© UNIVERSAL RESEARCH REPORTS | REFEREED | PEER REVIEWED
ISSN : 2348 - 5612 | Volume : 05 , Issue : 05 | April 2018

A Survey on Scalable and Parallel High Utility Itemset Mining

[1]*Sandeep Dalal, Maharshi Dayanand University Rohtak, Haryana, Email- sandeepdalal.80@gmail.com*
[2]*Vandna Dahiya, Maharshi Dayanand University Rohtak, Haryana, Email- vandanadahiya2010@gmail.com*

Abstract

Utility itemset mining aims to the discovery of itemsets or patterns with stimulating interest. While frequent itemset or pattern mining finds the interesting patterns based on the occurrence frequency of a pattern, utility itemset mining (UIM) is a further development in this field. It integrates the aspect of utility in some form like weight, cost, amount, profit or any other factor of interest. Utility mining is thus an objective-oriented approach, which aims to find the patterns with a high utility such as more profit or low cost/side-effect etc. However, utility mining is a complex process than frequency itemset mining (FIM) as anti-monotonicity property does not hold for the itemsets like as in FIM. Many algorithms have been developed to mine the itemsets with utility information in recent years. But most of them are not scalable for the nature of data with which we deal nowadays, called big data. This paper focuses on reviewing the recent advances in the field of high utility pattern mining for large datasets with scalable and parallel processing algorithms. The paper is concluded with open problems and future directions for research in the arena of big data.

Keywords: frequent pattern mining, association rules, high-utility item set mining, big data, interestingness measures.

1. Introduction

Frequent pattern mining (FMP) is a significant area in data mining that uncovers the interesting patterns in the large databases. It discovers the set of items that occurs frequently in a dataset and this knowledge can be used in different application areas such as market dataset analysis, indexing, and retrievals, detection of software bugs, web link analysis, etc. However, frequent pattern mining considers only the presence or absence of an item in a transaction. It reveals the patterns, which appear more than the user-specified support count. It may generate a large number of patterns, which are frequent but having low revenue; and may lose the information for valuable patterns, which are less frequent. In real life, the traders are concerned in business of the item sets, which give more profits, while the frequent pattern mining generates only item sets with high frequency without considering the profit. Hence, the traditional framework of FPM cannot satisfy the prerequisite of users who wish to find patterns with high utilities such as huge profits. To tackle these issues, utility pattern mining (abbreviated as UPM) emerges as a significant area in data mining. In utility mining, every item has some kind of utility factor associated with it, such as unit profit, cost or any other objective of interest and the item can appear in a transaction more than once. Thus, mining high utility patterns can be defined as finding all the patterns in a dataset with utility no less than the minimum specified utility threshold. Sequences containing such information are often encountered in real-life applications. For instance, in customer behavior analysis, a complex event sequence represents the shopping behavior of a customer. Now, with the vast amount of data, and with varied items in a database, the search space is very large with the explosion of combinations of items. It is very difficult and inefficient to mine the itemsets on a single machine. Scalable and parallel algorithms are needed to mine such data. In this paper, the recent advances have been reviewed to mine big data for HUIM with scalable and parallel algorithms.

In section 2, an overview of the problem of high utility itemset mining has been presented. In section

3, a survey has been done for existing serial and parallel algorithms for HUIM along with big data paradigm. Various challenges to mine the itemsets from big data has been explored in section 4. Lastly, section 5 draws a conclusion.

2. High Utility Itemset Mining

The problem of high-utility itemset mining can be seen as an extension of frequent pattern mining. Frequent pattern mining is a popular problem in data mining, which consists of finding frequent patterns in transaction databases. But these patterns are not sufficient enough to be used practically for business purposes. Because many patterns can be frequent but not related to the needs of the business. So, HUIM incorporates an objective apart from frequency, which can be taken in the form of profit, cost, user's interest, etc. HUIM can then be defined as mining the patterns with utility, which is no less than a minimum utility specified by the user.

Consider the following transaction database with a set of transactions made by the customers. A transaction database contains transactions where purchase quantities are taken into account as well as the unit profit of each item. For example, in the following database, the first customer buys items "α", "β", "γ", "δ" and "ϵ", while the second one buys items "β", "γ", "δ" and "ϵ".

Table 1 Transaction Database

Transaction No	Quantity of Transaction
T0	{α (2), β (4), γ (1), δ (5), ϵ (1)}
T1	{β (1), γ (2), δ (3), ϵ (1)}
T2	{α (1), γ (2), δ (1)}
T3	{α (2), γ (6), ϵ (4)}
T4	{β (2), γ (5), ϵ (1)}

Table 2 Unit Profits of Items

Item	Price of Each Item
α	10
β	2
γ	5
δ	4
ϵ	6

Consider transaction T3. It indicates that the corresponding customer has bought two units of item "α", six units of item "γ", and four units of item "ϵ". Now, lets see table 2. This table indicates the profit of each item for one unit. For example, the unit profit of items "α", "β", "γ", "δ" and "ϵ" are respectively 10, 2, 5, 4, and 6. This means that each unit of "α" that is sold generates a profit of 10.

The problem of high utility itemset mining is to discover the group of items that are bought together and give a high profit. A minimum value called minimum utility threshold is specified by the user to find all the item sets with utility at least greater than this minimum value.

3. Literature Survey

Various algorithms have been presented for HUIM, which can address the problem of mining the big data. They can be classified into two categories. Scalable Serial algorithms and Parallel Algorithms.

Scalable Serial Algorithms

Serial algorithms have been developed by many researchers to mine the high utility itemsets, which focus mainly on the challenges of memory. Because the memory constraint is the key issue for the larger datasets. CTU-PROL (Erwin et al., 2008) has been presented, which mines the itemsets of high utility using the method of pattern-growth. Transaction Weighted Utility (TWU) is calculated for the items and then itemsets are found with TWU no less than a minimum specified value. A compressed tree structure called CUP-Tree is created to mine the itemset, if the database is small. But if the database is large to fit in memory, parallel projections are created for the database and the algorithm works in parallel mode. Another scalable algorithm was presented to mine the HUIMs based on two-phase and TWU model, (Yin et al., 2008). The algorithm is a hybrid method, which divided the problem into smaller tasks and each task can run in parallel independently with different algorithms. This is an optimized method with less computation time but due to two phases, it suffers from the problem of generating a huge set of candidates. Another algorithm EFIM, (Zida et al., 2017), addresses the problem of two phases and ensure linear space and time for all the operations of an itemset. Identical transactions are merged and an efficient array-like data structure is used to store the itemsets.

Another algorithm was proposed (Ahmed et al., 2009) for incremental and interactive data, which could handle high volume and high velocity at the same time. Based on the property of ‘build once mine many’ three different tree structures were proposed, which can efficiently store the utility information. The problem of candidate generation and then test has been removed in this approach. The algorithm THUI-Mine (Tseng et al., 2006) was proposed which can handle data streams as well. It combines the advantages of two-phase and incremental mining algorithms and can work on temporal databases. It mines the HUIs locally for the current time frame and for the next time frame. It discovers length-2 candidates and then use these candidates to generate all the further candidates. It thus reduces the database scans and the overall time of the algorithm. Another algorithms, MHUI-BIT and MHUI-TID were proposed based on THUI-Mine for the data streams (Li et al., 2008).

Another framework, GUIDE was proposed for streams (Shie et al., 2012). It can work for three different models-landmark, sliding and time fading. A MUI-tree is constructed in a single scan of database which stores the information based on the type of model selected. HUPMS algorithm (Ahmed et al., 2012), also works for stream data and is based on TWU model. An incremental and interactive utility tree called HUS-tree was proposed for stream data to keep the information of utilities. HUS-tree is lexicographic tree and it sorts items in a lexicographic order. The nodes store id of item and TWU information to update the utility information. After processing all the items of the current window, it scans the data one more time to find the exact utility of the candidates and discovers HUIs.

For streaming algorithms, it is difficult to specify the minimum utility threshold in advance because at least a small set of items need to be analyzed before setting the minimum utility. Also, the HUIs may change with time in dynamic databases. To address this problem, algorithm with top-k HUIs have

been developed such as T-HUDS (Zihayat & An, 2014), where the value of minimum utility can be adjusted dynamically.

Distributed Processing Algorithms

Distributed algorithms use the paradigm of distributed computing where there is one master node and some slave nodes. Local processing is done on slave nodes and then the message is passed to master nodes. One such algorithm is (Vo et al., 2009), where the authors have presented a parallel method, based on master-slave architecture. The database is first divided into slave nodes, which then calculate local utilities of itemsets. The master considers only those itemsets, which occur on at least two slaves. The communication time is an overhead in this approach.

The recent paradigm is based on Map-Reduce framework to mine the HUIs in big data using the Apache Hadoop. Hadoop consists of two main components – Distributed File System, which stores the data in distributed blocks and MapReduce, which is the processing unit.

A parallel method called as PHUI-Growth was proposed which is based on Apriori algorithm (Lin et al., 2015). The general processing of the algorithm is shown in figure 1. The algorithm runs in two phases. In phase 1, TWU of all the itemsets are calculated using the MapReduce. Then, all the itemsets, which are having a value less than the TWU are pruned from the search space. Another method called as approximate HUIM was proposed, which is a parallel form of HUI-Miner. (Chen & An, 2016). The sampling method was used to decrease the size of database and approximate HUIs are found out. PHUI-Miner thus gives a tradeoff between accuracy and time-memory complexity of the algorithm. Hadoop based algorithms are suffering from the drawback of MapReduce framework that requires each mapper to be followed by a reducer. Each pair has to read and write data from the disks, which takes a lot of time and degrade the performance of the algorithm.

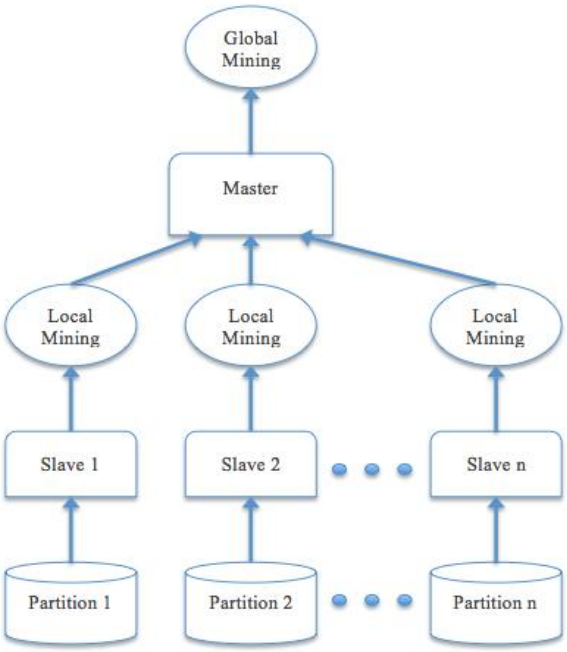

Figure 1 The General Framework of HUI Mining from Distributed Databases

4. Challenges of HUI Mining in Big Data

There are two main challenges in big data mining. The one is the volume of data and the other is computation time for such huge data. Many parallel algorithms have been developed for mining HUIs

in a big data environment. There are several issues related to parallel processing of the algorithms and the problem of HUIM from big data is an open problem. The challenges that need to be addressed to efficiently mine the itemsets from large datasbases can be described as follows:

Scalability: The algorithms are evaluated on datasets, which are not big enough to examine the real scalability of the algorithms.

Partition of Work and Load Balancing: One of the main challenges in parallel processing is to efficiently partition the load among all the nodes and estimation of resources required by each task. Dynamic balancing of the load is also important to re-distribute the load to the nodes, in case if some of the nodes have completed the task given to them.

Privacy: The algorithms for utility mining perform on personal data, preserving privacy is a very important factor lacking by the HUIM algorithms.

Complex Data: Most of the algorithms work on sequential and transaction data. Several applications can be developed if HUIs can be discovered from complex data such as graphs, time series data, etc.

Other Issues: There are various other issues related to HUIM in big data. Real big data is mostly noisy and with missing values. Data quality is of chief importance for many domains. Visualization of the pattern of big data is also challenging.

5. Conclusion and Future Works

An overview of scalable serial and parallel algorithms have been presented in this paper to mine the high utility itemsets from large databases. Various challenges related to the processing of parallel algorithms have also been discussed. It has been found that most of the algorithm focuses on memory scalability, which is a major concern during mining. Future work may incorporate the development of parallel algorithms that can address the open challenges shown in this paper.

References

1. Grama, A.: Introduction to Parallel Computing. Pearson Education (2003)
2. Spark, A. : Apache spark : lightning-fast cluster computing (2016)
3. Chen, Y., An, A.: Approximate parallel high utility itemset mining. Big Data Res. 6(Supplement C), 26–42 (2016). https://doi.org/10.1016/j.bdr.2016.07.001
4. Zihayat, M., An, A.: Mining top-k high utility patterns over data streams. Inf. Sci. 285, 138–161 (2014)
5. Tseng, V. S., Chu, C. J., Liang, T. : Efficient mining of temporal high-utility itemsets from data streams. In: ACM KDD Utility Based Data Mining, pp. 18–27 (2006)
6. Lin, Y.C., Wu, C.W., Tseng, V.S.: Mining High Utility Itemsets in Big Data, pp. 649–661. Springer International Publishing, Cham (2015)
7. Zihayat, M., Wu, C.W., An, A., Tseng, V.S.: Mining high utility sequential patterns from evolving data streams. In: ASE BD&SI 2015, pp. 52:1–52:6 (2015)
8. Vo, B., Nguyen, H., Ho, T.B., Le, B.: Parallel Method for Mining High Utility Itemsets from Vertically Partitioned Distributed Databases, pp. 251–260. Springer, Berlin (2009)
9. Kashyap, H., Ahmed, H.A., Hoque, N., Roy, S., Bhattacharyya, D.K.: Big data analytics in bioinformatics: a machine learning perspective. (2015). http://arxiv.org/abs/1506.05101
10. Li, H.F., Huang, H.Y., Chen, Y.C., Liu, Y.J., Lee, S.Y.: Fast and memory efficient mining of high utility itemsets in data streams. In: Proceedings of the 8th IEEE International Conference on Data Mining, pp. 881–886 (2008)

11. Mooney, C.H., Roddick, J.F.: Sequential pattern mining approaches and algorithms. ACM Comput. Surv. 45(2), 19:1–19:39 (2013)
12. Yu, G., Li, K., Shao, S.: Mining high utility itemsets in large high dimensional data. In: First International Workshop on Knowledge Discovery and Data Mining (WKDD), pp. 17–20 (2008). https://doi.org/10.1109/WKDD.2008.64
13. Subramanian, K., Kandhasamy, P., Subramanian, S.: A novel approach to extract high utility itemsets from distributed databases. Comput. Inform. 31(6+), 1597–1615 (2013)
14. Szlichta, J., Godfrey, P., Golab, L., Kargar, M., Srivastava, D .: Effective and complete discovery of order dependencies via set-based axiomatization. In: Proceedings of the VLDB Endowment, SEP vol. 10, no. 7, pp. 721–732 (2017)
15. Zida, S., Fournier-Viger, P., Lin, J.C.W., Wu, C.W., Tseng, V.S.: Efim: a fast and memory efficient algorithm for high-utility itemset mining. Knowl. Inf. Syst. 51(2), 595–625 (2017). https://doi.org/10.1007/s10115-016-0986-0
16. Shie, B. E., Yu, P.S., Tseng, V. S.: Efficient algorithms for mining maximal high utility itemsets from data streams with different models. Expert Syst. Appl. 39, 12947–12960 (2012)
17. Chan, R., Yang, Q., Shen, Y.: Mining high-utility itemsets. In: Proceedings of Third IEEE International Conference on Data Mining, pp. 19–26 (2003)
18. Erwin, A., Gopalan, R. P., Achuthan, N. R.: Efficient mining of high utility itemsets from large datasets, pp. 554–561. Springer, Berlin (2008)
19. 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, 1708–1721 (2009)
20. Ahmed, C.F., Tanbeer, S.K., Jeong, B. S.: Interactive mining of high utility patterns over data streams. Expert Syst. Appl. 39, 11979–11991 (2012)
21. Dawar, S., Sharma, V., Goyal, V. : Mining top-k high-utility itemsets from a data stream under sliding window model. Appl. Intell. 47(4), 1240–1255 (2017)
22. Cao, L., Zhao, Y., Zhang, H., Luo, D., Zhang, C., Park, E. : Flexible frameworks for actionable knowledge discovery. IEEE Trans. Knowl. Data Eng. 22(9), 1299–1312 (2010)