International Journal of Engineering & Technology, 7 (1.2) (2018) 211-214

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Review on high utility itemset mining algorithms for big data

Sandeep Dalal *, Vandna Dahiya

Maharshi Dayanand University, Rohtak-124001, Haryana
**Corresponding author E-mail: sandeepdalal.80@gmail.com*

Abstract

High Utility Itemset (HUI) mining is a developing range over the field of information investigation. It plans to find the item sets that are having a base/most themost extreme utility indicated by the client as per its target. In utility mining, everything has a factor related with it, which speaks to its essentialness, as cost, some, weight, intriguing quality, benefit or some other data in view of the inclination of the client. Utility item set mining is an essential undertaking and is having a thorough scope of uses in the present period of huge information, for example, site click-stream examination, portable registering, cross showcasing in different fields like retail locations and so forth. The way toward mining HUIs is unique and complex than visit thing set mining as a portion of the properties of FIM does not hold for HUIs mining. Mining turns out to be drearier when the data set is vast as inquiry space is expanded. A few calculations for utility digging have been proposed for substantial data sets. These paper accentuations on auditing different philosophies for existing cutting-edge calculations for utility item set digging for vast data sets to lead away for the further research.

Keywords: *Utility Mining; Candidate Pruning; Frequent Item Set; Transaction Weighted Utilization; Big Data.*

1. Introduction

Data analytics for large data sets has emerged as a novel research area in the recent years. With the exponential rise in data, it is very crucial to efficiently mine this huge data to gain various insights of it. Various techniques like classification, clustering, regression, trend analysis, and association rule analysis are used to mine the data based on the type of knowledge we want to mine. Association Rule Mining (ARM) [1] is an important information mining technique, which is used to find the association between different items that occur simultaneously in a database. Rules are framed and then data is mined based on them. Frequent Itemset Mining (FIM) is a primary approach for ARM. The patterns that occur frequently in a database are discovered, and rules are generated from them. This is useful for many applications like market basket analysis, indexing and retrievals, web link analysis, etc.. However, it only includes the associations or patterns that are based on the frequency of occurrence. It might discover the item sets that are of less importance or utility and lose some of the item sets that are less frequent but with high utility factor. Utility here is taken by absence or presence of the item that is not the sufficient reason to reflect its importance. For example, selling a refrigerator is much more profitable than selling a glass bottle but a sequence having a refrigerator is quite less frequent than the one having a bottle. Another limitation of FIM is it considers an item only once in a transaction. To find various patterns in a database it is important to mind other important factors also in addition to the frequency such as profit, weight, interestingness, etc. This issue is addressed by Weighted Association Rule Mining (WARM), which explores the database by taking note of some kind of profit or weight of items. The profitable sequences of items address various business decisions like how to maximize the prophet, how to minimize the costs on advertising, etc.

Various algorithms have been proposed in this category of ARM, which mines the items of certain utility. Proposed by Hong Yao in 2003, Utility Itemset Mining (UIM) is an extension to FIM where each data item is associated with some utility factor depending on the specifications and requirements for the user. While FIM only gives statistical correlation between the items, UIM gives semantic significance. Thus, it incorporates both- internal utility (quantity) and external utility (Profit/objective). An item set now is called the item set of high utility only if it satisfies a user-defined pre value or threshold for minimum utility. The utility is calculated as the product of external and internal utilities or quantity and profit. Then, for a transaction, the utility of an item set is characterized as the sum of utilities of all items of the item set in that transaction. Thus UIM is an approach to mine the item sets based on the importance of items rather than their occurrence count. For example, in a retail store, buying frequency of milk and butter is more than buying of a microwave oven but the unit profit of the oven is much more than the combination of milk and butter. Utility item set mining is a vibrant area of data mining and has various applications in today's era of big data like episode mining, cross-marketing in retail stores, website click-stream analysis, mobile computing and biomedical applications in discovering various rare patterns of some disease, etc.

Mining of high-utility item sets is usually known as more complex and interesting than FIM because the Apriori property or downward closure property of FIM is not valid for HUIs. In HUIM, the utility factor is not anti-monotonic, i.e. an item set of high utility may or may not have the subset of high utility. It could possibly have subsets or super sets, which are having equal, lower or even higher utilities. So, the search space is larger here, as it can't be lessened using Apriori property. So, techniques that are used in FIM are not utilized directly for HUIM. In recent times, many algorithms have been developed for HUIM. However, most of them consider small data sets. This work reviews some of the best algorithms, their point of interest, limitations and various challenges in HUI mining.

2. Research problem

The traditional structure of FIM may find a lot of successive however low income designs and lose the data on profitable examples having low offering frequencies. Consequently, it can't fulfill the prerequisite of clients who want to find designs with high utilities, for example, high benefits. To address these issues, utility thing set mining develops as an essential research territory in information mining.

2.1. Problem definition

Various definitions are described here that are used in HUIM. Consider, I as the set of items that are distinct. I = {I_1, I_2, I_3…I_N} and X is an item set of some length n. Let D be the transactional database, D = {T_1, T_2, T_3…T_n}, where each transaction is a subset of, I. Each item in the transaction has some quantity or internal utility Q (I_s, T_r) and is linked to a factor such as profit, called as external utility P (I_s, D). Let us consider an example of transaction database as shown in the table-

Table 1: Transaction Database

Tid	Transactions	Transaction-Utility
1	(A:2), (C:1), (D:1)	9
2	(A:1), (C:3), (E:2), (G:5)	28
3	(A:2), (B:2), (D:6), (E:1), (F:2)	31
4	(B:4), (C:3), (D:3), (E:1)	20
5	(B:2), (C:2), (E:1), (F:1), (G:2)	16

Table 2: External Utility of Items

Item	A	B	C	D	E	F	G
Unit Profit	3	2	1	2	3	3	2

Definition 1: *Utility of a thing, it is the fundamental unit, meant as EU (Is, Tr) and can be characterized as P (Is, D) * Q (Is, Tr). Here, Utility for a thing an in exchange T1 is 3*2 = 6.*

Definition 2: *Utility of an itemset in an exchange, it can be characterized as EU(X, Tr) = □Is □ X (Is, Tr).*

Utility of an itemset {A, C} in T1 is 3*2 + 1*1 = 7.

Definition 3: *Transaction utility and aggregate utility, the exchange utility (TU) of an exchange Tr is characterized as TU (Tr) = EU (Tr, Tr). The aggregate utility of a database D is indicated as Total UDB and characterized as □Tr□D TU(Tr), Here, exchange utility of T1 is (2*3+1*1+1*2) = 9.*

Definition 4: *Utility of an itemset in a database, the utility of an itemset X is characterized as EU(X) = □Tr□D□X□Tr(X, Tr), Utility for itemset {BD} is 30.*

Definition 5: High utility itemset, an itemset X is named as high utility itemset (HUI) if and just if the utility of itemset X, U(X) is no not as much as a base utility edge indicated by the client, min_util something else, X is an itemset of low utility. Let, the edge for utility is 34. At that point, HUIs are {BCD} =34, {BDE} =36, {BCDE} =40.

Definition 6: *Transaction-weighted use, TWU, for an itemset X is characterized as the entirety of the exchange utilities of the considerable number of exchanges which are having itemset X, is* $TWU(X) = \sum_{X \subseteq Tr \wedge Tr \in DTU}(T_r)$ *Now, TWU of {G} =* $TU(T_2) + TU(T_5) = 44$.

So, the problem statement is to find the desired number of HUIs or top-k HUIs from a given database or sets of databases.

3. Related work

In this section, a review of various algorithms, their methodologies, scope and limitations has been presented for HUIM.

3.1. Various methodologies for HUI mining

Various studies on mining have been carried out in the past for item sets or patterns discovery. Based on them, several techniques have been proposed for mining of item sets with desired frequency and utility. Some noticeable algorithms are HUI-Miner, TWU, FHM, UP- Growth, UP-Growth+, Two-Phase, etc. This section provides related works about HUIs mining.

Raymond Chan et al. [2], (2003) gave the concept of high utility item sets. An algorithm was proposed to mine the item sets based on various factors other than frequency of occurring only. A new pruning strategy is used other than non-monotonic apriori. No minimum threshold is specified, and the top-k item sets are mined directly to support a business objective.

Ying Liu et al. [40], (2005) proposed a Two-Phase algorithm which routes in two phases. This algorithm uses the downward closure property of FIM for transaction weighted utilization item set. This property lower down the number of candidate sets. The database is scanned in the first phase and for each transaction, transaction weighted utility is calculated and the candidates who are above the threshold values are taken into concern. It reduces the search space. In phase two of the algorithm, scanning is done for the database once again to find the high utility item set by calculating the utilities. Candidate generation is too high in this algorithm. This algorithm also simplifies the computation for calculating the utilities.

Alva Erwin et al. [3], (2008) proposed CTU-Mine where only one database scan is sufficient and pattern growth approach is used. However, the overall algorithm is complex because of using the tree structure. Disk storage scheme is used to store the intermediate candidate whenever memory is inadequate for dealing with large databases.

Hsin Yun et al. [4], (2008) proposed two memory base algorithms called as MHUI-BIT and MHUI-TID for data streams. Items are represented using bit-vector, and TIDlist based approaches. Sliding window scheme is used for transactions. When the sliding window is full, a lexicographical tree summary is generated base on deletion on insertion of items. It is a one pass algorithm and thus very efficient. Mining with constraints from data stream is a further research issue.

Mengchi Liu et al. [40], (2012) proposed HUI-Miner, which is a single-phase algorithm. It uses a data structure called utility list, which maintains the utility information of items and pruning information. It uses the depth first approach. Utility lists of 2-itemsets are constructed based on the intersection of utility list of single items. Further, utility list of higher order is generated so on from lower order utility-lists. It uses a vertical representation of mining HUIs where the correlation among the items is calculated with 'join' operation. It is time bearing and very costly.

D. Cheung et al. [5], (2013) proposed the algorithm FUP or Fast Update algorithm for dynamic databases where new tuples are being inserted. The framework of mining is incremental. FUP2 was also proposed later on, which also handles deletion in addition to addition of tuples in the database.

Sandy Moens et al. [6], (2013) proposed two different algorithms for item set mining- one is Dist-Eclat, and the other is a hybrid approach- BigFIM. Dist-Eclat is a fast algorithm, and its main focus is speed by balancing the load. A version of apriori is used in BigFIM to mine the very large-size databases and big data using various mappers. It uses a hybrid approach for mining. Firstly, Eclat is used to mine the frequent item sets. They are then distributed to the mappers. The overall distribution of the workload is balanced. However, the number of intermediate candidates are very high. So, this is a further area of research to balance the load without generating a big range of intermediate candidates.

Junfu Yin et al. [7], (2013) proposed an algorithm for mining the sequences called as Top-k high Utility Sequence (TUS). It mines top-k sequences without specifying minimum thresholds. Various strategies for pruning the databases are introduced for filtering the database.

Vincent S. et al. [11], (2013) proposed UP-Growth and UP-Growth+ where a global utility pattern tree construction is done for high-potential item sets. Two scans of the database are required. Various strategies are applied to decrease the candidates like locally discarding the unpromising items, decreasing local utilities of the node during the construction of the global utility pattern trees.

Philippe Fournier Viger et al. [12, 40], (2014) proposed an algorithm FHM, which is an extension to HUI-Miner. HUI-Miner has to perform the join operation, which is a costly operator in terms of time and space. FHM stands for Fast High-Utility Miner. This is also a two-phase algorithm, which scans the database twice. A new strategy of estimated utility co-occurrence pruning is used, which reduces the join operations by up to 95% and thus, FHM is faster than HUIM. For every pair of items, FHM maintains transaction-weighted utilization. TWU for each item is calculated in the first database scan and during the second scan, it sorts only those items of the transactions, which have TWU greater than the minimum specified utility. Different levels of the tree follow depth-first search and time complexity is decreased by 6 times.

Jerry Chun Wei LIN et al. [13], (2015) proposed an algorithm based on binary particle swarm optimization to efficiently mine high utility itemsets (HUIs). The discrete mechanism is used, and the size of each particle is set to 1. They are called as high TWI 1-item sets. Sigmoid function is a used and maximal pattern tree is designed to generate the valid combinations of the candidates. It reduces the need of multiple data scans. This algorithm is superior to many GA-based algorithms for mining HUIs in terms of execution time, and quality and completeness of the item sets produced.

Cheng-Wei Wu et al. [19], (2015) proposed a novel algorithm PHUI-Growth for efficient mining of high utility item sets from the data that is distributed across various sites. The Hadoop [38] framework is used to parallel mine the data. The advantage of this framework is fault tolerance, high scalability and low communications overhead. The whole mining task is divided into subtasks. HDFS is used to manage distributed data in a reliable manner.

Vikram Goyal et al. [33], (2015) proposed an algorithm called as UP-Rare Growth. This algorithm uses a data structure called UP-tree. The tree is created in two steps by scanning the database twice. Firstly, the TWU value is computed, and the least promising items are removed. Then transactions are organized in order of their decreasing TWU value. Each transaction is then processed. Thus avoiding the less useful branches of the tree reduces the searching space.

Unil Yun et al. [36], (2015) proposed an incremental tree based structure HUPID-tree and proposed the algorithm HUPID-growth for mining high utility incremental patterns. Also, a novel data structure called as Tail-node Information List or TIList is used to efficiently process the incremental database.

Quang Huy et al. (2016) proposed PHM, periodic high utility item set mining to uncover the periodic trends in a database that are repeated in some period. For example, some of the customers come to buy the ration every first week during the month. The algorithm shows that by pruning the non-periodic part of the database, the performance of the algorithm can be enhanced by several times.

Souleymane Zida et al. (2016) proposed EFIM, which is memory an efficient and fast algorithms for HUIs mining. It uses upper bounds sub-tree util and local-util to efficiently reduce the search space. An array-based method is also used for counting called as Fast Utility Counting for calculating the upper bounds. Database projection and merging schemes are also introduced to reduce the scans of the database.

Morrteza Zihayat et al. [37], (2017) proposed a memory adaptive strategy for the data having streams. Streaming data is continuously generated and changes very fast. It can exhaust the memory of computation and other resources. The algorithm given by Zihayat adopts an adaptive strategy in case of data streams. A tree structure called as MAS is used to potentially store the candidates of HUI of a stream. However, the candidates are approximated because of the pruning method used to adapt according to memory. The least promising part from the tree is pruned whenever there is a need of memory adaptation. So, it is an area of further research to improve the quality of approximated candidates.

Analysis of Various Algorithms studied above is summarized in the table below.

Table 3: Brief Analysis of Various Algorithms.

Sr No	Auth-or	Year	Techni-que	Benefits	Limitations
1	Morteza Zihayat et al.	2017	Memory Adaptive-HUSP	Memory adaptive over data streams	Approximation HUIs because of pruning the part of the tree, which is least promising when necessary, Overhead.
2	Souleymane Zida et al.	2016	EFIM	Utility counting is fast in linear time. Speed. Scalable	Slow if the database is sparse.
3	Jun-Zhe Wang et al.	2016	Top-k HUSP	Developed strategies for tree pruning and upper bounds for utility.	Run out of memory space in some cases.
4	Dachuan Huang et al.	2015	Smart-Cache	Introduced an additional layer of cache memory and a selective analyzer, thus reducing the execution time by 45.4% by reducing the scanning overhead.	All the intermediate information cannot be cached.
5	Liao J. et al.	2014	MRPrePost	Better stability & scalability, performance better than PFP & PrePost	Mined itemsets are Approximate which are closer to original result
6	Unil Yun et al.	2014	HUP in incremental Database Growth (HUPID)	Incremental database can be handled effectively	Costly Join Operation
7	Philipe Fournier et al.	2014	FHM	Estimated-Utility Co-occurrence pruning	Depth-first search is followed where different levels of HUIs are stored during mining. Utility list is larger as two transactions having the same set of itemset can't be merged during join.
8	Bai-En Shie	2012	Interesting Mobile-SPM	Efficient tree structure. Performance improved if constraints are also given.	Tree-based algorithms are better but use more memory than level-wise ones.
9	Mengchi Liu et al.	2012	HUI-Miner	Single phase, No multiple database scans	Uses vertical representations for mining HUIs where correlation among the items is calculated again with the 'join' operation which is time bearing and costly
10	Junqiang Liu	2012	DDHUP	No candidate sets are generated, therefore	Approximation HUIs

	et al.			scalable	
11	Vincent S. Tseng et al.	2010	UP-Growth	UP-Tree construction and then high potential item sets	Complex for evaluation as tree structure is used
12	Alva Erwin et al.	2007	CTU-Mine	Pattern growth algorithm is used, no second scan which is expensive to do	Complex for evaluation as tree structure is used
13	Ying Liu et al.	2005	Two Phase	Phase1 for finding candidate item sets, Phase 2 for calculating utilities	Multiple scans of database and huge candidate generation

4. Challenges of hui mining

Mining of HUIs for big data sets is very crucial for various applications and business purposes. The main challenge in utility mining is to restrict the size of search space and thus the candidate sets. Various findings from the above studies depict following challenges in this area-

1) Because of a large amount of data, search space is very large. So, mining becomes expensive in terms of computation. It becomes very difficult to scan the large databases again and again.
2) Downward closure property or Apriori of FIM does not hold for HUI. A subset of HUI set may or may not be HUI. An item set of high utility can have subsets or super sets with utilities equal, lower or higher values.
3) The problem of specifying the utility threshold- too high or too low. If the threshold is set too high, many important item sets would be missed out. If it is set too low, there would be many candidates.
4) Distributed environment is needed for large search space. So, a well- designed parallel architecture is needed to efficiently partition the search space, minimize the overhead of duplication and synchronization, fault tolerance and scalability issue.

5. Conclusion and future work

High Utility Mining is the emerging approach for ARM as it mines the item sets based on their utility and not only on the frequency of occurring. It is an important aspect of data mining with numerous applications. A literature survey is carried out in this review paper for high utility item set mining algorithms proposed by various researchers earlier to mine the HUIs efficiently. Various challenges for mining from big data sets have been identified. HUI mining is more tedious than FIM as the downward closure property is not valid for utility item sets. A subset of HUI may or may not be HUI. Specifying the threshold for utility is also difficult and greatly influences the performance of the algorithm. Mining of HUIs from large data sets is an area of further research because the search space is very large and distributed environment is needed for scalability and efficiently partition of the search space. Thus, various algorithms discussed here would be of great use to develop an improved technique(s) for mining the item sets or patterns with the utility from large data sets. A novel technique for high utility item set mining will be developed based on the distributed framework which can lead to wide rooms for future work of research into this area. Some of the research possibilities are to extend the algorithms for various versions of HUIM like incremental HUIMs, periodic HUIMs, top-k HUIMs, top-k utility episodes, top-k high utility web access patterns, sequential patterns etc.

6. Expected contribution of research

The review of various algorithms, their benefits and limitations are being studied here, and it would provide a base for the better understanding of utility base mining of item sets. This research focuses on mining the data with internal and external utilities associated with the items and item sets and then formulated the association rules from them. Some of the application areas for HUIM are market basket analysis, smarter healthcare, click stream analysis, cross marketing, genome analysis, drug design (molecular fragment mining), technical dependence, telecommunication systems, network diagnosis (fault co-occurrence), episode mining, e-commerce, etc. The novel technique(s) will help the data mining community to achieve a significant reduction in time and memory with efficient mining of the high utility item sets from large databases in the distributed environment.

References

[1] Agrawal, R. what's more, R. Srikant. 'Quick calculations for mining affiliation leads in substantial databases' The International Conference on Very Large Data Bases. San Francisco: Morgan Kaufmann Publishers, pp. 487– 499, 1994.

[2] Alva Erwin, Raj P. Gopalan, and N.R. Achuthan, 'Effective Mining of High Utility Itemsets from Large Datasets', Springer, pp 554-561, 2008.

[3] Yao H., H. J. Hamilton and C. J. Butz. 'A foundational way to deal with mining itemset utilities from databases', SIAM International Conference on Data Mining, pp. 211– 225, 2004.

[4] Yao, H. what's more, H. J. Hamilton. 'Mining itemset utilities from exchange databases' Data and Knowledge Engineering, volume-59, issue-3, pp-603– 626, 2009.

[5] Yen, S. J. what's more, Y. S. LEE. 'Mining high utility quantitative affiliation rules', Data Warehousing and Knowledge Discovery. Berlin: Springer, pp. 283– 292, 2007.

[6] P.K.Srimania, Malini M. Patilb, 'Visit Item set Mining utilizing INC_MINE in Massive Online Analysis Frame work', International Conference on Advanced Computing Technologies and Applications (ICACTA-2015), ScienceDirect, ELSEVIER.

[7] Kiran Chavan, Priyanka Kulkarni, Pooja Ghodekar, Frequent itemset digging for Big information, Conference on Green Computing and Internet of Things (ICGCIoT), 2015.

[8] Asbern A, P.Asha, Performance Evaluation Of Association Mining In Hadoop Single Node Cluster With Big Data, International Conference on Circuit, Power and Computing Technologies 2015 IEEE. https://doi.org/10.1109/ICCPCT.2015.7159257.

[9] Junqiang Liu, Ke Wang, Benjamin C. M. Fung, 'Coordinate Discovery of High Utility Itemsets without Candidate Generation', IEEE twelfth International Conference on Data Mining, pp-984-989, 2012.

[10] Wei Song, Yu Liu, Jinhong Li, 'Vertical Mining for High Utility Itemsets', International Conference on Granular Computing, IEEE, 2012. https://doi.org/10.1109/GrC.2012.6468563.

[11] Show-Jane Yen, Yue-Shi Lee, 'Mining High Utility Quantitative Association Rules', pp. 283-292, Springer, 2007.

[12] Tao-Yuan Jen, Claudia Marinica(B), and Abir Ghariani, 'Mining Frequent Itemsets with Vertical Data Layout in MapReduce', Springer, pp. 66– 82, 2016.

[13] Joy Rini, Sherly K, Parallel 'Incessant Itemset Mining with Spark RDD Framework for Disease Prediction', International Conference on Circuit, Power and Computing Technologies, IEEE, 2016.

[14] Fan Jiang, Carson Kai, Sang Leung, Richard Kyle MacKinnon, 'Discovering efficiencies in visit design mining from huge unverifiable information', Springer, 2016.

[15] Yen-hui Liang and Shiow-yang Wu, 'Arrangement Growth: A Scalable and Effective Frequent Itemset Mining Algorithm for Big Data Based on MapReduce Framework', International Conference on Big Data, IEEE, 2015.

[16] Kim, D., and Yun, U. 'Mining high utility itemsets in view of the time rotting model. Smart Data Analysis', pp 1157– 1180, 2016.

[17] Unil Yun, Heungmo Ryang, 'Incremental high utility example mining with static and dynamic databases', Applied Intelligence, Volume 42, Issue 2, pp 323– 352, Springer, 2015.

[18] Morteza Zihayat, Yan Chen, Aijun A, 'Memory-versatile high utility successive example mining over information streams', Machine Learning, pp 1-38, Springer, 2017.

[19] Ying Liu, Wei-keng Liao, Alok Choudhary, 'A Two-Phase Algorithm for Fast Discovery of High Utility Itemsets', Springer, pp 689-695, 2005.

[20] http://hadoop.apache.org.