Discovery of Weighted Continual Itemsets from Transactional Databases using Frequent Utility Pattern Algorithm

Hanegaonkar Mohammed Awais[1], **Kamlesh Amravatkar**[2]

[1]*Central Water & Power Research Station, Pune, Maharashtra*
[2]*Matoshri Pratishthan Group of Institutions, Nanded, Maharashtra*

---***---

Abstract - *Data Mining is nothing but 'knowledge mining from data'. High Utility Itemsets can be mined from Transactional databases. It refers to discovery of itemsets with high utility like profits. Many algorithms have been proposed in recent years in this area of research. However, they incur producing a large number of candidate itemsets for high utility itemsets. Such a large number of itemsets degrades the mining performance in terms of space requirement and execution time. This problem increases with the size of database which contains lots of high utility itemsets or long transactions. In this paper, we propose an algorithm which has set of effective strategies for pruning candidate itemsets as periodicity for mining high utility itemsets. A tree-based structure called Continual Utility Pattern (CUP) tree is used to maintain the information of high utility itemsets. This allows to efficiently generate candidate itemsets with only two scans of database. Experimental results shows that the proposed algorithm not only reduces the number of candidates effectively but also outperform other existing algorithms substantially in terms of databases contain lots of very long running transactions.*

Key Words: Data Mining, Frequent Itemsets, Continual UP, Utility Pattern, Candidate Pruning, High Utility Itemsets, Utility Mining, CUP-Tree

1.INTRODUCTION

Data mining is the process of revealing previously unknown, nontrivial and potentially useful information from large databases. Discovering useful patterns hidden in a database plays an essential role in several data mining tasks, such as high utility pattern mining, frequent pattern mining and weighted frequent pattern mining. Out of this, frequent pattern mining is very fundamental research topic that has been applied to different kinds of databases, such as transactional databases [1], streaming databases [2], and time series databases [3], and various other application domains such as Web click-stream analysis [4], mobile environments [5], and most importantly bioinformatics [6],[7].

Although, mining high utility itemsets from databases is not an easy task since downward closure property[1] in frequent itemset mining does not hold. As superset of a low-utility itemset may be a high utility itemset, pruning search space for high utility itemset becomes the difficult task. A naive solution to solve this type of problem is using the principle of exhaustion to enumerate all itemsets from databases. However, this method suffers problem of large search space specially when databases contain large number of long transactions or a low minimum utility threshold is set.

1.1 Literature Review

Existing Studies [8] [9] applied overestimated methods to facilitate the performance of utility mining. These studies search for potential high utility itemsets (PHUIs) first subsequently followed by additional database scans to identify their utilities. Different existing algorithms often generate very large number of PHUIs and hence resulting to overall degraded performance. Algorithms time complexity varies with the number of PHUIs, leading to a challenging problem when low thresholds are set or databases contains many long running transactions. This situation becomes worse when very low threshold are set or many long running long running transactions are present in the database. Algorithm time complexity depends on the number of PHUIs generated by the algorithm, more the number of PHUIs results in degraded performance.

R. Agrawal [10] introduced the concept of frequent itemset mining. It is an itemset whose support is greater than some user specified minimum support threshold.

Extensive studies have been proposed in past years for mining frequent patterns [1],[2],[11],[12]. The most famous frequent pattern mining algorithms are association rule mining [1] [11] [12] and sequential pattern mining [13]. A colonist for efficiently mining association rules from large databases is Apriori [14]. FP-Growth [14] is also proposed later as pattern growth-based association mining algorithm. It is broadly recognized that FP-Growth achieves better performance than its competitor like Apriori-based algorithm. This was possible as FP-Growth identifies frequent itemsets without generating any candidate itemsets and database scan is done just twice.

Cai et al. proposed the concept of weighted association rules [15] with weighted items. Performance of this algorithm could not be improved further as the framework of weighted association rules does not have downward closure property.

Later, weighted downward closure property [16] have been proposed. It uses transaction weight, weighted support to maintain the downward closure property during the mining process. This was proposed by Tao et al. In spite of the fact that, weighted association rule mining considers the importance of items, applications like transaction databases - items' quantities in transactions were not taken into consideration. Thus, issue of high utility mining [8] [9] is raised.

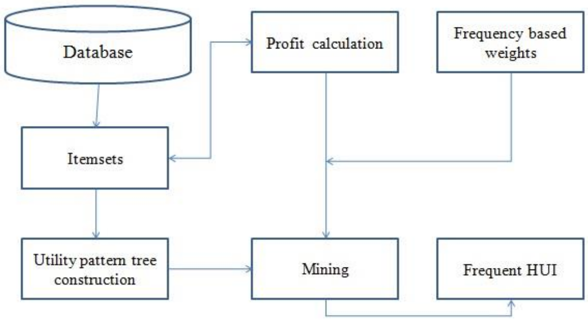


Fig -1: System Overview

The solution to this problem is proposed by Liu et al using an algorithm which is Two Phase [18] and is mainly composed of two mining phases only. In first phase, apriori based level wise method was used to enumerate high transaction weighted utility itemset (HTWUIs). Then Transaction Weighted Utility (TWU) is calculated for each candidate item set of length k. In the second phase, high utility itemsets are identified by performing one more database scan. Although this algorithm reduces search space by using transaction weighted downward closure property (TWDC), too many candidates to obtain HTWUIs are generated and requires multiple database scans. Later Isolated Item Discarding Strategy (IIDS) have been proposed by Li et al [17].

2. PROPOSED SYSTEM

The framework of the proposed system consists of steps – Scan the database to construct a CUP-Tree (Continual - Utility Pattern Tree). Recursive PHUIs from global UP-Tree and local UP-Tree are generated. Then identification of actual high utility itemsets are done from the set of PHUIs. Using effective strategies, the set of PHUIs are much smaller.

2.1 The Propounded Data Structure: CUP-Tree

Performance of existing algorithms is comparatively poor as large numbers of PHUIs are getting generated. When low threshold values are set and the databases contains very log running transactions then the situations becomes more critical.

To ease the mining performance and also to stay from scanning original database repeatedly, we use a dense tree structure, called CUP-Tree.

This CUP-Tree has information of database transactions and high utility itemsets. Each Node in CUP-Tree consists of its name, count, node's utility i.e. overestimated utilities of the node, information of parent node, and a pointer which points to the other node having same name as of current node. Pointers in the header table will help us to access all the nodes having same name efficiently in the CUP-Tree. A Header Table also facilitates the traversal of CUP-Tree, as entry in the table consists of an item name, an overestimated utility and a pointer. The pointer points to the last occurrence of the node which has the same item as the entry in the CUP-Tree.

Two strategies that are applied to minimize the overestimated utilities of each node during the construction of a global CUP-Tree are explained below.

2.1.1 Discarding Global Unpromising Items during CUP-Tree Construction

The construction of a global CUP-Tree can be performed with the two scans of the original database. In the first scan, the Transaction Utility (TU) and TWU of each item are calculated. An Item (i_p) is called a promising item if TWU(i_p) > min_util. If not, it is called unpromising item. min_util is the user-specified minimum utility threshold. An itemset is called high utility itemset if its utility is greater than min_util, otherwise it is called low-utility itemset. Without loss of generality, an item is also called a promising item if its overestimated utility is no less than min_util. Otherwise it is called unpromising item.

In the second scan, transactions (in the form of nodes) are inserted into the CUP-Tree. When a node is traversed to retrieve, the unpromising items from the transactions should be removed and their utilities should also be eliminated.

2.1.2 Discarding Global Node Utilities during CUP-Tree Construction

The tree-based framework for high utility itemset mining applies the divide and conquer technique in the mining processes. Hence, the search space is divided into subspaces. The items that are successors of nodes of the item i_m does not appears in the {i_m}-tree. Only the predecessor nodes of i_m appears in {i_m}-tree. This forms our second strategy for decreasing overestimated utilities is to remove the utilities of successor nodes from their node utilities in the global CUP-Tree.

By applying this strategy, the utilities of the nodes that are closer to the root of a global CUP-Tree are further reduced. This is useful when the database contains lots of long running transactions. Hence, more the number of utilities being discarded from the CUP-Tree for more items in the transactions.

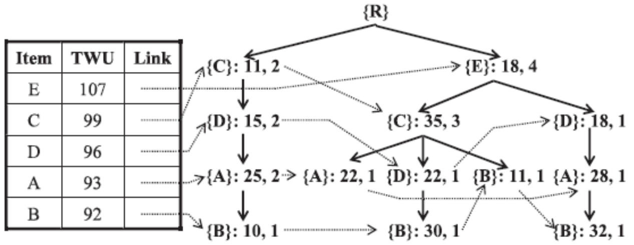
Below table shows the minimum utility table for global promising items of the database. It is important to note that, minimum item utilities of all the items can be collected during the first scan of the original database. Minimum item utilities are utilized to reduce utilities of local unpromising items in conditional patterns instead of exact utilities.

Table -1: Minimum Utility Table

Item	A	B	C	D	E
Minimum Item Utility	5	2	1	2	3

Utility(item) <= OEU(item)

Here, OEU is the overestimated utility of itemsets which is less than the minimum utility i.e. Utility(item)<min_util. It means, Utility(i)<OEU(i) < min_util. Only the supersets of promising items are possible to be high utility itemsets.



Item	TWU	Link
E	107	
C	99	
D	96	
A	93	
B	92	

Fig – 2: CUP-Tree after applying stratagies

3. PROPOSED MINING METHOD

UP-Growth algorithm achieves better performance than FP-Growth by using strategies explained in section 2.1 to decrease the overestimated utilities of itemsets. However, the overestimated utilities can be closer to their actual utilities if we eliminate estimated utilities which are close to actual utilities of unpromising items and descendant nodes.

After constructing a global UP-tree, a basic method for generating PHUIs is to mine UP-Tree by FP-Growth. Proposed novice algorithm for reducing overestimated utilities more efficiently and effectively uses some more strategies, and thus is better than UP-Growth$^+$ algorithm.

1. Trace the paths in the original tree to populate conditional pattern bases.
2. Using these conditional pattern bases, construct conditional tree.
3. Mine patterns from the conditional tree.

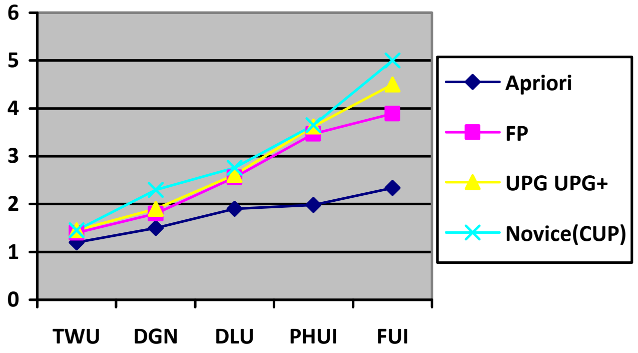


Chart -1: Performance Comparison Chart

4. RESULTS & CONCLUSIONS

Performance of proposed algorithms is compared to other state of art algorithms and it is found that the proposed algorithm outperform very well almost in all cases.

Table -2: Comparison of Candidate generation with respect to execution time of transaction

Apriori	FP Growth	Upg &Upg+	Novel
1.2	1.4	1.45	1.46
1.5	1.8	1.9	2.3
1.9	2.56	2.61	2.73
1.98	3.47	3.62	3.65
2.34	3.89	4.51	5.03

We have recognized subroutine to calculate frequent items from transactions; other researchers are requested to observe their results.

Proposed Mining algorithm discovers high utility itemsets by considering execution time, frequency based weights and also profits.

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] 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

[3] 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.

[4] M.-S. Chen, J.-S. Park, and P.S. Yu, “Efficient Data Mining for Path Traversal Patterns,”IEEE Trans. Knowledge and Data Eng.,vol. 10, no. 2, pp. 209-221, Mar. 1998.

[5] S.C. Lee, J. Paik, J. Ok, I. Song, and U.M. Kim, “Efficient Mining of User Behaviors by Temporal Mobile Access Patterns,”Int’l J. Computer Science Security,vol. 7, no. 2, pp. 285-291, 2007.

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

[7] E. Georgii, L. Richter, U. Ru¨ckert, and S. Kramer, “Analyzing Microarray Data Using Quantitative Association Rules,”Bioinformatics,vol. 21, pp. 123-129, 2005.

[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] 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.

[10] Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu, Fellow,”Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases”, Vol.. 25, No. 8, Aug 2013

[11] J. Han and Y. Fu, “Discovery of Multiple-Level Association Rules from Large Databases,” Proc. 21th Int’l Conf. Very Large Data Bases, pp. 420-431, Sept. 1995.

[12] 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.

[13] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Moal, and M.C. Hsu, “Mining Sequential Patterns by Pattern-Growth: The Prefixspan Approach,” IEEE Trans. Knowledge and Data Eng.,vol.16, no.10, pp. 1424-1440, Oct. 2004.

[14] 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.

[15] C.H. Cai, A.W.C. Fu, C.H. Cheng, and W.W. Kwong, “Mining Association Rules with Weighted Items,” Proc. Int’l Database Eng. and Applications Symp. (IDEAS ’98), pp. 68-77, 1998.

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

[17] 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.

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

BIOGRAPHIES

	Mr. Hanegaonkar Mohammed Awais obtained Bachelor’s of Engineering from M.G.M’s College of Engineering, Nanded. He is currently working as R.A. in Central Water and Power Research Station, Pune. He has worked as Senior Software Engineer in Persistent Systems. He has knowledge in the field of Data mining to identify Continual Itemsets from transactional databases. His area of interests also includes BigData, Hadoop, Apache Spark etc.
	Mr. Kamlesh Amrawatkar obtained his Master's of Engineering (Software Engineering) from San Jose State University, California, United States. He is currently as Assistant Professor in Matoshri Pratishthan Group of Institutions, Nanded. His areas of interests include Software Engineering and Data Mining.