IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

Time Efficient Item Elimination Based Technique for Mining High Utility Items from a Data Set

Bharti Ahuja[1], Mrs. Rupali Bhartiya[2]

Department of Computer Science & Engineering, SVITS, Indore, India[1, 2]

Abstract: The data mining and their different applications are becomes more popular now in these days a number of large and small scale applications are developed with the help of data mining techniques i.e. predictors, regulators, weather forecasting systems and business intelligence. There are two kinds of model are available for namely supervised and unsupervised. The performance and accuracy of the supervised data mining techniques are higher as compared to unsupervised techniques therefore in sensitive applications the supervised techniques are used for prediction and classification. This paper presents a high utility item set mining technique. In this technique, the useless patterns are removed at the initial stage of mining. So it is helping in getting less time consumption.

Keywords: Data mining, High Utility Item sets Mining, Transactional Utility, and Weighted Transactional Utility.

I. INTRODUCTION

In utility mining [3,4] we concentrate on utility value of itemset while in frequent item set mining we concentrate that how frequently items appears in transactional database. Frequently, data mining is the method of analyzing facts from dissimilar perspectives and summarizing it into needful information - information that can be used to enlarge profits, cuts expenses, or both. Data mining software is one of the analytical tools for searching data. It grants users to analyze data from many unlike scope or angles, classify it, and review the relationships identified. Technically, data mining is the procedure of result correlations or patterns among dozens of fields in huge relational databases [2].

A. Data

Data be any information, figures, or textbook that can be processed by a computer. Current, organizations are building up huge and rising amounts of data in different formats and different databases. This includes:

- Operational or transactional data such as, sales, price, stock, payroll, and accounting.
- Non-operational data, such as manufacturing sales, estimate data, and macro-economic data.

B. Information

The patterning, relations, or associations among all this facts can provide information. For example, study of retail point of trade transaction data can yield information on which goods are selling.

C. Knowledge

Information can be renewed into knowledge about historical patterns and future trends. For example, summing up information on deal supermarket sales can be examine in light of promotional efforts to offer knowledge of customer buying actions. Thus, a producer or dealer could determine which items are most liable to promotional efforts.

D. Data Warehouses

Theatrical advances in data imprison, processing power, data broadcast, and storage capabilities are enabling organizations to mix their various databases into data warehouses. Data warehousing is defined as a procedure of consolidating data management and repossession. Data warehousing, like data mining, is a relatively latest term although the approach itself has been around for years. Data warehousing represents a model idea of handling a central repository of all organizational data. Concentration of data is required to exploit user access and analysis. Powerful automation advances are making this idea a reality for numerous of companies. And, fairly powerfull advances in data analysis software are allowing users to access this data liberally.

The data analysis software is what supports data mining [2]. Some methods were proposed for mining high utility item or itemsets from the databases, such as UMining [9], Two-Phase [7,8], IIDS [6] and IHUP [5]. UMining algorithm [9] proposed by Yao et al. used an estimation method to prune candidate itemset in memory. Also it is shown to have good performance but it cannot capture the full set of high utility itemsets since some high utility patterns may be eliminated during the process.

II. BACKGROUND & RELATED WORK

The main concepts are as follow on the basis of tables shown below (Table 1 and Table 2)

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

Concept A
The High utility itemset is a collection of items that emerge at least in a pre-specified number of transactions. Formally, allow I = {i1, i2, . . . , im} be a set of items and DB = {TR1, TR2, ..., TRn} a set of transactions where each one transaction is also a set of items (i.e. itemset).

Concept B
The utility about an item ip is a numeral value yp delineated by the user. It is transaction autonomous and follows importance (usually profit) of the item. External utilities are gathered in an utility table.

Concept C
The utility about an item set X in a transaction TRi is stands for U(X,TRi) & it is calculated as follows. Like , U({AC}, TR1) = U({A}, TR1) + U({C}, TR1) = 5 + 1 = 6.

Concept D
The utility about an item set X in D is stands for U(X) & it is measured as follows like, U({AD})=U({AD}, TR1) + U({AD}, TR3) = 7 + 17 = 24.

Concept E
An itemset is define as high utility itemset if its utility is not lower than a user-specified minimum utility threshold so it is denoted as min_util. Or else, it is called a low utility itemset.

Concept F
The transaction utility of a transaction Td is denoted as TU(Td) and defined as u(Td, Td). For example, TU(TR1) = u({ACD}, TR1) = 8.

Table 1: Transaction Data Set

TR_ID	TRANSACTION	TU
TR1	(A,1) (C,1) (D,1)	8
TR2	(A,2) (C,6) (E,2) (G,5)	27
TR3	(A,1) (B,2) (C,1) (D,6) (E,1) (F,5)	30
TR4	(B,4) (C, 3) (D,3) (E,1)	20
TR5	(B,2) (C,2) (E,1) (G,2)	11

Table 2: Item & correspondent profit

ITEM	A	B	C	D	E	F	G
PROFIT	5	2	1	2	3	1	1

RELATED WORK

One of the renowned algorithms is the Apriori algorithm [1], which is the pioneer being efficiently mining association rules from huge databases. The tree-based approaches such as FP-Growth [5] were afterward proposed. It's generally recognized that FP-Growth achieves a improved performance than Apriori-based accesses since it finds frequent itemsets without developing several candidate itemset and it search database just twice. However, in the framework of frequent itemset mining [1, 5], the priority of items to users is not considered. The unit profits and purchased bulkes of the items are not taken into applications proposed by Yao et al. used an evaluation approach to prune search space. Although it is shown to have favorable performance, it cannot capture the entire set of high utility itemsets considering a few high utility patterns may be eliminated during the method.

Two-Phase algorithm [7] proposed by Liu et al. be of two phases. In phase I, Two-Phase algorithm operates a breadth first search strategy to calculate HTWUIs. It produces candidate itemsets about size k from HTWUIs of length (k-1) and eliminates candidate itemsets by TWDC property. In every pass, HTWUIs along with their evaluated utility values i.e., TWUs, are computed by finding database.

After that, the whole set of HTWUIs is assembled in phase I. In phase II, high utility itemsets along with their utilities are analyzed from the HTWUIs by finding original database once. Whereas Two-Phase algorithm effectively scale down the search space by TWDC property including captures the whole set of high utility itemsets, it still produces too many candidates for HTWUIs including it requires multiple database scans. To overcome this overhead, Li et al. [6] suggested an isolated items removing strategy, abbreviated as IIDS, to weaken the number of candidates. By eliminating isolated items mean while the level-wise search, the number of candidate itemsets for HTWUIs in phase I can be weaken effectively. Nevertheless, this approach still scans database multiple times including uses a candidate generation-and-test scheme to search high utility itemsets. To efficiently produces HTWUIs in phase I also avoid browsing database many times, Ahmed et al. [2] suggested a tree-based algorithm, labeled IHUP, as mining high utility itemsets.

They use an IHUP-Tree to managing the information of high utility itemsets also transactions. Each node in IHUP-Tree be of an item name, a support count, and a TWU value. The framework against the algorithm be of three process: (1) The construction of IHUP-Tree, (2) the generation of HTWUIs also (3) recognition of high utility itemsets. The phase I of IHUP In step 1, items in the transaction are reorganized in a established order such as lexicographic order, support descending order or TWU descending order. Then, the reorganized transactions are inserted into the IHUP-Tree. In step 2, HTWUIs are generated from the IHUP-Tree by operating the FP-Growth algorithm [5].So, HTWUIs in phase I can be being high efficiently without producing candidates for HTWUIs. In step 3, high utility itemsets also their utilities are analysed against the set of HTWUIs by browsing the original database once.

Although IHUP search HTWUIs without producing any candidates for HTWUIs and accomplish a better

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

achievement than IIDS and Two-Phase, it still produces several HTWUIs in phase I. remember that IHUP also Two-Phase produce the equivalent number of HTWUIs in phase I because they handling transaction-weighted utilization mining model [7] to overrate the utilities of the itemsets.

Nonetheless, this model may overestimate several low utility itemsets as HTWUIs and generates too many candidate itemsets in phase I. Such a huge number of HTWUIs degrades the mining achievement in phase I in terms of execution time and memory utilization. By, the number of HTWUIs in phase I also affects the achievement of the algorithms in phase II since the high HTWUIs are produced in phase I, the more execution time is required for determining high utility itemsets in phase II.Just as declared above, the number of HTWUIs produced in phase I forms a crucial problem to the achievement of algorithms. In view of this, we suggested four ideas to weaken the estimated utility values of the itemsets.

Through applying the suggested strategies, the number of candidates generated in phase I can be compacted effectively and the high utility itemsets can be determine more efficiently since the figures of itemsets needed to be checked in phase II is highly weaken in phase I.

III.PROBLEM STATEMENT & PROPOSED SOLUTION

Known transaction database D including a user – stated minimum utility threshold min unti_until, mining high utility itemsets from the transaction database is identical to determine from D all itemsets whose utilities are no less than min_util.Let us consider a transactional database D in which each transaction is shown by an unique transactional id called TR_ID .Each transaction contain some items and each item has its utility value. After calculating utility of each item in a particular transaction we get transaction utility TU and than for weighted transactional utility WTU we find out utility of each item or itemset but in whole transactional database D which contain this item or itemset. After compare WTU with user specified minimum utility (min_utly) we get high utility itemset as a candidate itemset for further processing. The objective of this research work is to develop a method to discover high utility itemset from the transaction database based on weighted transactional utility of item or itemset . The algorithm first finds an initial list of high utility itemsets. Then initial list is converted into the final list of itemsets by eliminating the less utility of itemsets. Our objective is to propose a method which generates the result list comparatively in less time than older algorithm.

We will propose a novel approach for high utility item set mining. The latest algorithm will do best than the previous algorithms in terms of execution time.

THE STEPS OF THE PROPOSED ALGORITHM ARE AS FOLLOWS:

Step A: An initial list of high utility item set is generated as follows:

- Given Transaction Utility (TU) the transaction utility about an item is the addition of the utilities of every items in that transaction.
- Weighted transaction utility of an item set: The weighted transaction utility of an item set is accessed by operating the addition of the transaction utility of every transaction containing that item set.
- Only those item sets are included in the initial high utility item set mining list whose weighted transaction utility is more than the minimum utility.

Step B: In this step, final high utility item set is generated by eliminating the infrequent item sets from the list of step 1. It is performed as follows:

- An item set is chosen from the list of step 1.
- If the utility of the item is less than the min_utility (Minimum Utility) than the item is erased. Otherwise, the item set is selected in the final list of the high utility item set.

Step C: From candidate of size 1, we recursively create candidates of greater size as follows:

- For each itemset I1 and I2 of level k-1.
- Here we compare items of item set1 and item set2.If they have all the same k-1 items and the last item of itemset1 is smaller than the final item of itemset2, we will merge them to generate a candidate.
- Calculate TWU of itemset.
- if the transaction weighted utility (TWU) is high enough.
- add it to the set of HWTUI of size .
- Continue this process until there are candidates to combine.

Step D: If the utility of a candidates is less than the minimum threshold then remove such candidate from the list of high utility items.

Step E**:** Return all high utility item sets found.

Step F**:** End of process.

IV.EXPERIMENTAL RESULTS

Table3: Items Utility for Transaction

TR_ ID	Items	Transaction utility value	Item utility values for this transaction
tr1	3 5 1 2 4 6	30	1 3 5 10 6 5
tr2	3 5 2 4	20	3 3 8 6
tr3	3 1 4	8	1 5 2
tr4	3 5 1 7	27	6 6 10 5
tr5	3 5 2 7	11	2 3 4 2

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

Result :-It took Minutil=30.

We compared the performance of the UP Growth and proposed new algorithm. The result obtained is as follows: Both algorithm produced the same set of high utility item sets. It is as follows:

Table4: Items with their Transaction Utility

ITEMS	TRANSACTION UTILITY
6 4 2 1 5	30
4 2	30
4 2 5	36
4 2 5 3	40
4 2 3	34
2 5	31
2 5 3	37
1 5 3	31

The time consumed by UP Growth is 79 ms & by the new proposed is 31 ms.

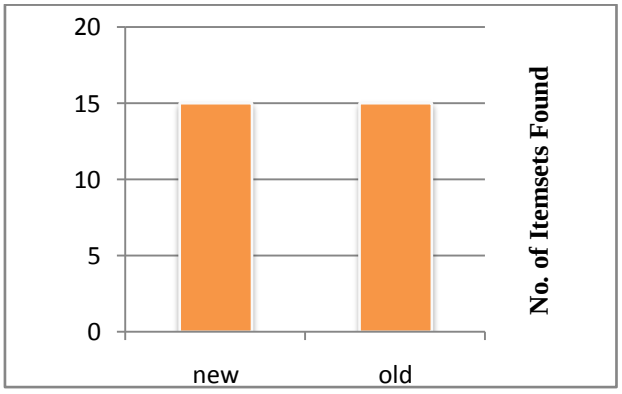

Figure 1: Result Comparison

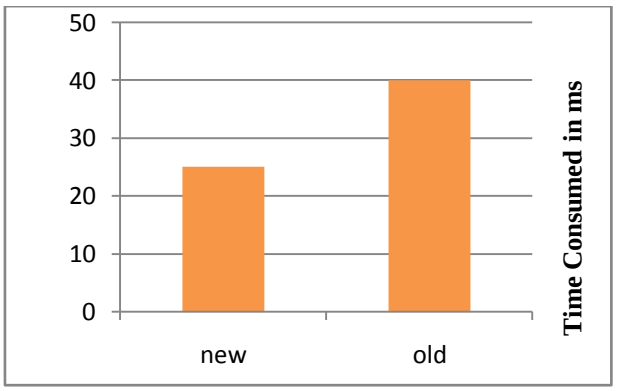

Figure 2: Time Consumption Comparison

As shown in above graphs the number of itemsets found by the both algorithms are same but the time consumption is very less in the proposed algorithm.

V. CONCLUSION

The data capturing technologies is also increasing. In utility mining we concentrate on utility value of itemset while in frequent item set mining we concentrate that how frequently items appears in transactional database. In this paper, individually surveyed the account of existing high utility mining techniques. However we surveyed different concepts of Association rule mining and frequent itemset mining techniques which play significant role for basic of utility itemset mining but we restricted ourselves to the classic high utility mining problem. This paper has proposed a time efficient algorithm for mining high utility item sets from a transaction data set.

ACKNOWLEDGMENT

This research work is approved by the SVITS as to improve with the current approaches in data mining using this method. Thus, the authors also wish to acknowledge institute administration for their guide & motivation during this research work.I would additionally like to offer gratitude to Mrs. Rupali Bhartiya for support in regards to the situational mindfulness framework & for building the methodology adjusted for this paper.

REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proc. of the 20th Int'l Conf. on Very Large Data Bases, pp. 487-499, 1994.
[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. In IEEE Transactions on Knowledge and Data Engineering, Vol. 21, Issue 12, pp. 1708- 1721, 2009.
[3] R. Chan, Q. Yang, and Y. Shen. Mining high utility itemsets. In Proc. of Third IEEE Int'l Conf. on Data Mining, pp. 19-26, Nov., 2003.
[4] A. Erwin, R. P. Gopalan, and N. R. Achuthan. Efficient mining of high utility itemsets from large datasets. In Proc. of PAKDD 2008, LNAI 5012, pp. 554-561.
[5] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In Proc. of the ACM-SIGMOD Int'l Conf. on Management of Data, pp. 1-12, 2000.
[6] 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.
[7] Y. Liu, W. Liao, and A. Choudhary. A fast high utility itemsets mining algorithm. In Proc. of the Utility-Based Data Mining Workshop, 2005.
[8] Piatetsky-Shapiro, G. (1991), Discovery, analysis, and presentation of strong rules, in G. Piatetsky-Shapiro & W. J. Frawley, eds,, Knowledge Discovery in Databases", AAAI/MIT Press, Cambridge.
[9] J. Pei, J. Han, and R. Mao. Closet: An efficient algorithm for mining frequent closed itemsets. In DMKD '00: ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pages 21–30, May 2000.
[10] M. J. Zaki and C.-J. Hsiao. Charm: An efficient algorithm for closed itemset mining. In SDM '02: Proceedings of the second SIAM International Conference on Data Mining, April 2002.
[11] G.Grahne and J. Zhu. Efficiently using prefix-trees in mining frequent itemsets. In FIMI '03: Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations, November 2003.

BIOGRAPHIES

Bharti Ahuja M.Tech Student, Department of CSE, SVITS, Indore India.

Mrs. Rupali Bhartiya Associate Professor, Department of CSE, SVITS, Indore, India.