. Chen, “Mining Frequent Itemsets from Data Streams with a Time-Sensitive Sliding Window,” Proc. SIAM Int’l Conf. Data Mining (SDM ’05), 2005.