IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 5, Oct-Nov, 2013
ISSN: 2320 - 8791
www.ijreat.org

A Survey on Efficient Algorithm for Mining High Utility Itemsets

Sadak Murali[1] and Kolla Morarjee[2]

[1]Department of Computer Science and Engineering, CMR Institute of Technology, Medchal, 501401, Hyderabad, Andhra Pradesh, India

[2] Asst. Professor, Department of Computer Science and Engineering, CMR Institute of Technology, Medchal, 501401, Hyderabad, Andhra Pradesh, India

Abstract

Efficient discovery of frequent itemsets in large datasets is a crucial task of data mining. From the past few years many methods have been proposed for generating high utility patterns, by this there are some problems as producing a large number of candidate itemsets for high utility itemsets and probably degrades mining performance in terms of speed and space. The compact tree structure which is proposed recently, viz., FP-Tree and UP-Tree, these maintains the information of transaction and itemsets, mining performance and avoid scanning original database repeatedly. In this paper to obtain a structured way up-tree is adopted, it scans the database only twice to obtain candidate items and manage them in an efficient data structured way. Applying UP-Tree to the UP-Growth takes more execution time for phase II. Hence in this paper we present the modified algorithm aiming to reduce the execution time by efficiently identifying high utility itemsets.

Keywords**:** *candidate pruning, frequent itemsets, utility mining, high utility itemsets.*

1. Introduction

Data mining refers to extracting or mining knowledge from large amounts of data. Thus, data mining should have been more appropriately named "knowledge mining from data". The task of finding frequent patterns in data mining in large databases is very important use full in many applications over the past few years. The primary goal is to discover hidden patterns, unexpected trends in the data. Data mining is concerned with analysis of large volumes of data to automatically discover interesting regularities or relationships which in turn leads to better understanding of the underlying processes. Data mining activities uses combination of techniques from database artificial intelligence, statistics, technologies machine learning. This includes bioinformatics, genetics, medicine, clinical research, education, retail and marketing research.

Utility mining is one of the most challenging data mining tasks is the mining of high utility itemsets efficiently. Identification of the itemsets with high utilities is called as Utility Mining. The utility can be measured as per the user preferences utility can be measured in terms of cost, profit or other expressions. The limitations of frequent or rare 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.

Mining high utility itemsets from databases refers to finding the itemsets with high profits. Here, the meaning of itemset utility is interestingness, importance or profitability of an item to users. High utility itemsets mining has become one of the most interesting data mining tasks with broad applications and it identifies itemsets whose utility satisfies a given threshold. By using different values it allows users to quantify the

usefulness or preferences of items using different values. A high utility itemset is defined as: A group of items in a transaction database is called itemset. This itemset in a transaction database consists of two aspects: First one is itemset in a single transaction is called internal utility and second one is itemset in different transaction database is called external utility. Mining high utility itemsets from databases is an important task has a wide range of applications such as website click stream analysis [13, 16, 21], business promotion in chain hypermarkets, cross-marketing in retail stores [4, 9, 14, 22, 24], online e-commerce management, mobile commerce environment planning and even finding important patterns in biomedical applications.

The *frequent itemset mining* [2] is to find items that co-occur in a transaction database above a user given frequency threshold, without considering the quantity or weight such as profit of the items. An itemset can be defined as a non-empty set of items. An itemset with k different items is termed as a k-itemset. For e.g. {bread, butter, milk} may denote a 3-itemset in a supermarket transaction. The notion of frequent itemsets was introduced by Agrawal et al [2].Frequent itemsets are the itemsets that appear frequently in the transactions. Frequent itemsets are the itemsets that occur frequently in the transaction data set. The goal of frequent itemset mining is to identify all the itemsets in a transaction dataset. The itemsets which appear frequently in the transactions are called frequent itemsets.

In Data Mining the task of *finding frequent pattern* in large databases is very important use full in many applications over the past few years. This task is computationally more expensive, especially when a large number of patterns exist. This large number of patterns which are mined during the various approaches makes the user very difficult to identify the patterns which are very interesting for him. The goal of frequent itemset mining is to identify all frequent itemsets. The generations of association rules are straight forward, once the frequent itemsets are identified. In the real world, however, each item in the supermarket has a different importance/price and single customer will be interested in buying multiple copies of same item. Therefore, finding only traditional frequent patterns in a database cannot fulfill the requirement of finding the most valuable customers/itemsets that contribute the most to the total profit in a retail business.

2. Literature Review

R. Agrawal et al in [2] proposed Apriori algorithm, it is used to obtain frequent itemsets from the database. in miming the association rules we have the problem to generate all association rules that have support and confidence greater than the user specified minimum support and minimum confidence respectively. The first pass of the algorithm simply counts item occurrences to determine the large 1-itemsets. First it generates the candidate sequences and then it chooses the large sequences from the candidate ones. Next, the database is scanned and the support of candidates is counted. The second step involves generating association rules from frequent itemsets. Candidate itemsets are stored in a hash-tree. The hash-tree node contains either a list of itemsets or a hash table. Apriori is a classic algorithm for frequent itemset mining and association rule learning over transactional databases. After identifying the large itemsets, only those itemsets are allowed which have the support greater than the minimum support allowed. Apriori Algorithm generates lot of candidate item sets and scans database every time. When a new transaction is added to the database then it should rescan the entire database again.

J. Han et al in [11] proposed frequent pattern tree *(FP-tree) structure,* an extended prefix tree structure for storing crucial information about frequent patterns, compressed 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. FP-growth is not able to find high utility itemsets.

W. Wang et al in [23] proposed weighted association rule. In WAR, we discover first frequent

itemsets and the weighted association rules for each frequent itemset are generated. In WAR, we use a twofold approach. First it generates frequent itemsets; here we ignore the weight associated with each item in the transaction. In second for each frequent itemset the WAR finds that meet the support, confidence. Weighted association rule mining first proposed the concept of weighted items and weighted association rules. However, the weighted association rules does not have downward closure property, mining performance cannot be improved. 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.

Liu et al in [15] proposes a Two-phase algorithm for finding high utility itemsets. The utility mining is to identify high utility itemsets that drive a large portion of the total utility. Utility mining is to find all the itemsets whose utility values are beyond a user specified threshold. Two-Phase algorithm, it efficiently prunes down the number of candidates and obtains the complete set of high utility itemsets. We explain transaction weighted utilization in Phase I, only the combinations of high transaction weighted utilization itemsets are added into the candidate set at each level during the level-wise search. In phase II, only one extra database scan is performed to filter the overestimated itemsets. Two-phase requires fewer database scans, less memory space and less computational cost. It performs very efficiently in terms of speed and memory cost both on synthetic and real databases, even on large databases. In Two-phase, it is just only focused on traditional databases and is not suited for data streams. Two-phase was not proposed for finding temporal high utility itemsets in data streams. However, this must rescan the whole database when added new transactions from data streams. It need more times on processing I/O and CPU cost for finding high utility itemsets.

Li et al in [13] propose two efficient one pass algorithms MHUI-BIT and MHUI-TID for mining high utility itemsets from data streams within a transaction sensitive sliding window. To improve the efficiency of mining high utility itemsets two effective representations of an extended lexicographical tree-based summary data structure and itemset information were developed.

V.S. Tseng et al in [21] proposes a novel method THUI (Temporal High Utility Itemsets)-Mine for mining temporal high utility itemset mining. The temporal high utility itemsets are effectively identified by the novel contribution of THUI-Mine by generating fewer temporal high transaction weighted utilization 2-itemsets such that the time of the execution will be reduced substantially in mining all high utility itemsets in data streams. To generate a progressive set of itemsets THUI-Mine employs a filtering threshold in each partition. In this way, the process of discovering all temporal high utility itemsets under all time windows of data streams can be achieved effectively. The temporal high utility itemsets with less candidate itemsets and higher performance can be discovered by THUI- mine. From these candidate *k*-itemsets to find a set of high utility itemsets finally, it needs one more scan over the database. Huge memory requirement and lot of false candidate itemsets are the two problems of THUI- Mine algorithm.

J. Hu et al in [12] defines an algorithm for frequent item set mining, that identify high utility item combinations. The goal of the algorithm is different from the frequent item mining techniques and traditional association rule. This algorithm is to find segment of data, which is defined with the combination of few items i.e. rules, a predefined objective function and satisfy certain conditions as a group. The problem considered in high utility pattern mining is different from former approaches as it conducts rule discovery with respect to the overall criterion for the mined set as well as with respect to individual attributes.

Erwin et al in [8] observed that the conventional candidate-generate-and-test approach for identifying high utility itemsets is not suitable for dense date sets. The high utility itemsets are mined using the pattern growth approach is the novel algorithm called CTU-Mine.

Shankar [19] presents a novel algorithm Fast Utility Mining (FUM) which finds all high utility itemsets within the given utility constraint threshold. To generate different types of itemsets the authors also suggest a technique such as Low Utility and High Frequency (LUHF) and Low Utility and Low Frequency

(LULF), High Utility and High Frequency (HUHF), High Utility and Low Frequency (HULF).

Cheng-Wei Wu et al in [22] presented a novel algorithm with a compact data structure for efficiently discovering high utility itemsets from transactional databases. Depending on the construction of a global UP-tree the high utility itemsets are generated using UP-Growth which is one of the efficient algorithms. In phase-I three steps are followed by framework of UP-tree as: (i). UP-Tree construction, (ii). Generation of PHUIs from the UP-Tree and (iii). The high utility itemsets should be identified using PHUI.

Global UP-Tree construction is as follows as: (i). To eliminate the low utility items and their utilities from the transaction utilities is done by discarding global unpromising items (i.e., DGU strategy), (ii). During global UP-Tree construction discarding global node utilities (i.e., DGN strategy) the node utilities which are nearer to UP-Tree root node are effectively reduced by DGN strategy. The PHUI is similar to TWU, in which the itemsets utility is computed with the help of estimated utility and from PHUIs value the high utility itemsets (not less than min_sup) have been identified finally. The global UP-Tree contains many sub paths. From bottom node of header table the each path is considered. And the path is named as conditional pattern base (CPB).

Even the numbers of candidates in Phase 1 are efficiently reduced by DGU and DGN strategies. (i.e., global UP-Tree). But during the construction of the local UP-Tree (Phase-2) they cannot be applied. For discarding utilities of low utility items from path utilities of the paths DLU strategy should be used instead of it and for discarding item utilities of descendant nodes during the local UP-Tree construction DLN strategy should be used. Even though the algorithm is facing still some performance issues in Phase-2.

Authors	**Algorithm**	**Features**	**Problem**
R.Agrawal et. al	Apriori	Frequent itemsets, candidate and association rule.	More candidates generation and rescan database every time.
J. Han et. al	FP-growth	Frequent itemsets without candidate key generation and less time.	
W. Wang et. al	WAR	Items with support and confidence, weights for items.	Downward approach and no high priority data.
Liu et. al	Two-Phase	High utility itemset in traditional database , less candidate	Rescan database, no temporal itemsets
Tseng et. al	THUI-Mine	Generates few candidate and high performance	Lot of false candidate itemsets
Li et. al	MHUI-BIT & MHUT-TID	Item information, lexTree and HTU for data stream	More time and candidate test fails
J. Hu et. al	High yield partition tree	Binary tree partition, iterations to prune item	Lot of low utility values
Erwin et. al	CTU-Mine	High utility itemset for pattern growth and dense data	Overestimated real utility
V.S. Tseng et. al	UP-Growth	Pruning candidate itemset with two scans.	Performance,

3. Conclusion

This paper presents a survey on various High Utility Itemsets algorithms that were proposed by earlier researches for the better development in the field of Data Mining. Various algorithms and methods discussed above will help in developing efficient and effective High utility itemsets for data mining. In the future scope, we will be presenting a comparative study of various algorithms for mining high utility itemset.

References

[1] R. Agrawal , T. Imielinski, A. Swami, "Mining association rules between sets of items in large databases", in *proceedings of the ACM SIGMOD International Conference on Management of data*, pp. 207-216, 1993

[2] R. Agrawal and R. Srikant. "Fast algorithms for mining association rules," in *Proc. of the 20th VLDB Conf.*, pp. 487-499, 1994.

[3] R. Agrawal and R. Srikant, "Mining Sequential Patterns," in *Proc. of the 11th Int'l Conference on Data Engineering*, pp. 3-14, Mar., 1995.

[4] 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 Transactions on Knowledge and Data Engineering*, Vol. 21, Issue 12, pp. 1708-1721, 2009.

[5] C. H. Cai, A. W. C. Fu, C. H. Cheng and W. W. Kwong, "Mining Association Rules with Weighted Items," in *Proc. of the Int'l Database Engineering and Applications Symposium* (IDEAS 1998), pp. 68-77, 1998.

[6] R. Chan, Q. Yang and Y. Shen. "Mining high utility itemsets," in *Proc. of Third IEEE Int'l Conf. on Data Mining*, pp. 19-26, Nov., 2003.

[7] M.-S. Chen, J.-S. Park and P. S. Yu, "Efficient data mining for path traversal patterns," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 10, no. 2, pp. 209-221, 1998.

[8] A. Erwin, R. P. Gopalan and N. R. Achuthan, "Efficient mining of high utility itemsets from large datasets," in *Proc. of PAKDD 2008, LNAI 5012*, pp. 554-561.

[9] J. Han, G. Dong, Y. Yin, "Efficient Mining of Partial Periodic Patterns in Time Series Database," in *Proc. of the Int'l Conf. on Data Engineering*, pp. 106-115, 1999.

[10] J. Han and Y. Fu, "Discovery of multiple-level association rules from large databases," in *Proc. 21th VLDB Conf.*, Sep. 1995, pp. 420–431.

[11] J. Han, J. Pei, Y. Yin, "Mining frequent patterns without candidate generation," in *Proc. of the ACM-SIGMOD Int'l Conf. on Management of Data*, pp. 1-12, 2000.

[12] J. Hu, A. Mojsilovic, "High-utility pattern mining: A method for discovery of high-utility item sets", Pattern Recognition 40 (2007) 3317 – 3324.

[13] H. F. Li, H. Y. Huang, Y. C. Chen, Y. J. Liu and S. Y. Lee, "Fast and Memory Efficient Mining of High Utility Itemsets in Data Streams," in *Proc. of the 8th IEEE Int'l Conf. on Data Mining*, pp. 881-886, 2008.

[14] 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," in *Proc. of the SIAM Int'l Conference on Data Mining* (SDM 2005), 2005.

[15] Y. Liu, W. Liao and A. Choudhary, "A fast high utility itemsets mining algorithm," in *Proc. of the Utility-Based Data Mining Workshop*, 2005.

[16] B.-E. Shie, V. S. Tseng and P. S. Yu, "Online mining of temporal maximal utility itemsets from data streams," in *Proc. of the 25th Annual ACM Symposium on Applied Computing*, Switzerland, Mar., 2010.

[17] K. Sun and F. Bai, "Mining Weighted Association Rules without Preassigned Weights," *IEEE Trans. on Knowledge and Data Engineering*, Vol. 20, No. 4, 2008.

[18] S. K. Tanbeer, C. F. Ahmed, B.-S. Jeong and Y.-K. Lee, "Efficient frequent pattern mining over data streams," in *Proc. of the ACM 17th Conference on Information and Knowledge Management*, 2008.

[19] S.Shankar, T.P.Purusothoman, S. Jayanthi,N.Babu, A fast agorithm for mining high utility itemsets , in :Proceedings of IEEE International Advance Computing Conference (IACC 2009), Patiala, India, pp.1459-1464

[20] F. Tao, F. Murtagh and M. Farid, "Weighted Association Rule Mining using Weighted Support and Significance Framework," in *Proc. of the ACM SIGKDD Conf. on Knowledge Discovery and Data Mining* (KDD 2003), pp. 661-666, 2003.

[21] V. S. Tseng, C. J. Chu and T. Liang, "Efficient Mining of Temporal High Utility Itemsets from Data streams," in *Proc. of ACM KDD Workshop on Utility-Based Data Mining Workshop* (UBDM'06), USA, Aug., 2006.

[22] V. S. Tseng, C.-W. Wu, B.-E. Shie and P. S. Yu, "UP-Growth: An Efficient Algorithm for High Utility Itemsets Mining," in *Proc. of the 16th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining* (KDD 2010), pp. 253-262, 2010.

[23] W. Wang, J. Yang and P. Yu, "Efficient mining of weighted association rules (WAR)," in *Proc. of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (KDD 2000), pp. 270-274, 2000.

[24] H. Yao, H. J. Hamilton and L. Geng, "A unified framework for utility-based measures for mining itemsets," in *Proc. of ACM SIGKDD 2nd Workshop on Utility- Based Data Mining*, pp. 28-37, USA, Aug., 2006.

[25] C.-H. Yun and M.-S. Chen, "Using pattern-join and purchase-combination for mining web transaction patterns in an electronic commerce environment," in *Proc. of 24th IEEE Annu. Int. Computer Software and Application Conf.*, pp. 99–104, Oct., 2000.