ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

A Survey Mining High Utility Item Sets And Frequent Item Sets

R.Shyamala Devi[1] , D.Shanthi[2]

P.G. Student, Department of Computer Science and Engineering, PSNA Engineering College, Dindigul, India[1]

Professor, Department of Computer Science and Engineering, PSNA Engineering College, Dindigul, India[2]

ABSTRACT: The frequent item set mining and high utility item set mining are becoming hectic tasks in data mining. Frequent Item set Mining (FIM) treats all items having the same importance or same profit and it assumes that every item in a transaction in the same level and it is very difficult to find the High utility item (HUI) without redundancy. FIM cannot satisfy the requirement of users who desire to discover item sets with high utilities such as high profits. The utility represents the importance of the product and interestedness of the users. Moreover in HUI, even though the large number of results with high utility were produced, the user is not able to get the comprehend result. Meanwhile in a longer transaction the truncation technique leads the loss the information and the privacy is also lagging. Hence to overcome these issues it is proposed to provide a High Utility Item without redundancy and to find the closest item set. In this study in a longer transaction the truncation techniques will be used instead of splitting techniques to maintain privacy. After having detailed literature survey it is decided to develop a new hybrid Frequent Pattern- Apriori Algorithm (FP-AP) to mine the HUI sets.

KEYWORDS: Frequent item set mining, Utility Mining, Data Mining, lossless and concise representation, closest item set mining.

I. INTRODUCTION

FIM may discover a large amount of frequent but low profit item sets and lose the information on valuable item sets having low selling frequencies .These problems are FIM treats all items as having the same importance/unit profit/weight and it assumes that every item in a transaction appears in a binary form, i.e., an item can be either present or absent in a transaction, which doesn't include its purchase quantity in the transaction. Hence, FIM cannot satisfy the requirement of users who desire to discover item sets with high utilities such as high profits . The utility of an item set indicates its importance, which can be measured in terms of weight, profit, cost, quantity or other information depending on the user importance. An item set is known as high utility item set (HUI) if its utility is no less than a user-specified minimum threshold .Utility mining is a needed task and has a wide range of applications such as website stream analysis marketing in retail stores, mobile commerce environment and biomedical applications .

Many studies [4],[11] were proposed for mining HUIs, but they often present a large number of high utility item sets to users. A very large number of high utility item sets makes it difficult for the users to comprehend the results. The performance of the mining process decreases greatly for low minimum utility thresholds or when dealing with dense databases. Privacy assures that the output of a computation is insensitive to changes in any individual's record, and thus restricting privacy leaks through the result

The utility-privacy trade-off can be improved by limiting the length of transactions Thus, the transaction truncating (TT) approach proposed in [6] is not suitable to avoid privacy breach, we add noise to the support of item sets That is, if a transaction has more items than the limit, we divide it into multiple subsets (i.e., sub-transactions) and guarantee each subset is under the limit. to preserve more frequency information in subsets, we propose a graph-based approach to reveal the correlation of items within transactions and utilize such correlation to guide the splitting. High Utility Item set is based on the profit of the product. While considering the utility the redundancy may occur, so need to reduce the redundancy in the High Utility Item set. After finding the High Utility Item sets find the closest item set which consist of related item set. In FIM, to reduce the computational cost of the mining task present fewer but more important

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

patterns to users, many studies focused on developing concise representations

II. PROCESS FLOW

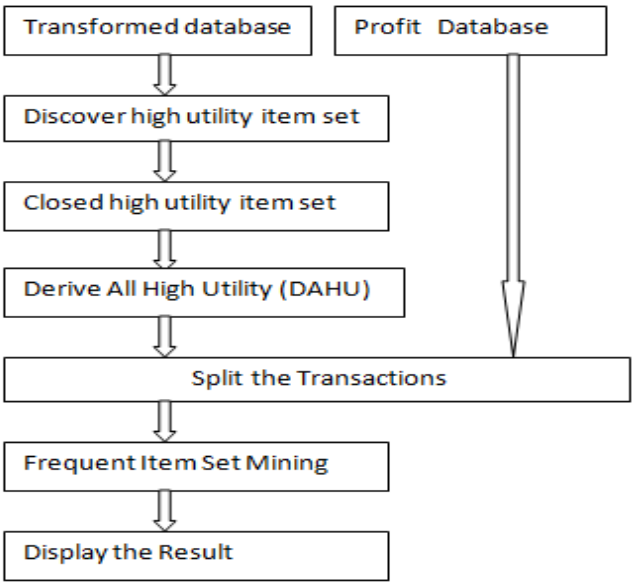

Fig 1. Process flow for finding HUI with transaction splitting

Fig.1 illustrates that there will be two database one for storing the transactions and another for profit details. High Utility Items will be found by discarding unpromising and isolated item set. Find the closest item set from the high utility item set. An item set is closed if none of its immediate supersets has same support as item sets. An item set is maximal frequent if none of its immediate supersets is frequent. Remove the unpromising items reduce the redundancy

DAHU (Derive All High Utility Item sets)is used to derive all the all the high utility from the DAHU tree . For a longer transaction, if truncation is done then information loss may occur. Instead, use splitting methods to split the longer transaction. Find the frequent item set with item set having the high utility. Privacy is the major issue in the real world. So add privacy to the result and display the result .

The reminder of this paper is organized as follows. Section II, describes the Related Works. Section III summarizes the conclusion.

III. RELATED WORKS

Different algorithms were proposed for finding High Utility Item sets and Frequent Item set mining and their pros and cons were listed below

A. High Utility Item sets and Frequent Item Set:

Smart truncation:

It solves the classical frequent item set mining problem, in which the goal is to find all item sets whose support exceeds a threshold. This raises the possibility of improving the utility-privacy trade off by limiting transactions' cardinality. In general impose such a limit so instead, we explore enforcing the limit by truncating transactions. Experimental results with datasets indicate that truncating has a large positive impact on quality ,limiting the maximal

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

cardinality of transactions is simple so truncate a transaction whose cardinality violates that constraint by only keeping a subset of that transaction. That heuristic technique requires us to compute the percentage of transactions for each cardinality, which also has privacy implications, and thus, we add geometric noise to that computation.

If a transaction has more than a specified number of items, we delete items until the transaction is under the limit. It is hard to precisely quantify the relationship between the maximal cardinality of transactions and the sensitivity of multiple item set. FIM treats all items as having the same importance/unit profit/weight and it assumes that every item in a transaction in the same priority.

CTU-PROL algorithm*:*

Improved performance by reducing both the search space and the number of candidates. Especially UP-Growth, not only reduce the number of candidates effectively but also outperform other algorithms substantially in terms of runtime. Does not provide comprehend result to the users and lots of useless candidates are pruned carries more computations and thus slower. [3]

Projection Based Mechanism:

It can directly generate the required item sets from the transactions. The indexing structure is designed to quickly find the transactions of each item set in the database. The original database is not copied for each item set to find high utility item sets, but instead, they are directly extracted through the indexing structure [3]. The memory consumed is thus less than that needed by directly copying the original database for mining. Each candidate sequence could be thought of as a query condition, and the tuples in the database that contained these sequences were projected. If Transaction is longer, then the performance will be degraded. Only the high utility item set were found but closed and maximal item set were not found.

CTU-Mine Algorithm

CTU-Mine algorithm to mine the complete set of high utility item sets from relatively dense or long pattern datasets. The data structure and algorithm extend the pattern growth approach, taking into account the lack of ant monotone property for pruning utility based patterns. Each item of the Item Tables is used as the starting point to mine all high utility item sets in the associated sub tree by first checking its TWU.

If a pattern generated by this transaction is included in some node of the current CTU-tree, then we update the TWU and the array of total quantities associated with it. While updating the transaction weight unit for any pattern the contributions of the items not included in the Item Table are deducted from the corresponding transaction value.

For high thresholds, Two-phase runs relatively fast compared to CTU-Mine, but when the utility of the threshold becomes lower, CTU-Mine performs Two-Phase. For very low utility thresholds, Two Phase could not complete the computation even after 10 hours of CPU Time. High utility item sets will be displayed to the user with redundancy and there is no concise result.

EFIM Algorithm (Efficient High-Utility Item Set Mining)

EFIM (Efficient high-utility Item set Mining), which introduces several new ideas to more efficiently discovers high-utility item sets both in terms of execution time and memory. EFIM relies on two upper-bounds named sub-tree utility and local utility to more effectively prune the search space. It also introduces a novel array-based utility counting technique called Fast Utility Counting to calculate these upper-bounds in linear time and space.[17]

Transaction merging is obviously desirable. However, a key problem is to implement it efficiently. To find identical transactions in O(n) time,, sort the original database according to a new total order T on transactions. Sorting is achieved in time, and is performed only once. Projected databases generated by EFIM are often very small due to transaction merging.

Privbasis Algorithm

It meets the challenge of high dimensionality by projecting the input dataset onto a small number of selected dimensions that need to be cared. In fact, PrivBasis often uses several sets of dimensions for such projections, to avoid

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

any set containing too many dimensions. Every basis in B corresponds to one such set of dimensions for projection. This technique enables one to select which sets of dimensions are most helpful for the purpose of finding the k most frequent item sets. [8]

When the column N is larger than the truncated frequency approach is completely ineffective. The reason why the Truncation Frequency (TF) method does not scale is that when one needs to select the top k item sets from a large set U of candidates, with large size causes two difficulties. The first is regarding the running time. Even if every single low-frequency item set in U is chosen with only a small probability, the sheer number of such low-frequency item sets means that the k selected item sets likely includes many infrequent ones.[6]

The TF technique tries to notify the running time by pruning the search space, but it does not address the accuracy challenge. This addresses the symptom for a larger candidate set, but not the root cause.

Clasp (Closed Sequential Patterns Algorithm)

Different strategies have been proposed so far, among which SPADE [16], exploiting a vertical database format Free Span and Prefix Span, based on projected pattern growth are the most popular ones. These strategies show good performances in databases containing short frequent sequences

Even worse, sometimes it is impossible to complete the algorithm execution due to a memory overflow. There are two main different changes added in ClaSP[15] with respect to SPADE is the step to check if the sub tree of a pattern can be skipped and the step where the remaining non-closed patterns are eliminated . Regarding the non-closed pattern elimination the process consists of using a hash function with the support of a pattern as key and itself as value. If two patterns have the same support we check if one contains the other, then the condition is satisfied, we remove the shorter pattern. It doesn't found the item which has the high utility .

Sparse vector mechanism

Discovers frequent item sets using a small portion of the privacy budget. Algorithm uses the remaining privacy budget to efficiently and effectively build a differentially private FP-tree. [5]

For longer transactions the computation may take for longer time so if truncation is done then information loss may occur .The user always expect the result should be in a concise manner without any information loss.[4]

SPADE (Sequential Pattern Discovery Using Equivalence Classes)

SPADE not only minimizes I/O costs by reducing database scans, but also shortens the computational costs by applying efficient search schemes. The vertical id-list based approach is also insensitive to data-skew. SPADE outperforms previous approaches by a factor, and by an order of magnitude if we have some included off-line information. Further, SPADE scales linearly in the database size, and a number of other database parameters.

The key challenge of handling graph datasets is the unknown output space while applying exponential mechanism. The Diff-FPM algorithm meets the challenge by unique frequent pattern mining and applying differential privacy into an MCMC sampling framework. removing a node in a star graph may result in an empty graph.

CHARM

That it is not necessary to mine all frequent item sets in the initial step, instead it is sufficient to mine the set of closed frequent item sets, which is much smaller than the set of all frequent item sets. It is also not necessary to mine the set of all possible rules. We show that any rule between item sets is equivalent to some rule between closed item sets. Thus many redundant rules can be eliminated.

When the number of closed item sets itself becomes very large, we cannot hope to keep the set of all closed item sets in memory CHARM does better at lower supports.

IUPG-Algorithm

These algorithms are experimented on synthetic datasets and real time datasets for different support threshold. IUPG algorithm performs well than UPG algorithm for different support values. IUPG algorithm scales well the size

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

of the transaction database increases.
values are beyond a user specified threshold in a transaction database.

Although DGU and DGN strategies are effectively reduce the number of candidates in Phase 1(i.e., global UP-Tree). But they cannot be applied during the construction of the local UP-Tree. Instead use, DLU strategy (Discarding local unpromising items) to remove the utilities of low utility items from path utilities of the paths and DLN strategy (Discarding local node utilities) to remove utilities of descendant nodes during the local UP-Tree construction.[6][7]

Two Phase Algorithm

It is used to mine prefixes and a candidate set of substring patterns. Consider both *substring* and *prefix* patterns. The frequency of the substring patterns is again refined in the successive phase where we employ a new transformation of the original data to reduce the noise a local transformation technique on the original string records which reduces the amount of noise injected by the Laplace Mechanism [7].

The magnitude of the noise does not depend on the real answer, the mechanism turns out to be in favour of queries with a large count. It doesn't include the mining of noisy frequent patterns as well as the mining of patterns with concise representation such as closed sequential patterns. The existing solutions to the issue make repeated database scans,[16] and use complex hash structure. SPADE uses combinatorial properties to divide the lager problem into smaller sub-problems that can be individually solved in main-memory using better lattice search techniques, and using simple join operations. Thus depending on the amount of main-memory available, that can recursively partition large classes into smaller ones, until each class is small enough to be solved independently in main-memory ,to implement the subsequence pruning step efficiently, we need to store all frequent sequences from the previous levels in some sort of a hash structure. compression can't be done in SPADE.

Private Frequent Pattern Mining Algorithm*:*

It has been well recognized that simple anonymization schemes that remove obvious identifiers carry heavy risks to privacy. Even privacy-preserving graph mining techniques based on k-anonymity are now often considered to offer insufficient privacy under strong attack models.[8] Recently, the model of differential privacy was proposed to restrict the inference of secured information even in the presence of a strong adversary. It needs the output of a differentially private algorithm is identical (in a probabilistic sense), or not a participant contributes her data to the dataset [12][13]..

IV. PROPOSED SYSTEM

Regarding the survey discussed above the issues like redundant utility items, time consumption, information loss, no comprehend results and privacy problems were occurred. Focusing on those issues the proposed work is to provide a comprehend results to the user without redundancy and also provide a closest item sets.Eventhough, for a longer transactions, a new splitting technique is used it maintains the privacy. The proposed algorithm FP-AP is used for mining high utility with frequent item set.

V. CONCLUSION

Frequent item set mining treats all the item in the same level so it is difficult to find the items which have the high utility. FIM mines the items which occur frequently but, HUI (high utility item set were used to identify the items with high profit or according to the user interestedness. Even in the HUI redundancy may occur which doesn't have any comprehend result for longer transactions the truncation is used which may loss some information. For sensitive information privacy issue occur This survey helps understand the issues of HUI and would serve as a beginning point to focus on mining closest item set and using different techniques for longer transactions and end result added with privacy.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

REFERENCES

[1] Sen su Shengzhi zu ; " Differentially Private Frequent Item Set Mining Via Transaction Splitting "IEEE Trans. Knowl. Data Eng", Volume:27, Issue 7 July 1 2015
[2] C. F. Ahmed, S. K. Tanbeer, B.-S. Jeong and Y.-K. Lee "Efficient tree structures for high utility pattern mining in incremental databases", *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 12, pp.1708 -1721 2009
[3] Tseng V.S.,"Efficient Algorithm For Mining Concise And Lossless Representation For Mining High Utility Item sets "IEEE Trans. Knowl. Data Eng", Volume:27, Issue 3 March 1 2015
[4] G.-C. , T.-P. , V. and S. Tseng "An efficient projection-based indexing approach for mining high utility item sets", *Knowl. Inf. Syst*, vol. 38, no. 1, pp.85 -107 2014
[5] R. Chan, Q. Yang and Y. Shen "Mining high utility itemsets", *Proc. IEEE Int. Conf. Data Min.*, pp.19 -26
[6] N. Li, W. Qardaji, D. Su and J. Cao "Privbasis: Frequent item set mining with differential privacy,", *Proc. VLDB Endowment*, vol. 5, no. 11, pp.1340 -1351 2012
[7] L. Bonomi and L. Xiong "A two-phase algorithm for mining sequential patterns with differential privacy,", *Proc. 22nd ACM Conf. Inf. Knowl. Manage.*, pp.269 -278 2013
[8] E. Shen and T. Yu "Mining frequent graph patterns with differential privacy", *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, pp.545 -553 2013
[9] C.-W. , T.-P. and W.-H. "An effective tree structure for mining high utility item sets", *Expert Syst. Appl.*, vol. 38, no. 6, pp.7419 -7424 2011
[10] R. Agrawal and R. Srikant "Fast algorithms for mining association rules", *Proc. 20th Int. Conf. Very Large Data Bases*, pp.487 -499
[11] A. Erwin, R. P. Gopalan and N. R. Achuthan "Efficient mining of high utility item sets from large datasets", *Proc. Int. Conf. Pacific-Asia Conf. Knowl. Discovery Data Mining*, pp.554 -561
[12] Y. Liu, W. Liao and A. Choudhary "A fast high utility item sets mining algorithm", *Proc. Utility-Based Data Mining Workshop*, pp.90 -99
[13] C.-W. , T.-P. and W.-H. "An effective tree structure for mining high utility item sets", *Expert Syst. Appl.*, vol. 38, no. 6, pp.7419 -7424 2011
[14] C. Lucchese, S. Orlando and R. Perego "Fast and memory efficient mining of frequent closed item sets", *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 1, pp.21 -36 2006
[15] V. S. Tseng, C.-W. Wu, B.-E. Shie and P. S. Yu "UP-Growth: An efficient algorithm for high utility item set mining", *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, pp.253 -262
[16] C.-W. Wu, B.-E. Shie, V. S. Tseng and P. S. Yu "Mining top-k high utility item sets", *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, pp.78 -86
[17] C.-W Wu, P. Fournier-Viger, P. S. Yu and V. S. Tseng "Efficient mining of a concise and lossless representation of high utility item sets", *Proc. IEEE Int. Conf. Data Mining*, pp.824 -833