International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 6 | ISSN : 2456-3307

# Chui : Mining Closed High Utility Itemsets

**Khushali Kumari*, A.R. Deshpande**
Computer Science, SPPU, Pune, Maharashtra, India

## ABSTRACT

In association rule mining, a transaction is a set of items called itemset, where each item represents a product or a service that customer buy in one transaction. In an e-commerce application, an itemset represents a set of items that a customer bought in one transaction. Frequent Itemset Mining (FIM) is a very popular data mining approach which is essential to a wide range of applications. For a transactional database, FIM generates frequent itemsets i.e. groups of items (itemset) appearing frequently in transactions. However, one of the drawback of FIM is that it assumes that each item can appear only once in every transaction and that all items have the same importance (weight, unit profit or value). To address above mentioned issues, the High-Utility Itemset Mining (HUIM) has been defined. As opposed to FIM, HUI considers the case where items can appear any number of times in a transaction and where each item has a weight called utility (e.g. unit profit). Therefore, mining high utility itemset can be used to discover itemsets having a high-importance (e.g. high profit), that is called High-Utility Itemsets. An itemset is called high utility itemset (HUI) only if its utility is not less than a user-specified minimum utility threshold minutil. Discovering or generating high-utility itemsets in transactional databases is a popular data mining task. A limitation of traditional algorithms is that too many number of high-utility itemsets may be presented to the user out of which some are redundant. To provide a concise and lossless representation of results the support count measure can be considered, hence the concept of closed itemset mining can be used.

**Keywords:** High Utility Itemsets, Itemsets Utility, Transactional Database, Transactional Utility

## I. INTRODUCTION

Mining frequent itemset basically tells us, which set of itemset are very frequent to occur in a transaction. Though frequent itemset mining plays an important role in an e-commerce application, it also has some of drawback such as- (1)FIM assumes that one item can occur only once in a transaction, even if a buyer buys two units of the same item it is considered as one unit only (2)FIM assumes all items same importance, which is not true in an e-commerce application because profit of every item will be different. In frequent itemset mining we find association between different items that occur together very frequently. Whereas, in utility mining we discover those set of itemsets which is likely to give maximum profit(or which is more important) to the vendor. Therefore from vendors perspective it is very important to know which set of items are important to them so that they can earn maximum profit. Only those set of itemsets can be said to give maximum profit whose utility value is either equal or higher than user specified minimum threshold utility value. With increasing growth in e-commerce, mining high utility itemset has become an important research topic. Mining high utility itemset has not only its importance in market basket analysis but also in mobile commerce, click stream analysis, biomedicine and cross-marketing.

There are many algorithms that has been discovered by many scholars for generating high utility itemsets.

In year 2004, an algorithm called two phase[10] high utility itemset mining algorithm was discovered.Though the algorithm was able to discover high utilty itemset but since the algorithm works in two phase, execution time is little high. After Two- Phase[10] many other algorithms were also discovered such as in year 2008 UP-Growth[1] which was based on incremental tree structure of itemsets, and then FHM[4] and so on.

However, an important drawback of traditional approach is that they produce large number of high utility itemsets out of which many are redundant. Time consumed in calculating those huge set of itemsets is very high. Other than this the traditional approach will also consumes large amount of storage space. To consider all these drawback, support count measure of high utility itemsets can be used. Mining high utility itemsets using support is also called closed high utility itemset.

## II. METHODS AND MATERIAL

This section introduces various system modules used in system. The aim here is to make our search efficient and to decrease the number of database scan. Some of the steps are given below-

- The Search Space- In order to create a search space we can select any of search space such as either BFS or DFS. But since in DFS we use depth first approach, so at each level one item gets added to the existing itemset and at the last level we will get complete set of itemset. This can be more clearly shown using figure 1. In figure 2 a set-enumeration tree using lexicographical order for the set say S = {a, b, c, d} is shown.
- Scanning the Database Efficiently- Inorder to scan the database efficiently we need to reduce the database size. There are two methods that can be used to reduce the database size given below-

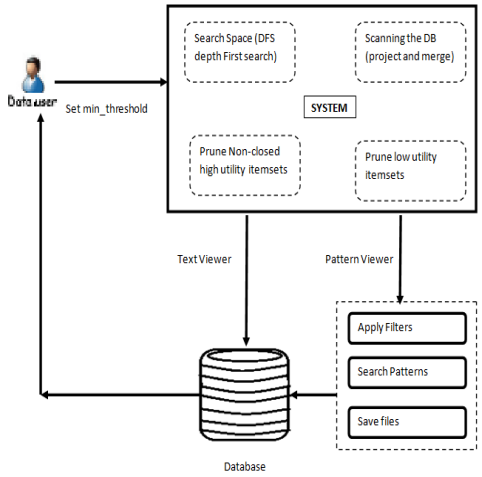


**Figure 1.** System Architecture

**High-utility Database Projection (HDP).** This technique is based on the observation that when an itemset is considered during the depth first search, all items x does not belongs to E(a) can be ignored when scanning the database to calculate the utility of itemsets in the sub-tree of, or upper-bounds on their utility. A database without these items is called a projected database.

**High-utility Transaction Merging (HTM).** There are some transactions that are identical to each other. Dealing with such identical transaction database will only consume memory and increase the execution time. Therefore it is desirable to merge those transactions that are identical and update its support count by increasing its value by one.

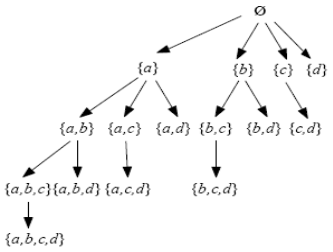


**Figure 2.** Set Enumeration Tree

## III. RESULTS AND DISCUSSION

CHUI system is capable of calculating high utility itemsets, can be judge on four parameters i.e. memory used, execution time, number of visiting nodes and number of high utility itemsets it generates. Expected results of our system is therefore analyzed using above stated four parameters. The results given in figure 3 and 4 are being calculated using a tool called SPMF tool. Since our system is more similar to CHUI-miner(both are using support count for pruning nodes) the expected result will be similar to what CHUI-miner is giving. From the figure 3,4 and 5 we can say that CHUI-miner results are better than CHUD and HUI-miner. In figure 3, on X axis we have number of transactions whereas on Y axis minimum utility threshold. Similarly in figure 4 and 5 Y axis will remain same whereas X will vary for memory and time.

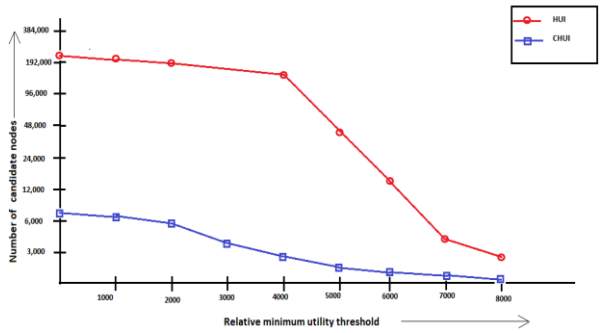


**Figure 3.** Comparison based on Number of candidate nodes generated on varying minimum threshold utility

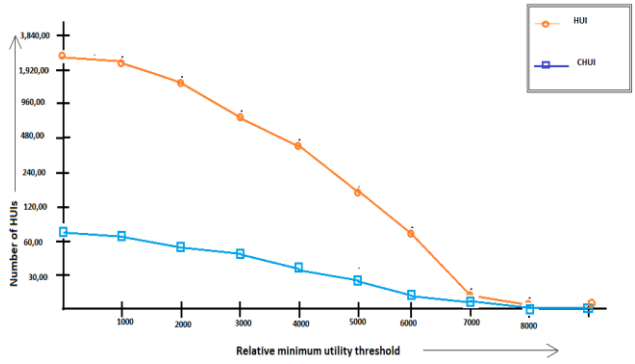


**Figure 4.** Comparison based on Number of High utility itemsets generated on varying minimum threshold utility

## IV. CONCLUSION

Since mining high utility itemset from a transactional database is comparatively difficult task than mining frequent itemset. This is the reason why high utility itemset mining algorithms are slower than frequent itemset mining algorithms. Inorder to deal with above given problem, some of the techniques such as efficient database merging and pruning can be used. And to resolve the drawback of generating too many Figure 4. Comparison based on Number of High utility itemsets generated on varying minimum threshold utility itemsets in output, closed itemset mining can be used. For evaluating and experimental purpose, the result would be using both synthetical and real dataset. For synthetical foodmart, mushrom and chess dataset can be used.

## V. REFERENCES

[1]. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proc. Int. Conf. Very Large Databases, pp. 487{499, (1994)

[2]. Ahmed, C. F., Tanbeer, S. K., Jeong, B.-S., Lee, Y.-K.: 'Efficient tree structures for high-utility pattern mining in incremental databases'. IEEE Trans. Knowl. Data Eng. 21(12), 1708{1721 (2009)

[3]. Fournier-Viger, P., Wu, C.-W., Zida, S., Tseng, V. S.: FHM: Faster high-utility itemset mining using estimated utility co-occurrence pruning. In: Proc. 21st Intern. Symp. on Methodologies for Intell. Syst., pp. 83{92 (2014)}

[4]. Fournier-Viger, P., Gomariz, A., Gueniche, T., Mwamikazi, E., Thomas, R.: Efficient Mining of Top-K Sequential Patterns. In: Proc. 9th Intern. Conf. on Advanced Data Mining and Applications Part I, pp. 109{120, Springer (2013)}

[5]. Fournier-Viger, P., Gomariz, A., Gueniche, T., Soltani, A., Wu., C., Tseng, V. S.: SPMF: a Java Open-Source Pattern Mining Library. Journal of Machine Learning Research (JMLR), 15, pp. 3389-3393 (2014)

[6]. Lan, G. C., Hong, T. P., Tseng, V. S.: An efficient projection-based indexing approach for mining high utility itemsets. Knowl. and Inform. Syst. 38(1), 85{107(2014)}

[7]. Song, W., Liu, Y., Li, J.: BAHUI: Fast and memory efficient mining of high utility itemsets based on bitmap. Intern. Journal of Data Warehousing and Mining. 10(1), 1{15 (2014)}

[8]. Liu, M., Qu, J.: Mining high utility itemsets without candidate generation. In: Proc.22nd ACM Intern. Conf. Info. and Know. Management, pp. 55{64 (2012)}

[9]. Liu, Y., Liao, W., Choudhary, A.: A two-phase algorithm for fast discovery of high utility itemsets. In: Proc. 9th Pacific-Asia Conf. on Knowl. Discovery and Data Mining, pp. 689{695 (2005) }

[10]. Tseng, V. S., Shie, B.-E., Wu, C.-W., Yu., P. S.:Efficient algorithms for mining high utility itemsets from transactional databases. IEEE Trans. Knowl. Data Eng. 25(8), 1772{1786 (2013)}

[11]. Tseng, V., Wu, C., Fournier-Viger, P., Yu, P.: Efficient algorithms for mining the concise and lossless representation of closed+ high utility itemsets. IEEE Trans Knowl. Data Eng. 27(3), 726{739 (2015)}

[12]. T. Uno, M. Kiyomi, H. Arimura, "LCM ver. 2: Efficient mining algorithms for frequent/closed/maximal itemsets," Proc. ICDM'04 Workshop on Frequent Itemset Mining Implementations, CEUR, 2004.

[13]. Wang, J., Han, J., Li, C.: Frequent closed sequence mining without candidate maintenance. IEEE Trans. on Knowledge Data Engineering 19(8), 10421056 (2007)

[14]. Yun, U., Ryang, H., Ryu, K. H.: 'High utility itemset mining with techniques for reducing overestimated utilities and pruning candidates'. Expert Syst. with Appl. 41(8), 3861{3878 (2014)}

[15]. Zida, S., Fournier-Viger, P.,Wu, C.-W., Lin, J. C. W., Tseng, V.S.: Efficient mining of high utility sequential rules. In: Proc. 11th Intern. Conf. Machine Learning and Data Mining (MLDM 2015), pp. 1{15 (2015)}

[16]. http://www.philippe-fournier-viger.com.