A Comparative Analysis of Various Algorithms for High Utility Itemset Mining

Mansi Jaiswal[1], Vijay Prakash[2]

[1]PG Student, CSE Department, Shri Vaishnav Vidhyapeeth Vishwavidyalaya , Indore

[2]Assistant Professor, CSE Department, Shri Vaishnav Vidhyapeeth Vishwavidyalaya, Indore

Abstract: ***Frequent pattern mining has been an important topic since the concept of frequent itemsets was first introduced by Agrawal et al [6]. Given a dataset of transactions, frequent pattern mining finds the itemsets whose support (i.e. the percentage of transactions containing the itemset) is no less than a given minimum support threshold. However, neither the number of occurrences of an item in a transaction, nor the importance of an item, is considered in frequent pattern mining. Itemsets with more occurrences or importance may be more interesting to users, since they may bring more profit. In light of this, high utility itemset mining has been studied [9, 15, 42, 35]. In high utility itemset mining, the term utility refers to the importance of an itemset; e.g., the total profit the itemset brings. An itemset is a High Utility Itemset (HUI) if the utility of the itemset is no less than a given minimum threshold. High utility itemset mining focuses more on the utility values in the dataset, which are usually related to profits for the business. Such utilities are interesting to the business owners, who could gain more profits from them. For example, supermarkets use frequent itemset mining to find merchandises customers usually buy together, so as to make recommendations to customers. However, with high utility itemset mining, supermarkets will be able to recommend not only the merchandises people usually buy together, but also the merchandises which will lead to more profits for the store.1 Most of the frequent pattern mining algorithms prune off itemsets in an early stage based on the popular Apriori property [8]: every sub-pattern of a frequent pattern must be frequent (also called the downward closure property). However, this property does not hold in high utility itemset mining, which makes mining high utility itemsets more challenging. The state-of-the-art approaches achieve good performance when the dataset is relatively small. However, the volume of data can grow so faster than expected, that a single machine may not be able to handle a very large amount of data.***

Keywords: RUP/FRUP-GROWTH algorithm , HUI , data mining , apriori , big data

1. Introduction

The rapid growth of data generated and stored has led us to the new era of Big Data [3, 4, 14, 18, 19]. Nowadays, we are surrounded by different types of big data, such as enterprise data, sensor data, machine-generated data and social data. Extracting valuable information and insightful knowledge from big data has become an urgent need in many disciplines. In view of this, big data analytics [3, 4, 14, 18, 19] has emerged as a novel topic in recent years. This technology is particularly important to enterprises and business organizations because it can help them to increase revenues, retain customers and make more intelligent decisions. Due to its high impact in many areas, more and more systems and analytical tools have been developed for big data analytics, such as Apache Mahout [14], MOA [3], SAMOA [19] and Vowpal Wabbit [20]. However, to the best of our knowledge, no existing studies have incorporated the concept of utility mining [2, 6, 7, 8, 11, 12, 13] into big data analytics.

Utility mining is an important research topic in data mining. The main objective of utility mining is to extract valuable and useful information from data by considering profit, quantity, cost or other user preferences. High utility itemset (HUI) mining is one of the most important tasks in utility mining, which can be used to discover sets of items carrying high utilities (e.g., high profits). This technology has been applied to many applications such as market analysis, web mining, mobile computing and even bioinformatics. Due to its wide range of applications, many studies [2, 6, 7, 8, 11, 12, 13] have been proposed for mining HUIs in databases. However, most of them assume that data are stored in centralized databases with a single machine performing the mining tasks. However, in big data environments, data may be originated from different sources and highly distributed. A large volume of data also makes it difficult to be moved to a centralized database. Thus, existing algorithms are not suitable for the applications of big data. Although mining HUIs from big data is very desirable for many applications, it is a challenging task due to the following problems posed: First, due to a large amount of transactions and varied items in big data, it would face the large search space and the combination explosion problem. This leads the mining task to suffer from very expensive computational costs in practical. Second, pruning the search space in HUI mining is more difficult than that in frequent pattern mining because the downward closure property [1] does not hold for the utility of itemsets. Therefore, many search space pruning techniques developed for frequent pattern mining cannot be directly transferred to the scenario of HUI mining. Third, a large amount of data cannot be efficiently processed by a single machine. A well-designed algorithm incorporated with parallel programming architecture is needed. However, implementing a parallel algorithm involves several problematic issues, such as search space decomposition, avoidance of duplicating works, minimization of synchronization and communication overheads, fault tolerance and scalability problems.

2. Literature Review

In 2017 , Jue Jin and Shui Wang in their research work titled “RUP/FRUP-GROWTH: AN EFFICIENT ALGORITHM FOR MINING HIGH UTILITY ITEMSETS” proposed an improvement of the UP-Growth algorithm called RUP-

Growth, and develops a new algorithm called FRUP-Growth to take into consideration both the minimum support number and the minimum utility value to mine frequent & high utility itemsets. The experimental results show that their proposed strategies are more efficient and effective; especially with real-life marketing database, the advantage is more obvious.

In 2015, Ying Chun Lin and others in their research work titled "Mining High Utility Itemsets in Big Data" propose a new framework for mining high utility itemsets in big data. A novel algorithm named PHUI-Growth (Parallel mining High Utility Itemsets by pattern-Growth) is proposed for parallel mining HUIs on Hadoop platform, which inherits several nice properties of Hadoop, including easy deployment, fault recovery, low communication overheads and high scalability. Moreover, it adopts the MapReduce architecture to partition the whole mining tasks into smaller independent subtasks and uses Hadoop distributed file system to manage distributed data so that it allows to parallel discover HUIs from distributed data across multiple commodity computers in a reliable, fault tolerance manner.

In 2012, Adinarayanareddy B and others in their research work titled "An Improved UP-Growth High Utility Itemset Mining" adopted UP-Tree (Utility Pattern Tree), which scans 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 this paper presents modified algorithm aiming to reduce the execution time by effectively identifying high utility itemsets.

In 2014 , Junqiang Liu and others in their research work titled "Direct Discovery of High Utility Itemsets without Candidate Generation" proposed a high utility itemset growth approach that works in a single phase without generating candidates. Their basic approach is to enumerate itemsets by prefix extensions, to prune search space by utility upper bounding, and to maintain original utility information in the mining process by a novel data structure. Such a data structure enables them to compute a tight bound for powerful pruning and to directly identify high utility itemsets in an efficient and scalable way. We further enhance the efficiency significantly by introducing recursive irrelevant item filtering with sparse data, and a lookahead strategy with dense data. Extensive experiments on sparse and dense, synthetic and real data suggest that their algorithm outperforms the state-of-the-art algorithms over one order of magnitude.

In 2007, Alva Erwin and others in their research work titled "A Bottom-Up Projection Based Algorithm for Mining High Utility Itemsets" proposed a new algorithm called CTU-PRO that mines high utility itemsets by bottom up traversal of a compressed utility pattern (CUP) tree. They have tested the algorithm on several sparse and dense data sets, comparing it with the recent algorithms for High Utility Itemset Mining and the results show that their algorithm works more efficiently.

3. RUP/FRUP Growth Algorithm

Mining frequent itemsets from a transaction database is an important task in the field of data mining. Its goal is to identify the itemsets with their appearing frequencies above a certain threshold. Rakesh Agrawal developed the first frequent itemset mining algorithm, named Apriori [1], for mining association rules from sales data in 1994. Since then, new algorithms are proposed constantly, such as FP-Growth [2], COFI [3], COFI2 [4], Pincer-search [5], MAFIA [6], CLOSET [7], CHARM [8], and so on. The existing algorithms of mining frequent itemsets only consider each item in a transaction database as a 0/1 value. Moreover, items having high/low selling frequencies may have low/high profits, respectively. The information of profit and quantity of each item in a transaction itemset is of great importance for market analysis. In view of this, high utility itemsets mining emerges as an important topic in data mining. Yao et al. proposed the high utility itemsets mining model [9] in 2004. They defined two types of utilities for items: internal utility and external utility. The internal utility of an item in a transaction is defined according to the information stored in the transaction itself, such as quantity of the merchandise. The external utility of an item is based on information not available in the transaction database, for example, the profit value of the merchandise sold in the marketplace. The utility of an item is defined as its internal utility multiplied by its external utility. The utility of an itemset is defined as the sum of its all items' utilities. An itemset X is a high utility itemset if its utility is not less than a user-specified minimum threshold. High utility itemsets mining is widely used in applications such as sales data analysis, etc., to find the suitable combinations of items that are more profitable; and many algorithms have been proposed recently, such as those described in [9-17]. The existing algorithms mentioned above apply overestimated method; they firstly find candidates for high utility itemsets, and then identify the high utility itemsets from the candidates by one additional database scan. The performance bottleneck of these algorithms is the generating & processing of the candidate itemsets; and with the increasing of the number of long transaction itemsets and the decreasing of the minimum utility threshold, the situation may become worse. Additionally, as stated above, the existing algorithms mine either frequent itemsets or high utility itemsets. However, in real world, an itemset may be a high utility itemset, but not a frequent itemset, so we can not get an association rule from this itemset, and can not get useful knowledge for the future business arrangement. Thus, in this paper, we firstly propose new strategies to reduce the number of candidates; then propose a new idea to mine frequent & high utility itemsets, which satisfy both the minimum support threshold as well as the minimum utility threshold, from transaction datasets.

4. FP Growth Algorithm

The essential theorem of mathematics says that every positive integer has a completely unique high factorization. What the FP-growth does is getting a not unusual suffix after which extracts all viable prefixes and after joining them to the suffix a common pattern is created. In the FP-growth set of rules it isn't always crucial that we are searching out all common patterns cease to a selected suffix like “I5” or we want to extract all the frequent patterns. In contrast with FPincrease the FPPF for mining of all frequent styles cease to a specific suffix like “I5”, does no longer create complete of the tree and just makes a speciality of prefixes related to that specific suffix. Without producing a tree, our set of rules referred to as common sample-high aspect (FPPF) extracts the common prefixes and generates the common itemset which ends up with that suffix. In table three all of the used symbols and acronyms which might be used in this section are supplied. The following affords a few primitive definitions which can be important to clarify the frequent sample mining hassle.

There are four predominant methods used for mining high software itemsets from transactional databases which can be given as follows: 5.1. Data Structure A compact tree structure, UP-Tree, is used for facilitate the mining performance and keep away from scanning authentic database repeatedly. it will additionally hold the transactions information’s and high application itemsets. 5.2. UP-boom Mining technique After creation of worldwide UP tree, mining UP-Tree through FP- increase for producing PHUIs will generate so many candidates with a view to avoid that UP-increase technique is used with two strategies: One is discarding unpromising objects in the course of building a neighborhood UP-Tree. another is discarding local node utilities.

Preceding studies, problems in this section arise: 1) number of HTWUIs is too big; and (2) scanning original database could be very time eating. In our framework, overrated utilities of PHUIs are smaller than or same to TWUs of HTWUIs for the reason that they're decreased via the proposed strategies. as a result, the quantity of PHUIs is a good deal smaller than that of HTWUIs. therefore, in segment II, our technique is tons green than the preceding techniques. moreover, although our methods generate fewer candidates.

5. Problem Formulation

It has been seen in the literature review that there are several approaches for mining the high utlity itemsets namely :-

1) RUP/FRUP-GROWTH algorithm
2) PHUI growth algorithm.
3) Direct discovery algorithm without candidate generation.
4) CTU-PRO algorithm.

The major research lacuna observed is that these algorithm have been compared with one or two algorithm and never been compared in totality. Also the above algorithms have not been tested by a same dataset and thus their performance can’t be compared until a single dataset based comparison is not carried out.

6. Research Objective

1) To understand and implement the above mentioned four algorithms using a suitable platform. We would be using Matlab parallel processing toolbox in conjunction with Hadoop to analyze the said algorithms.
2) To compare the above algorithms using a single dataset so that their performance can be evaluated exhaustively.

7. Conclusion

Most of research on high utility itemset focuses on static databases (eg. Transaction database). With the emergence of the new application, the data processed may be in the continuous dynamic data streams. Because the data in streams come with high speed and are continuous and unbounded, mining result should be generated as fast as possible and make only one pass over a data. In this paper, we have proposed two algorithms named UP- Growth and UP-Growth+ 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 UPGrowth+ , outperform the state of-the-art algorithms substantially especially when databases contain lots of long transactions or a low minimum utility threshold is used. Proposed system Iishaving applications in Website click stream analysis, Business promotion in chain hypermarkets, Cross marketing in retail stores, online e-commerce management, Mobile commerce environment planning and even finding important patterns in biomedical applications. In this Paper we have presented a review on various algorithms, work, idea and limitations of different methods for high utility Itemset mining using a transaction dataset.

References

[1] Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Database. In: ACM SIGMOD International Conference on Management of Data (1993) .

[2] A. Erwin, R.P. Gopalan, and N.R. Achuthan,“Efficient Mining of High Utility Itemsets from Large Datasets”, T. Washio et al. (Eds.):PAKDD2008, LNAI 5012, pp. 554–561, 2008. © Springer-Verlag Berlin Heidelberg 2008.

[3] Yao, H., Hamilton, H.J., Buzz, C. J., “A Foundational Approach to Mining Itemset Utilities from Databases”, In: 4th SIAM International Conference on Data Mining, Florida USA (2004).

[4] “A Two-Phase Algorithm for Fast Discovery of High Utility Itemsets”, Ying Liu, Wei-Keng Liao, and Alok Choudhary, Northwestern University, Evans.

[5] “CTU-Mine: An Efficient High Utility Itemset Mining Algorithm Using the Pattern Growth Approach” In: Seventh International Conference on Computer and Information Technology (2007).

[6] “UP-Growth: An Efficient Algorithm or High Utility Itemset Mining ”, Vincent S. Tseng, Cheng-Wei Wu, Bai-En Shie, and Philip S. Yu. University of Illinois at Chicago, Chicago, Illinois, USA, 2010.

[7] Mengchi Liu Junfeng Qu, “Mining High Utility Itemsets without Candidate Generation”, 2012.

[8] "FHM: Faster High-Utility Itemset Mining using Estimated Utility Co-occurrence Pruning", Philippe Fournier-Viger1, Cheng-Wei Wu 2014.

[9] Smita R. Londhe,, Rupali A. Mahajan,, Bhagyashree J. Bhoyar,"Overview on Methods for Mining High Utility Itemset from Transactional Database", International Journal of Scientific Engineering and Research (IJSER), Volume 1 Issue 4,December2013.

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

[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] Sucahyo, Y.G., Gopalan, R.P., CT-PRO: "A BottomUp Non Recursive Frequent Itemset Mining Algorithm Using Compressed FP-Tree Data Structure", In: IEEE ICDM Workshop on Frequent Itemset Mining Implementation (FIMI), Brighton UK (2004).

[13] G. Salton, Automatic Text Processing, AddisonWesley Publishing, 1989.

[14] J. Pei, J. Han, L.V.S. Lakshmanan, "Pushing convertible constraints in frequent itemset mining", Data Mining and Knowledge Discovery 8 (3) (2004) 227–252.

[15] A. Erwin, R.P. Gopalan, and N.R. Achuthan, "Efficient Mining of High Utility Itemsets from Large Datasets", T. Washio et al. (Eds.): PAKDD 2008, LNAI 5012, pp. 554–561, 2008. © SpringerVerlag Berlin Heidelberg 2008.

[16] Bin Chen, Peter Hass, Peter Scheuermann, "A New Two-Phase Sampling Based Algorithm for Discovering Association Rules", SIGKDD '02 Edmonton, Albetra, Canada © 2002 ACM 1 58113 567 X/02/2007.

[17] Ming-Yen lin, Tzer-Fu Tu, Sue-Chen Hsueh, "High utility pattern mining using the maximal itemset property and, lexicographic tree structures", Information Science 215(2012) 1-14.

[18] Sudip Bhattacharya, Deepty Dubey, "High utility itemset mining, International Journal of Emerging Technology and advanced Engineering", ISSN 2250-2459, Volume 2, issue 8, August 2012.