Spvryan's International Journal of Engineering Sciences & Technology (SEST)
ISSN : 2394-0905

Survey on Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases

MR.S.H.Sangale
Research Scholar, Dept. of Computer Engg. R.H. Sapat College, Pune University, Nashik , Maharashtra, India

Prof . Mrs. R. C. Samant
Asst. Professor, Dept. of Computer Engg. R.H. Sapat College, Pune University, Nashik , Maharashtra, India

Abstract: Data Mining high utility itemsets from a large transactional database refers to the discovery of knowledge like high utility itemsets (profits). Since a number of relevant algorithms have been proposed in past years, they fall into the problem of producing a large number of candidate itemsets for high utility itemsets. Such a huge number of candidate itemsets decrease the mining performance in terms of time and space complexity. The situation may become worse when the database contains lots of long transactions or long high utility itemsets. An emerging concept in the field of data mining is utility mining. To identify the itemsets with highest utilities is the main objective of utility mining , by considering profit, quantity, cost or other user preferences. This topic is having many applications in website click stream analysis, cross marketing in retail stores, business promotion in chain hypermarkets, online e-commerce management, finding important patterns in biomedical applications and mobile commerce environment planning.

Keywords: *Candidate pruning, frequent itemset, high utility itemset, utility mining, data mining.*

1. INTRODUCTION

Data mining, also known as knowledge discovery in databases has established its position as a prominent and important research area. The goal of data mining is to extract higher-level hidden information from an abundance of raw data. Data mining has been used in various data domains. Data mining can be regarded as an algorithmic process that takes data as input and yields patterns, such as classification rules, item sets, association rules, or summaries, as output. The traditional Association Rule Mining (ARM) approaches consider the utility of the items by its presence in the transaction set. The frequency of item set is not sufficient to reflect the actual utility of an item set. For example, the sales manager may not be interested in frequent item sets that do not generate significant profit. Recently, one of the most challenging data mining tasks is the mining of high utility item sets efficiently. Identification of the item sets with high utilities is called as Utility Mining.

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.

In view of this, utility mining emerges as an important topic in data mining field. 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. Utility of items in a transaction database consists of two aspects, External utility: The importance of distinct items, which is called external utility and Internal utility: The importance of items in transactions, which is called internal utility. Utility of an itemset is defined as the product of its external utility and its internal utility. An itemset is called a high utility itemset. If its utility is no less than a user-specified minimum utility threshold; otherwise, it is called a low-utility itemset.

1.1 Association rule Mining

Mining Association rules is one of the research problems in data mining [1]. Given a set of transactions where each transaction is a set of items, an association rule is an expression of the form X=>Y, where X and Y are sets of items. The problem of mining association rules was first introduced in [1] and later broadened in [2], for these of databases consisting of categorical attributes alone. **Association rule mining (ARM)** is a popular technique for finding co-occurrences, correlations, frequent patterns, associations among items in a set of transactions or database. Rules with confidence and support above user-defined thresholds were found. As data continues to grow and its complexity increases, newer data structures and algorithms are being developed to match this development. Once the frequent item sets are found, association rules are generated [3]. ARM is widely used in market-basket analysis. For example, frequent item sets can be found out by analysing market basket data and then association rules can be generated by predicting the purchase of other items by conditional probability [1], [2].

1.2 Frequent Item set Mining

R. Agrawal et al in [1] introduced the concept of frequent item set mining. **Frequent item sets** are the item sets that occur frequently in the transaction dataset. The goal of **Frequent Item set Mining** is to identify all the frequent item sets in a transaction dataset. Let $I = \{i1, i2, \ldots, in\}$ be a set of n distinct literals called *items*. An **item set** is a non-empty set of items. An item set $X = (i1, i2, \ldots, ik)$ with k items is referred to as k-item set, A **transaction** $T = \langle$TID, $(i1, i2, \ldots, i\,k)\rangle$ consists of a transaction identifier (TID) and a set of items $(i1, i2, \ldots, i\,k)$, where $ij \hat{I} I, j = 1, 2, \ldots, k$. The frequency of an item set X is the probability of Occurring in a transaction T. A frequent item set is the item set having frequency support greater a minimum
User specified threshold

1.3 Rare Item set Mining

The basic Bottleneck of association rule mining is Rare Item Problem. Most approaches to mining association rules implicitly consider the utilities of the item sets to be equal [3]. The utilities of item sets may differ. In many applications, some items appear very frequently in the data, while others rarely appear. If frequencies of items vary, two problems encountered –(1) If min sup is set too high, then rules of rare item swill not be found (2) To find rules that

involve both frequent and rare items, min sup has to be set very low. **Rare item sets** are the item sets that occur infrequently in the transaction dataset. In most business applications, frequent item sets may not generate much profit while rare item sets may generate a very high profit .For example [4], in the security field, normal behaviour is very frequent, whereas abnormal or suspicious behaviour is less frequent. Considering database where the behaviour of people in sensitive places such as airports are recorded, if those behaviours are modelled, it is likely to find that normal behaviour scan be represented by frequent patterns and suspicious behaviours by rare patterns. In this paper we have presented a literature survey of the various approaches and algorithms for high-utility mining and rare item set mining.

2. LITERATURE SURVEY

R. Agrawal et al in [2] proposed Apriori algorithm, it is used to find 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 [6] 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 [7] 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 [8] 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 [9] 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 [13] 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 [10] 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 [11] 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.

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.

4. ACKNOWLEDGMENT

We sincerely thanks to all the people who helped us to write this review paper. We also like to thank all the open source software developer communities and the researchers for publishing their research work as a guideline.

5. REFERENCES

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

[2] R. Agrawal and R. Srikant, 1994, "Fast Algorithms for Mining Association Rules", in Proceedings of the 20th International Conference Very Large Databases, pp. 487-499.

[3] H.Yao, H. J. Hamilton, and C. J. Butz, "A Foundational Approach to Mining Item set Utilities from Databases", Proceedings of the Third SIAM International Conference on Data Mining, Orland, Florida, pp. 482-486, 2004.

[4] M. Add, L. Wu, Y. Fang, "Rare Item set Mining", Sixth International conference on Machine Learning and Applications, 2007, pp 73-80.

[5] R. Chan, Q. Yang, Y. D. Shen, "Mining High utility Item sets", In Proc. of the 3rd IEEE Intel.Conf. On Data Mining (ICDM), 2003.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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

[11] 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

[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] 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.

[14] 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

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