Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 6, June 2015, pg.695 – 700

RESEARCH ARTICLE

Efficient Approach for Mining of High Utility Itemsets from Transactional Database

Pranoti Meshram
Department of C.S.E, Nagpur university
pnmeshram301@gmail.com

Prof. Vikrant Chole
Department of C.S.E, Nagpur university
vikrantchole@gmail.com

Abstract— Mining high utility itemsets from a transactional database refers to the discovery of itemsets with high utility like profits. Although a number of relevant approaches have been proposed in recent years, but they incur the problem of producing a large number of candidate itemsets for high utility itemsets. Such a large number of candidate itemsets degrades the mining performance in terms of execution time and space requirement. The situation may become worse when the database contains lots of long transactions or long high utility itemsets. To overcome this all limitation In this paper we proposed two algorithm , namely UP Growth and UP Growth plus algorithm for mining high utility itemsets with effective set of pruning strategy.The Experimental outcomes shows that the proposed algorithm , particularly utility pattern Growth plus , required less execution time and reduced memory usage when databases include lots of the high transactions.

Keywords—: Data mining,Candidate itemsets , high utility itemset, utility mining.

I. INTRODUCTION

A. Data Mining

Data mining, the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Knowledge Discovery in Databases (KDD) is the non-trivial process of identifying valid, previously unknown and potentially useful patterns in data. These patterns are used to make predictions or classifications about new data, explain existing data, summarize the contents of a large database to support decision making and provide graphical data visualization to aid humans in discovering deeper patterns. Discovering useful patterns hidden in a database plays an essential 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 research topic that has been applied to different kinds of databases, such as transactional databases, streaming databases, and time series databases, and various application domains, such as bioinformatics, Web click-stream analysis, and mobile environments.

The basic goal of frequent itemset mining is to identify all frequent itemsets. In past to find this frequent itemsets the generations of association rules and Apriori algorithm was used, once the frequent itemsets are identified and producing the itemsets with candidate and without candidates. But it is not producing the customer requirement like profit, sales in particular item. The unit profits and purchased quantities of items are not considered in the framework of mining frequent itemset. Hence, it cannot satisfy the requirement of the user who is interested in discovering the itemsets with high sales profits .Thus Mining high utility itemsets from databases refers to finding the itemsets with high profits .

B. Utility Mining

The limitations of frequent itemset mining motivated researchers to conceive a utility based mining approach, which allows a user to conveniently express his or her perspectives concerning the usefulness of itemsets as utility values and then find itemsets with high utility values higher than a threshold. In utility based mining the term utility refers to the quantitative representation of user preference i.e. according to an itemsets utility value is the measurement of the importance of that itemset in the user's perspective. The traditional ARM approaches consider the utility of the items by its presence in the transaction set. The frequency of itemset is not sufficient to reflect the actual utility of an itemset. For example, the sales manager may not be interested in frequent itemsets that do not generate significant profit. Recently, one of the most challenging data mining tasks is the mining of high utility itemsets efficiently.

In view of this, utility mining emerges as an important topic in data mining for discovering the itemsets with high utility like profits. Identification of the itemsets with high utilities is called as Utility Mining. The utility can be measured in terms of cost, quantity, profit and user preference .for this Utility mining model was proposed to define the utility of itemset. In this model by considering u(X) as a utility of an itemset X, which is the sum of the all utilities of itemset X in all the transactions containing X. then an itemset X is called a high utility items if its utility greater or equal to user- defined minimum utility threshresold.

II. **LITERATURE SURVEY**

A brief overview of various algorithms, Mining Frequent pattern defined in different research papers have been given in this section which is as follows:

A. Fast Algorithms for Mining Association Rules

R. Agrawal et al in [3] proposed Apriori algorithm, it is used to obtain frequent itemsets from large database. In Apriori algorithm there are two main steps involve to find out all larger itemsets from the database . In the first Process step simply counts item occurrences to further determine the large one itemsets. for this it first generates the candidate sequences and then it chooses the large sequences from the candidate ones. Next, the database scan is performed to count the support of candidates itemsets. Then second process step was performed which involves generating association rules from frequent itemsets. After identifying the large itemsets, only those itemsets are allowed which have the support greater than the minimum support allowed. but disadvantage of using Apriori Algorithm is that it generates lot of candidate item sets and scans database every time and when a new transaction is added to the database then it should rescan the entire database again.

B. Mining Frequent Pattern without Candidate generation.

Mining frequent patterns in transaction and many other kinds of databases has been popularly important research in data mining . The previous studies adopt an Apriori-like candidate set generation-and-test approach. However, candidate set generation is still costly, especially when there exist long patterns. for this J. Han et al in [4] proposed a novel method of frequent pattern tree (FP-tree) structure, an extended prefix tree structure for storing crucial information about frequent patterns into compressed structure and develop an efficient FP-tree based mining method is Frequent pattern tree structure. Pattern fragment growth mines the complete set of frequent patterns using the FP-growth. It constructs a highly compact FP-tree, which is usually substantially smaller than the original database, by which costly database scans are saved in the subsequent mining processes. It applies a pattern growth method which avoids costly candidate generation. but FP-Growth Consumes more memory and performs badly with long pattern data sets. Thus it is not able to find high utility itemsets.

C. Mining Association Rules with Weighted Items

W. Wang et al in [5] proposed weighted association rule. This method extends the traditional association rule problem by allowing a weight to be associated with each item in a transaction, to reflect intensity of the item within the transaction. This provides us in turn with an opportunity to associate a weight parameter with each item in the resulting association rule. We call it weighted association rule (WAR). In WAR, we use a twofold approach. First it generates frequent itemsets. In second for each frequent itemset the WAR finds that meet the support, confidence. However, the weighted association rules does not hold downward closure property, mining performance cannot be improved.

D. Two Phase Algorithm

To address the above problem Liu et al. proposed [6] an algorithm named Two-Phase algorithm to efficiently prune down the number of candidates and can precisely obtain the complete set of high utility itemsets. In the first phase, a model that applies the transaction-weighted downward closure property (TWDC) on the search space to expedite the identification of candidates. In the second phase, one extra database scan is performed to identify the high utility itemsets. It performs very efficiently in terms of speed and memory cost. Although two-phase algorithm reduces search space by using TWDC property but it still generates too many candidates to obtain HTWUIs and requires multiple database scans.

E. Isolated Items Discarding Strategy for Discovering High Utility Itemsets

Traditional methods of association rule mining consider the appearance of an item in a transaction, whether or not it is purchased, as a binary variable. However, customers may purchase more than one of the same item, and the unit cost may vary among items. Utility mining, a generalized form of the share mining model, attempts to overcome this problem. Since the Apriori pruning strategy cannot identify high utility itemsets, developing an efficient algorithm is crucial for utility mining. To overcome this problem, Li et al. [7] proposed an isolated items discarding strategy (IIDS) to reduce the number of candidates. By pruning isolated items during level-wise search, the number of candidate itemsets for HTWUIs in phase one can be reduced. However, this algorithm still scans database for several times and uses a candidate generation-and-test scheme to find high utility itemsets and thus cannot improved performance.

F. Efficient Tree Structures for High Utility Pattern Mining in Incremental Databases

Ahmed et al. [8] 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. Although IHUP achieves a better performance than IIDS and Two-Phase, it still produces too many HTWUIs in phase one. since the overestimated utility calculated by TWU is too large. Such a large number of HTWUIs will degrade the mining performance in phase one substantially in terms of execution time and memory consumption. Moreover, the number of HTWUIs in phase one also affects the performance of phase second due to more execution time required for identifying high utility itemsets .

III. **PROBLEM DEFINITION**

We have studied some proposed algorithms in related work. But all these algorithms incurred the problem of producing a large number of candidate itemsets. Such a large number of candidate itemsets degrades the mining performance in terms of execution time and space. And also problem occurred due to multiple database scan and thus higher processing time is required to finding high utility itemsets . To overcome this limitation in proposed system two novel algorithm used for mine high utility itemsets from transactional database.

IV. **PROPOSED SYSTEM**

The system architecture of proposed system is shown as below:

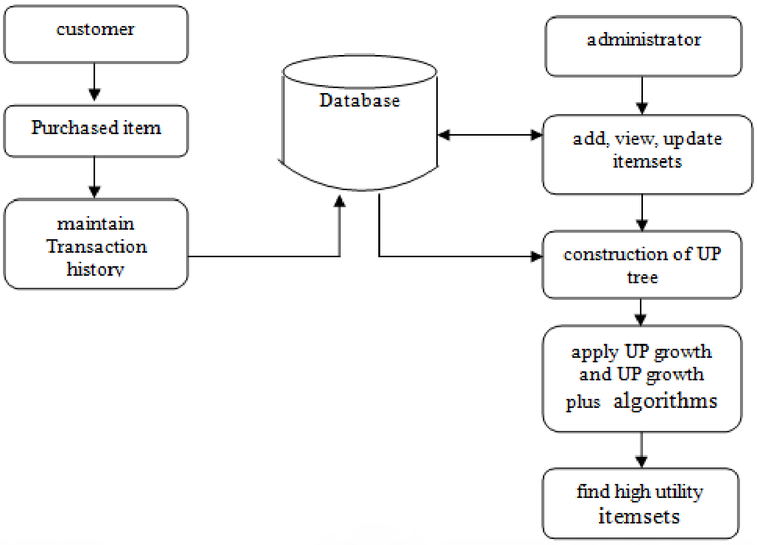

customer
Purchased item
maintain Transaction history
Database
administrator
add, view, update itemsets
construction of UP tree
apply UP growth and UP growth plus algorithms
find high utility itemsets

fig 1: System Architecture

1) Customer can purchase the items. All the purchased items history are stored in the transaction database.

2) Administrator : The administrator maintain database of the transactions made by customers. In the daily market basis, each day a new product is released, so that the administrator would add the product, update the new product view the stock details.

3) Construction of UP-Tree

1. First scan:-

Initially Transaction Utility (TU) of each transaction is computed. Then TWU of each single item is also accumulated.
Discarding global unpromising items.
Utilities of unpromising items are eliminated from the TU of the transaction.
Then remaining promising items in the transaction are sorted according to the descending order of TWU.

2. Second scan:-

UP-Tree is constructed by inserting transactions.

4) UP-Growth Algorithm - UP-Growth efficiently generates PHUIs from the global UP-Tree with two strategies, namely DLU (Discarding local unpromising items) and DLN (Decreasing local node utilities). For this Minimum Item Utility Table, abbreviated as MIUT, is used to maintain the minimum item utility for all global promising items. In DLU(Discarding local unpromising items) strategy the minimum item utilities of unpromising items are discarded from path utilities of the paths during the construction of a local UP-Tree. In DLN (Decreasing local node utilities) the minimum item utilities of descendant nodes for the node are decreased during the construction of a local UP-Tree. It is applied during the insertion of the reorganized paths.

5) UP-Growth plus Algorithm - Applying UP-Tree to the UP-Growth takes more execution time for Phase II. A modified algorithm i.e. UP-Growth plus reduce the execution time by effectively identifying high utility itemsets. It computes the Maximum transaction Weighted Utilization (MTWU) from all items and considering multiple of *min_sup* as a user specified threshold value.

The main difference between above stated algorithm is that in UP-Growth , minimum item utility table is used to reduce the overestimated utilities and in UP-Growth plus, minimal node utilities in each path are used to make the estimated pruning values closer to real utility values of the pruned items in database.

V. RESULT AND EVALUATION

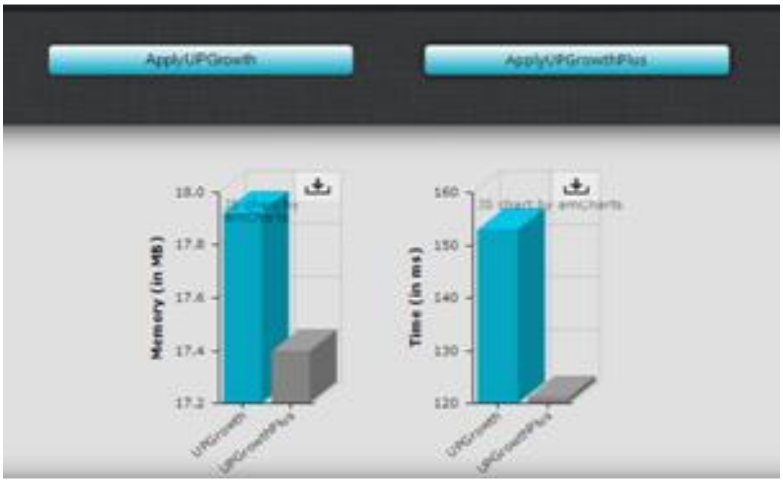
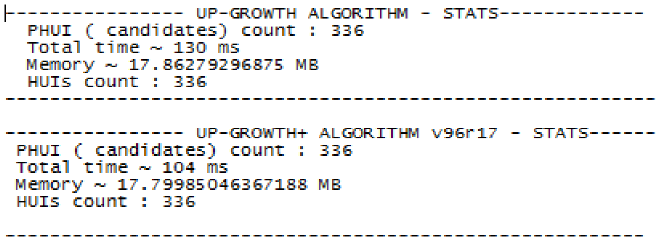


Fig 2: snapshot for showing the result after applying UP Growth and UP Growth Plus algorithm



```
|---------------- UP-GROWTH ALGORITHM - STATS-------------
  PHUI ( candidates) count : 336
  Total time ~ 130 ms
  Memory ~ 17.86279296875 MB
  HUIs count : 336
----------------------------------------------------------

---------------- UP-GROWTH+ ALGORITHM v96r17 - STATS------
 PHUI ( candidates) count : 336
 Total time ~ 104 ms
 Memory ~ 17.7998504636718 8 MB
 HUIs count : 336

---------------------------------------------------------
```

Fig 3: snapshot for showing Performance Comparison of UP growth and UP growth plus Algorithm

The results are analysed from the above two snapshot that for finding high utility itemsets , UP Growth Plus algorithm required less execution time and less memory usage as compared to UP Growth for mining high utility itemsets count from the set of PHUI count with less number of candidate itemsets generation.

VI. CONCLUSION

In this paper, we have proposed two algorithms named UP Growth and UP-Growth plus for mining high utility itemsets from transaction databases. A data structure named UP-Tree was proposed for maintaining the information of high utility itemsets. PHUIs can be efficiently generated from UP-Tree with only two database scans. Moreover, we developed several strategies to decrease overestimated utility and enhance the performance of utility mining. Comparison results show that the strategies considerably improved performance by reducing both the search space and the number of candidates. Proposed algorithms, especially UP Growth plus is more efficient than UP Growth substantially especially when databases contain lots of long transactions.

REFERENCES

[1] Sadak Murali & Kolla Morarjee, "*A Novel Mining Algorithm for High Utility Itemsets from Transactional Databases*," Global Journal of Computer Science and Technology Software & Data Engineering Volume 13 Year 2013

[2] Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu, Fellow," *Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases*" IEEE Trans. Knowledge and Data Eng., VOL. 25, NO. 8, AUGUST 2013.

[3] 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.

[4] 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.

[5] W. Wang, J. Yang, and P. Yu, "*Efficient Mining of Weighted Association Rules (WAR*)," Proc. ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD '00), pp. 270-274, 2000.

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

[7] Y.-C. Li, J.-S.Yeh, and C.-C. Chang, "*Isolated Items Discarding Strategy for Discovering High Utility Itemsets*," Data and Knowledge Eng., vol. 64, no. 1, pp. 198-217, Jan. 2008.

[8] 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.

[9] 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.

[10] H.F.Li, H.Y. Huang , Y.Cheng Chen, y. Liu, S.Lee, "Fast *and memory efficient mining of high utility itemsets in data streams*", Eight International Conference of Data Mining 2008.

[11] J.Pillai , O.P.Vyas ,"*Overview of itemset utility mining and its applications*" ,International Journal of Computer Applications (0975-8887), Volume 5-No.11 ,August 2010.

[12] H.Yao, H.J.Hamilton , "*Mining itemset utilities from transaction databases*", in Data and Knowledge Engineering 59(2006) pp.603-626.