International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 09 | Sep 2018 www.irjet.net p-ISSN: 2395-0072

Comparison of Different Techniques of High Utility Mining for Representation and Graphical Analysis

P. Payal Swamy[1]

[1]*Faculty, Dept. of CSE, Indo Asian Womem's pu and Degree College, Bengaluru*

Abstract - *In this paper we considered the problems we come across while mining high utility itemsets these are excessive memory usage , time required and the generation of many number of candidate itemsets. High utility itemsets may be defined as the itemsets with maximum profit. A comparative study of different algorithm along with the proposed approach is done by us. It's found that the proposed methods perform better as compared to the already existing algorithms. The results of all three algorithms namely CHUD Miner (closed high utility itemsets discovery), UP Growth and UP++ Growth, are compared using graphical analysis. Association rule mining algorithm is also used to as to get the total number of high utility itemsets from candidate itemsets without accessing the original database.*

Key Words: Data mining, threshold value, Utility mining, high utility mining, association rule mining

1. INTRODUCTION

Mining of frequent itemsets focuses on the threshold value only, and doesn't consider other factors such as profit gained, quantity or some other. High utility mining finds the items which pass the minimum threshold value and the itemsets which comes above that defined threshold are known as promising itemsets. This doesn't consider the quantity of the purchased item, so there is no necessities to find the significance of the items that are there in database. Threshold value may be said as the minimum limit that must be present; below that particular limit the items are rejected.

High utility itemsets have utility greater than user-defined minimum utility threshold if the utility is less than defined one then it is called a low-utility itemsets, threshold value is defined before only. Advancement in different technologies has made it achievable for retail organization to collect and stock up huge amount of sales data which is well known as basket analysis. Earlier defined methods that were used for utility mining produces large itemsets that will degrade performance, consequently this has become a challenging problem to the mining performance. To address this issue, we proposed a new algorithm with a compact data structure which will help efficiently in discovering high utility itemsets from transactional database. The compact data structure used will also help to reduce the amount of time required and memory usage in finding high utility itemsets.

The new algorithm introduced is compared with the already existing techniques and results are seen, these results are compared on the basis of memory usage and time required for representing high utility itemsets. Graphical analysis is done for seeing the results as these are in demand nowadays.

2. RELATED WORK

The redundancy issue in high utility itemset mining was solved by proposing a compact representation of all high utility itemsets. An approach was suggested by cheng et.al [1], CHUD (closed + high utility itemset discovery) so as to get high effectiveness for the mining task. After that, to pick up all high utility itemsets from set of closed+ high utility itemsets a new algorithm DAHU was proposed which will show the results without accessing the original database. The outcomes showed that the proposed algorithm achieves an enormous decrease of about 800 times and more. In addition, the paper showed that the mixture of CHUD and DAHU outperforms the state-of-the-art algorithms for mining high utility itemsets.

The research by Yun et.al [9], invented a new data mining technique to identify significant rare data rules and although they are really sparse in database, they are highly associated with very specific data. This play an important role when the data is stored and processes involved are humongous. The paper experimentally compared the new algorithm with existing data mining techniques to find out association rules. The backbone of proposed technique lies in adopting a new constraint called relative support that enables to identify the strong co- relation between infrequent rare data items. The experimental results showed that that the latest algorithm outperforms the existing ones.

The paper authored by Liu et.al [10], is one of the relevant paper that speaks about association rule mining in database. The article discussed about setting user specific constraints called minsup and minconf for generation of association rules in a database. Minsup controls the minimum support, or the data association needed for a rule to be valid, whereas minconf controls the predictive length of a rule. However extreme delicacy is needed while setting the parameter minsup as it needs to be low enough to cover all rare-frequent data associations, but at the cost of exponentially blown up combinatorial problem of data associations. The novel solution proposed to the problem called rare item problem is to set up multiple minsup to truly capture the varieties of frequency of different data items and their appearances. The paper was concluded by stating the experimental results that, a single minsup is insufficient for association rule mining so a relevant hybrid solution between setting the minimum support value neither too high nor too low should be taken into view. This leads to the invention to the invention of most relevant rare item rules omitting

meaningless ones. The paper mining interesting imperfectly sporadic rule by the koh et.al [13] is the accomplished future work that was mentioned by him earlier in the paper Inding sporadic rules using Apriori inverse. The article discussed on the mining interesting sporadic rules. This paper concludes by stating that the existing algorithm hardly tries to extract sporadic like infrequent itemsets, though the rules can be interesting enough for further investigation, the proposed method embraces the chance of accepting rules occurred by pure chance.

The paper finding sporadic rules using Apriori inverse authored by Koh et.al[14], proposed about a new category of rules called sporadic rules. Those are the rules, by definition, with low support but high confidence in a database, example a rare association of two symptoms indicating a rare disease. However, the paper proposed a method called Apriori inverse to discover sporadic rules which eliminates candidate rules that lies above a maximum support threshold. This leads to the classification of sporadic either perfect or imperfect basis. The article concludes with speaking relevance of sporadic rules and how the current mining algorithms fail to capture them.

3. PROPOSED METHOD

The proposed flow will be as –

Consider the figure shown, the block named as transactional database contains all the transactions involved (our experiments were done on book store database). Transactional utility of each item may be defined as the individual profit of particular itemsets in a transaction.

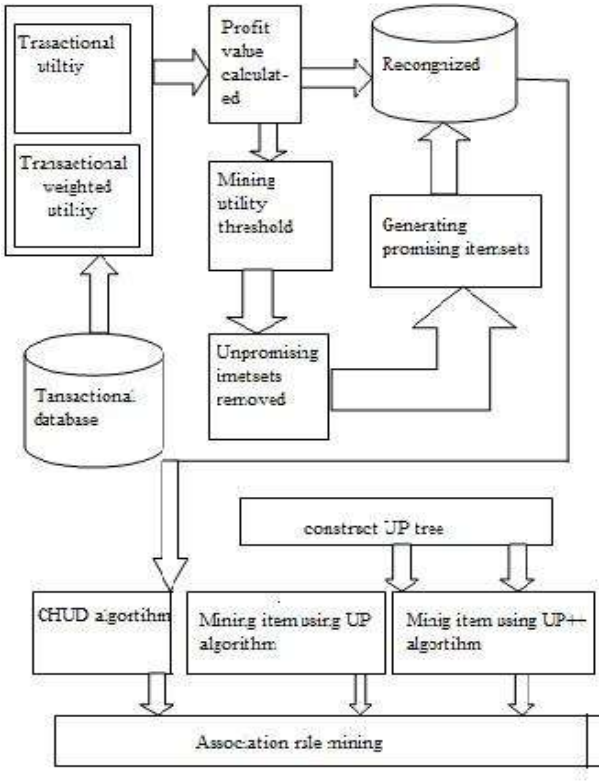

Fig-1: Flow of proposed system

For all the transaction the profit value is calculated by considering the transactional utility and transactional weighted utility, A minimum amount of profit value is decided which is termed as the threshold value if the itemsets has profit lesser than the decided threshold, the itemset get rejected. All the unpromising itemsets are removed and only the promising ones are recognized for further use.

The set of promising itemsets are kept aside and the different algorithms are applied on them. With the dataset of promising itemsets UP tree and UP+ tree is constructed and on the tree the respective algorithms were applied namely UP+ Growth and UP++ Growth. further more to this on the promising itemsets CHUD miner is applied.

An association rule mining algorithm DAHU (Derive all high utility itemsets) will show all the high utility itemsets in the transaction without considering the original database.

The UP-Growth+ algorithm will work as follows -

Subroutine: UP-Growth+ (Tp, Hp, X)

Input: A UP-Tree Tp, a header table Hp for Tp and an itemset X.

Output: All PHUIs in Tp.

Procedure UP-Growth+ (Tp, Hp, X)

Step 1 For every admission z(i) in Hp do

Step 2 Produce a PHUI P = X ∪ z (i);

Step 3 The approximation utility of P is set as z (i)'s usefulness value in Hp;

Step 4 Construct P's conditional pattern base P-CPB;

Step 5 Put local promising items in P-CPB into Hy;

Step 6 DLU to reduce path utilities is applied;

Step 7 Apply strategies DLN and insert paths into Tq;

Step 8 If Tq ≠ null then call UP-Growth+ (Tq, Hq,P);

Step 9 End for.

The UP tree maintained by using the promising itemsets will be considered as the input in applying the UP+ Growth algorithm. Along with this a header table is maintained which is initially empty

For every entry in the header table a PHUI P (promising high utility itemsets) is generated.

Two strategies DLU (discarding local unpromising itemsets) and DLN (discarding local node) are applied and the paths are inserted to Tq. The algorithm is called recursively till Tq is empty.

The CHUD algorithm work as follows -

First of all we need to provide the input dataset which is basically the transactional dataset along with the minimum utility.

Next to this the database is been scanned for finding the promising transactional utilization. Again for the second time the database is been scanned to generate Extended Utility.

Further, on the generated promising itemsets the algorithm GEN-CHUI is applied which works recursively to generate the candidate itemsets.

Two arrays are maintained by the algorithm post-array and the next-array, which will lead to the generation of candidate itemsets. Another strategies used will discard the local promising nodes and thus the number of candidate keys generated is very less.

The Up-Growth++ algorithm will work as follows -

Subroutine: UP-Growth++ (Tp , Hp , X)

Input: A UP-Tree++ Tp , a header table Hp for Tp and an itemset X.

Output: All PHUIs in Tp .

Procedure UP-Growth++ (Tx , Hp , X)

Step 1 For each entry zi in Hp do

Step 2 Generate a PHUI P = X ∪ zi ;

Step 3 The approximation utility of P is set as zi's profit value in Hp;

Step 4 Construct P's conditional pattern base P-CPB;

Step 5 Put local promising items in P-CPB into Hq

Step 6 Apply strategy DGU to reduce path utilities of the paths;

Step 7. Apply strategy DLN and insert paths into Tq ;

Step 8. If Tq ≠ null then call UP-Growth (Tq , Hq , P);

Step 9. End for

The already generated UP+ tree will be used for applying the above algorithm along with it a header table is maintained, same as in the previous algorithm.

For each entry in the header table a PHUI P (promising high utility itemsets) is generated.

The estimated utility of P is set as the items utility, after this conditional pattern base of P is generated and the promising itemsets are kept in this.

Two strategies DGU (discarding global unpromising itemsets) and DLN (discarding local node) are applied and the paths are inserted to Tq. If Tq has any value then UP-Growth++ is called until Tq becomes empty.

After applying the above two algorithms CHUD Algorithm which is an extension of DCI Closed is applied to mine closed Itemsets, DCI is one of the best methods to find high utility itemsets. In CHUD

Algorithm for mining CHUIs(candidate high utility are calculated and include several effective strategies for reducing the number of candidates generated in Phase1. Finally, the Main procedure performs Phase2 on these candidates to obtain all CHUI

At the end the UP+ Growth and UP ++ Growth results are seen for the synthetic and real database.

4. EXPERIMENTAL RESULTS

Experiments results show that the UP++ Growth outperforms than the already existing UP+ Growth algorithm. A synthetic dataset named Mybook store is maintained and database of this store is maintained on monthly basis. Amazing results were available when the improved UP Growth is been applied on this.

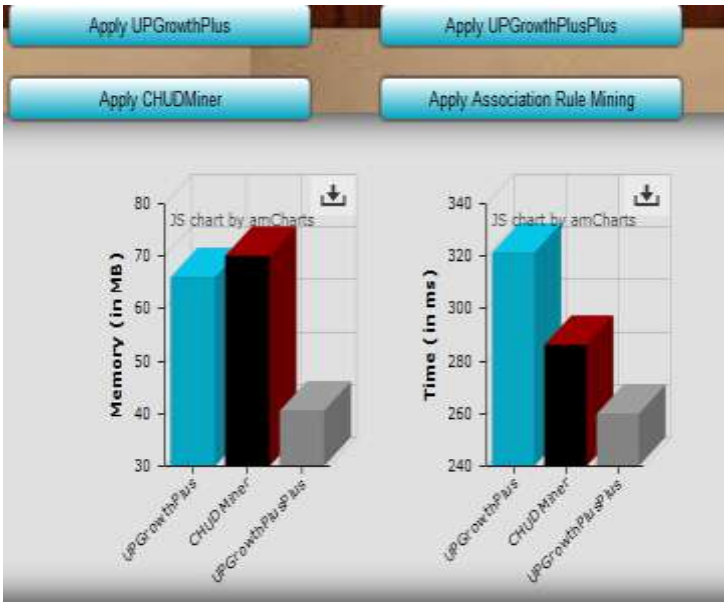

Fig-2: Graphical representation of memory usage and time required in UP-Growth, UP+- Growth and CHUD algorithm

In figure, all the algorithms applied on the same datasets and the results shows that the memory usage significantly decreased in improved UP++ algorithm as compared to the UP+ Growth algorithm. But in case of CHUD the results found to be similar as the proposed algorithm.

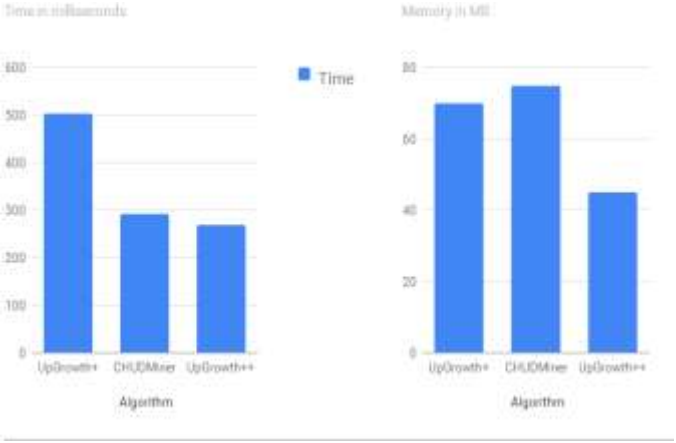

Fig-3: Showing the comparative analysis of all three algorithms using google charts

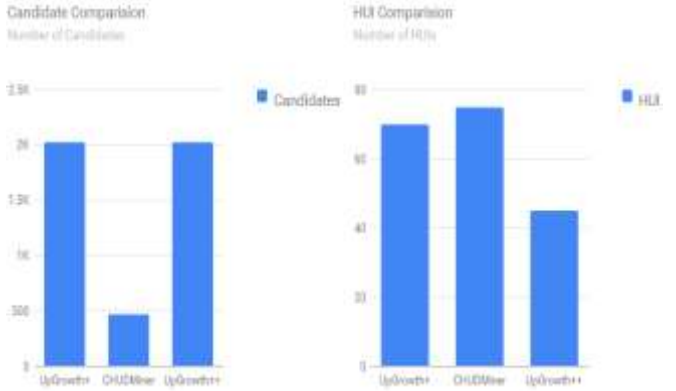

Fig-4: Showing the candidate itemsets generated and HUIs produced in each algorithm.

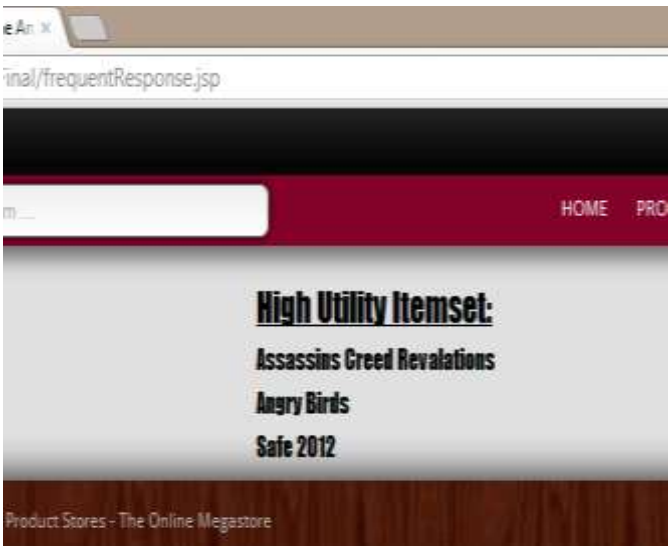

Fig-5: Product with high utility.

Association rule mining will select the itemsets with maximum profit from the candidate itemsets generated and provide us the same without accessing the original database. In our experimental results figure 4 shows the top 3 high utility itemsets present in the transactional database.

5. CONCLUSION

The algorithm named UP++ Growth is proposed here for mining high utility itemsets from transactions. UP++ Growth proved to be more efficient than the UP+ Growth as well as CHUD algorithm. Moreover, the generation of candidate itemsets is done only with two scans of the original database. It's also been seen that the number of high utility itemsets produced are comparatively lesser than in the other two algorithms

The mining performance is improved as the search space; number of candidates and time required are effectively reduced by the proposed strategies as compared to the UP Growth. The experimental results show that UP++Growth outperforms in terms of memory usage and time, especially when the size of the database is very large.

In future, there are many other compact representations such as getting the odd and even ratio as well as rare association rule mining which is not been applied till now can be included along with work done till now.

6. REFERENCES

[1] V. S. Tseng, B. E. Shie, C.-W. Wu and P. S. Yu, Fellow," Efficient Algorithms for Mining the Concise and Lossless Representation of High Utility Itemsets", IEEE Transaction on Knowledge and Data Engineering, vol. 27, no. 3, March,2015.

[2] F. A. Chawdhary, S. K. Tanbeer and B.-S. Jeong, and Young-Koo Lee, Member, IEEE "Efficient Tree Structures for High Utility Pattern Mining in Incremental Databases" IEEE Transactions Knowledge and Data Engineering, Vol. 21, Issue 12, pp. 1708-1721, December 2009.

[3] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules", In Proceedings of the 20th International Conference on Very Large Data Bases, pp. 487-499, 1994.

[4] C. F. Ahmed, S. K. Tanbeer, B.-S. Jeong and Y.- K. Lee, "Efficient tree structures for high utility pattern mining in incremental databases", In IEEE Transactions on Knowledge and Data Engineering, Vol. 21, Issue 12 , pp. 1708-1721, 2009.

[5] R. Chan, Q. Yang and Y. Shen, "Mining high utility itemsets", In Proceedings of Third IEEE International Conference on Data Mining, pp. 19-26, Nov., 2003.

[6] A. Erwin, R. P. Gopalan and N. R. Achuthan," Efficient mining of high utility itemsets from large datasets", In Proceedings of PAKDD, LNAI 5012, pp. 554-561,2008.

[7] H. Li, J. Li and L. Wong, "Relative Risk and Odds Ratio: A Data Mining Perspective", international conference on management, PODS June 13-15,2005.

[8] C. Nawapornanan and V. Boonjing, "An efficient algorithm or mining complete share-frequent item sets using Bit Table and heuristics", Proceeding of ICMLC, Vol.1, pp.96-101, 2012.

[9] Y.-C. Li, J.-S. Yeh and C.-C. Chang, "Isolated items discarding strategy for discovering high utility itemsets", In Data & Knowledge Engineering, Vol. 64, Issue 1, pp. 198-217, Jan., 2008.

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

[11] L. Szathmary, P.Valtchev and A. Napoli," Finding Minimal Rare Itemsets and Rare Association Rules", Proceedings of the 4th International Conference on Knowledge Science, Engineering and Management ,2010.

[12] Y. Hei., H. D.Hwang, B. Ryu, "Mining association rules on significant rare data using relative support", Journal of Systems and Software, vol. 3, Page no 181–191, 2003.

[13] K. Y. Rountree, N.o. keefee, "Finding Sporadic Rules Using Apriori-Inverse". In: Ho, T.-B., Cheung, D., Liu, H. (eds.), LNCS (LNAI), and Vol. 3518, pp. 97–106. Springer, Heidelberg (2005).

[14] K. Y. Rountree, N.o. keefee, "Mining Interesting Imperfectly Sporadic Rules" vol. 3918, pp. 473–482. Springer, Heidelberg (2006).

[15] J.-F. Boulicaut, A. Bykowski and C. Rigotti, "Free-sets: A condensed representation of Boolean data for the approximation of frequency queries," Data Mining Knowledge Discovery, vol.7, no. 1, pp. 5–22, 2003.

[16] T. Calders and B. Goethals, "Mining all non-derivable frequent itemsets",in Proceedings International Conference Principles Data Mining Knowledge Discovery, pp. 74–85, 2002.

[17] T. Calders and B. Goethals, "Mining All Non-Derivable Frequent Itemsets", in proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery, pp. 74-85,2002.

[18] J. Han and Y. Fu, "Discovery of Multiple-Level Association Rules from Large Databases", in the Proceedings of the International Conference on Very Large Data Bases Switzerland, pp. 420-431,2002.

[19] N.Pasquier, Y.Bastide, R.Taouil and L.Lakhal, "Efficient mining of association rules using closed itemset lattices,"Information Systems, Vol. 24, No. 1, pp. 25-46,1999.

[20] G.D.Ramkumar, Sanjay Ranka and Shalom Tsur, "WeightedAssociation Rules: Model and Algorithm", 1998.

[21] C.-K. Chui, B. Kao and E. Hung, "Mining Pacific-Asia conferences Advances in Knowledge Discovery and Data Mining , pp. 47-58, 2007.

[22] S. Brin, R. Motwani, J. D. Ullman, S.Tsur, "Dynamic itemset counting and implication rules for market basket data", Procedings of the ACM SIGMOD International conference on Management of Data, USA, 1997.

[23] P. swamy, A. pimpalkar, "Enhancing Data mining Techniques for Graphical Analysis and Representation of High Utility Itemset", IJCSET, Vol 5, Issue 12, 414-416,December 2016.