International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 15 (2018) pp. 12326-12330
© Research India Publications. http://www.ripublication.com

An Efficient Algorithm for High Utility Pattern Mining from Transactional Databases

Vandan Tewari[1], **Anita Panwar**[2]

[1]*Assistant Professor, Department of Computer Engineering*

[2]*M.E. Scholar, Department of Computer Engineering*

Abstract

Main purpose of data mining is to find useful data set from raw data. Various data mining techniques are present. One of them is Frequent Pattern Mining technique which was used for find frequent patterns from databases. For usefulness of such frequent patterns, many constraints had been proposed by many researchers like utility parameters (price, profit, quantity etc.)as well as weight of an itemsets etc. Mining high utility patterns from transaction database mainly focuses on the utility value of an itemsets. Many algorithms have been proposed for finding user's goal previously, but they contain some limitations for large datasets when number of candidate itemsets are large. And when we talk about number of itemsets when large number of candidate itemsets are present as raw data, it degraded the performance of the algorithm in the terms of memory requirement and execution time. The most significant problem of utility mining is that these patterns do not satisfy anti-monotonicity property and hence mining high utility patterns using traditional association rule mining algorithm becomes difficult. Additionally when long transaction are considered the situation become worse. In this paper, we present a survey and comparison among various association rule mining algorithms which deals with high utility patterns mining are considered.

Keywords: Data Mining, Association Rule Mining, Interestingness Measure, Utility Mining.

INTRODUCTION

A. DATA MINING

Data mining is also known as knowledge discovery in database. It established a prominent and research area in recent years. The main goal of data mining is to mine useful data or we can also use information from the raw data. It has been used in different domain. Algorithmic process in which output is generated for the respective input, in the data mining same as algorithmic process input are taken in the form of dataset and output is generated in the form of High utility patterns.

B. ASSOCIATION RULE MINING

To apply Association Rules Mining and get rules from transactional databases is one of the research problems in data mining when the itemsets share framework. For finding frequent patterns and rules among the itemsets Association Rule Mining is the best algorithm. From the transaction dataset, itemsets which have support more than minimum support were found and rules with confidence having confidence more than user defined threshold are found as frequent patterns. In these algorithms various data structures are used. When the number of transaction dataset increases then it also increases its complexity, many newer data structures and algorithms are being developed to match this development.

Association Rule Mining process consist two steps. In first steps, from the dataset all frequent itemsets are found and in second step association rules with respect to the frequent patterns are generated.

C. UTILITY MINING

Utility Mining is shown as a new development in data mining technology. A pattern is of utility if it helps him in decision making. It is refers to allow a user to express his or her perspective concerning the usefulness and utility of patterns and At last find the patterns which have utility value higher than a user defined threshold. Utilities of patterns are used to describe the user's goal, it is described by utility based measures. Utility can be classified into two categories as follows:

Transaction Utility: It is the value or information which is directly from the transaction dataset e.g. Weight associated with the item.

External Utility: It is the utility which is given by the user, it is based on user interest for e.g. profit associated with item.

Normally the utility is defined as:

$$UOI = EU(e)^{*}\ IU(e) \qquad \ldots\ldots\ldots\ldots\ldots(1)$$

Where UOI stands for Utility of itemsets, EU (e) stands for External Utility and IU (e) stands for Internal Utility.

D. INTERESTINGNESS MEASURE

Interestingness measure [6] can play an important role in Utility Mining for fulfilling the user's goals. It depends on the utility (usefulness) of the itemsets. Interestingness Measure can be classified into three category as follows:

Objective Measure: It is mainly based only on the raw data. In it,user's knowledge and application knowledge is not required. Most of these measures are based on the theories in statistics, probability or information theory. For e.g. Apriori Algorithm considers only numbers and occurrence.

Subjective Measure: It is mainly based on both the data and user's of these data. User's domain is required in these measure along with the background knowledge about the data. This can be accessed by interacting with the user or by understanding the user's goals.For e.g. Utility Mining.

Semantic Measure: It considers the explanations as well as semantics of the patterns. Because It involves domain knowledge from the user hence researcher sometimes considers it as a special type of subjective measures.

PROBLEM STATEMENT

Utility Mining Problem

There is a transaction database D is given. Along with the dataset minimum utility threshold min utility" is also given here, the main objective is to discover all the itemsets which have high-utility. Let us consider the example database shown in Table I and in Table II, the profit is given with respect to the transaction dataset. In the transaction dataset, each value indicates the quantity sold for an item. The support and confidence calculated in table III using internal utility given in Table I and external utility given in Table IV

Table I. Transactional Table

Transaction ID	Item P	Item Q	Item R	Item S
Tr1	4	0	1	0
Tr2	2	0	0	6
Tr3	0	0	21	30
Tr4	3	0	0	5
Tr5	2	1	1	7
Tr6	5	1	3	11
Tr7	3	1	1	2
Tr8	2	2	2	9
Tr9	1	2	1	11
Tr10	6	1	1	10

Table II. Profit Table

Item Name	Profit
Item P	5
Item Q	100
Item R	38
Item S	1

Profit here represents the external utility measures which have been discussed earlier.If the minimum support is taken 40% , it can be observed that the frequent itemsets in Table 4.3 are S, P, PS, and R, but the four most profitable itemsets are PQRS, PQR, R, and RS, all of which are infrequent itemsets. Therefore it is not necessary that the itemsets which have high support also have high utility.

Table III. Table for Comparison between Support and Profit

Itemsets	Support	Profit
P	90	120
Q	60	800
R	80	1178
S	90	91
PQ	60	885
PR	70	495
PS	80	166
QR	60	1142
QS	60	850
RS	60	392
PQR	60	1227
PQS	60	935
PRS	60	477
QRS	60	1192
PQRS	60	1277

Another example shows that no anti-monotonicity properties is not satisfied in utility mining problem in which itemsets share framework. Let us consider another transactional database shown in Table IV and external utility table shown in Table V.

Table IV. Transactional Table

Transaction ID	I1	I2	I3	I4	I5	I6	I7
Tr1	2	0	2	0	2	0	0
Tr2	7	3	3	0	0	6	0
Tr3	2	2	2	3	7	0	6
Tr4	4	2	0	5	4	0	0
Tr5	3	2	0	3	0	3	0

Table IV shows the transactional dataset in which weight is given respect to the given item for various transactions. Table

V presents the profit corresponds to the given itemset in Table IV.

Table V. Profit Table

Item Name	Profit
I1	3
I2	5
I3	7
I4	4
I5	4
I6	3
I7	3

A itemset is high utility itemset or here represented by HUPSet which have utility value less than the predefined minimum threshold value.

High Utility Patterns are generated are shown below:

HUPset = I1; I2; I4, I1; I2; I4, I1; I4; I5, I1; I2; I4; I5, I2; I3; I4, I3; I4, I1; I2; I3; I4; I5; I7.

Here HUPset stands for High Utility patterns itemset. It can be observed that pattern are not anti-monotone, because subsets of the frequent patterns are also frequent but in the case of utility it is fail. Anti-monotonicity property is not applied for utility mining so it is a new approach for high utility pattern mining is proposed.

RELATED WORK DONE

A. APRIORI ALGORITHM

Agrawal et al. [1] propose an algorithm which is based on frequent pattern and also known as frequent pattern mining algorithm, named as Apriori Algorithm where target was found in second phase. In it, support measure is considered. The support is used for finding the finding the frequent patterns. If support measures of candidates are greater than minimum threshold value then the itemsets are frequent patterns. for mining frequent patterns, It is a very famous breadth-first algorithm , which scans the disk-resident database as many times as the maximum length of frequent patterns. However disadvantage of this popular algorithm is that it assumes transaction database are memory resident and it requires numerous database scans which increase time and space complexities. Anti-monotonicity property does not hold in Transaction dataset when we talk about share framework, for resolving this problem Tao et al. [11] proposed a new concept of weighted closure property.

Although weighted association rules mining considers the importance of items, in some application such as transaction databases items quantities in transaction are not considered.

B. TWO PHASE ALGORITHM

Ying Liu et al. [2] proposed a new algorithm for the same objective that discovering high utility patterns. Algorithm named as Two Phase Generation Algorithm. Two steps are presents in the proposed algorithm. It find the high utility pattern in the first phase and scan the data one or more time to identify the high utility pattern in second phase. The main shortcoming of this approach is that when the number of candidates are increases, the algorithm becomes inefficient in terms of space requirement.

C. ISOLATED ITEM DISCARDING (IDS)

To overcome memory insufficiency problems in Two Phase Algorithm a new algorithm was proposed by Lie et al. [4] for the same aim named Isolated Item Discarding Strategy (IDS). It is mainly proposed for the reduce the number of candidate generated in the first phase of Two Phase Algorithm. By pruning isolated item sets for HTWUIS (High Transaction Weighted Utilities) in first phase, it can be reduced. Due to several database scan time, it takes more time. And after it, using the candidate and test scheme to discover the high utility itemsets but still it also becomes inefficient in time.

D. IHUP TREE ALGORITHM

To efficiently generate high transaction weighted utilities in first phase and avoid database scan time, Ahmed et al.proposed a new algorithm based on a tree data structure. It is named as Tree I-HUP Algorithm, same as the name, these algorithm is based on the tree based data structure. Information about itemsets and their utility is maintained by these data structure. I-HUP Tree consist nodes, which consist of an item name and transaction weighted support count and utility value with it. This algorithm contains three steps. First is the I-HUP tree construction, second step is to generate the high transaction weighted utilities and third is identification of high utility itemsets. With compare to IDES and Two Phase candidate generation algorithm, THUP-Tree algorithm achieves the better performance. But still too many high utility transaction weighted itemsets are generated by this algorithm in phase one. Such a large number of high transaction weighted utility itemsets in first phase it causes performance degradation in terms of execution time and space requirement. Huge number of transaction weighted utilities in first phase also affect the second phase algorithm. Since more are high transaction weighted utilities itemsets are generated. Huge number of generated high utility transaction weighted utilities is a critical issue when we talk about the performance of the algorithm.

Huge number of transaction weighted utilities in first phase also affect the second phase algorithm. Since more are high transaction weighted utilities itemset are generated. Huge number of generated high utility transaction weighted utilities is a critical issue when we talk about the performance of the algorithm.

E. HUI-MINER ALGORITHM

Liu et al. [3] in their work proposed a new algorithm along with a new data structure. Utillity-list data structure is used in this algorithm. Algorithm is HUI-Miner algoritgm for the discovering high utility pattern. The utility-list of an itemsets stores its exact utility as well as an upper bound on the expected utility values of its supersets by using the remaining utility values stored in the list. The items are sorted and processed in the ascending order of transaction utility. The algorithm avoids the candidate verification and generation cost of itemset. On the other side the utility joining operation is very costly and hampers the overall performance of the algorithm.

F. FP GROWTH ALGORITHM

J. Han et al. [5] proposed a novel method for the same aim of discovering the high utility pattern from the databases. Frequent pattern tree (FP-tree) structure was proposed; FP-tree structure as an extended prefix tree structure for storing crucial information about frequent patterns into compressed structure proposed an extended prefix tree structure of frequent pattern tree. And also developed an efficient FP-tree based mining method that is frequent pattern were mined by the pattern fragment growth using the FP-Growth. In it, a new highly compact FP-tree are constructed, which is usually smaller than the original databases, since the databases scan cost is minimize in the subsequent mining process. It reduces cost of candidate generation by applying growth method. But FP Growth algorithm consume more memory and performs badly with long pattern dataset and therefore not able to find high utility patterns.

PROPOSED WORK

A novel algorithm has been proposed for discovering high utility patterns in single phase association rule mining which uses parameters such as statistical threshold based pruning. Pruning is used here for reducing the memory and time required for mining high utility itemsets.

Discovery of high utility patterns from a dataset is done by setting a threshold value which is often derived through several runs or experiments with the algorithm. If Utility of an itemset is less than minimum threshold utility than that itemset will be an uninteresting pattern. For finding high utility itmesets we can follows these steps:

First the algorithm takes input as Transaction dataset, External Utility value and Minimum threshold value. After that database projection is performed for the itemset and then identical transactions are merged. Each projection of database takes only linear time. Two upper bound methods are then applied on utility value. The upper bound is calculated and finally high utility itemsets are mined.

RESULT ANALYSIS

Testing has been done on the datasets as defined previously. Two datasets have been used and results are shown in which comparison is presented in terms of Memory requirement, Time requirement and number of Overlapped Patterns. The proposed and existing algorithms have been evaluated on two different samples. The graph have been plotted after testing on the different number of samples of datasets with different minimum support for Apriori Algorithm and minimum threshold for proposed algorithm. The above graph Fig.1 has been shown for time required to find the frequent itemsets.

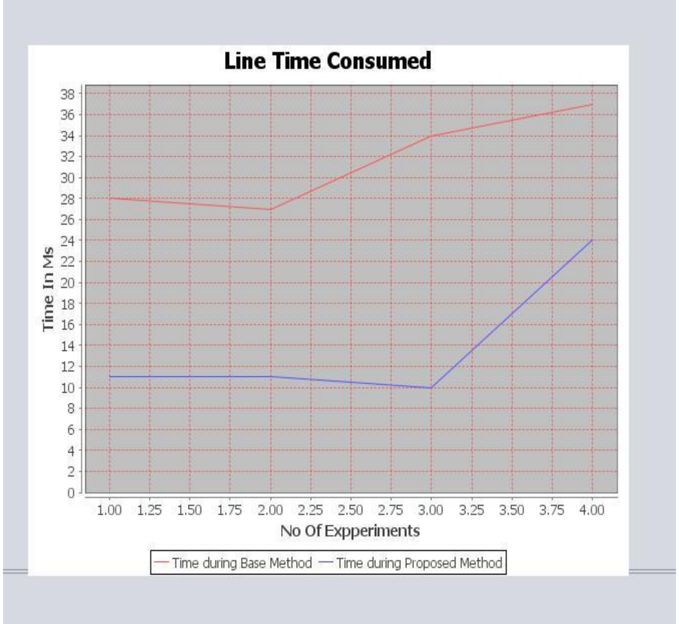

Figure 1. Graph for Time Required

Number of experiment represents the number of folds taken during bagging. Here '4' has been taken as number of folds. For the given number of folds time required for the proposed algorithm is less than the time required for based algorithm. It reduces time complexity in the proposed algorithm, and improves performance. The Fig.2 has been drawn between 'Memory ' and 'Number of Experiments'. Number of experiment represents the number of folds taken during bagging. Here '4' has been taken as number of folds. For the given number of folds memory requirement for the proposed algorithm is less than the time required for based algorithm. It reduces space complexity in the proposed algorithm, and improve performance. Figure.3 shows 'Number of overlapped patterns' vrs 'Number of Experiments'. For the given number of folds number of overlapped patterns for the proposed algorithm is less than the time required for based algorithm. It also improves performance of the algorithm.

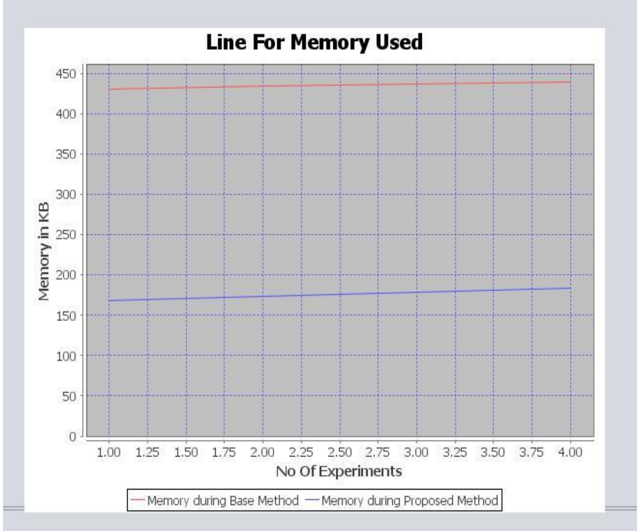

Figure 2. Graph for memory

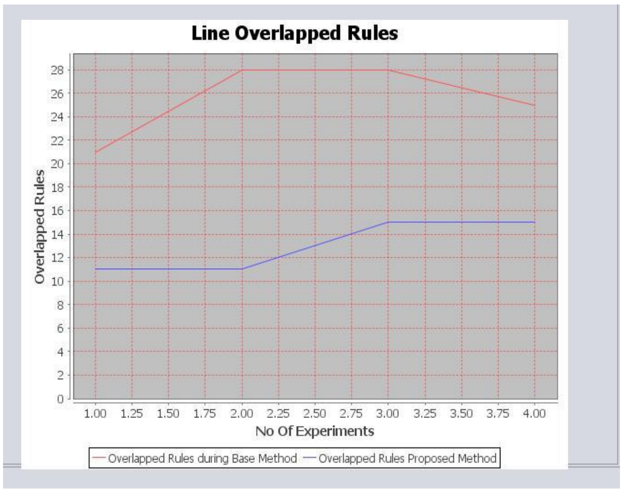

Figure 3. Graph for Number of overlapped patterns

CONCLUSION

In data mining, utility mining is a new approach in which mining results must meet user's goals. Existing algorithms of association rule mining do not consider interestingness measures for users. Previously many algorithms were proposed for frequent pattern mining, but most of them mainly based on the count or occurrence value of an itemset. In this project, a new approach for high utility pattern mining has been proposed which uses pruning and bagging methods to improve performance. Pruning has been used on minimum threshold value to reduce candidate itemsets while sampling with replacement using bagging method has been used to find best results. The proposed approach perform better in discovering the high utility patterns, it is shown in the experiments results, however memory required is sometimes depending on samples. As the proposed approach uses pruning for eliminating uninteresting patterns for reducing the time and memory required, it reduces the time but for different sample mempry requirement may change.

FUTURE WORK

In this paper, high utility patterns are mined to presents appropriate results to a User. Future work may focus on the changing memory requirement with the changing samples.

REFERENCES

[1] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in Proc. 20th Int. Conf. Very Large Databases, 1994, pp. 487 –499.

[2] Y. Liu, W. Liao, and A. Choudhary," A Two-Phase Algorithm for Fast Discovery of High Utility Itemsets" in Pro Conference on Knowledge Discovery and Data Mining 2005 pp. 253–262.

[3] M. Liu and J. Qu, "Mining high utility itemsets without candidate generation" IEEE Trans. Knowl. Data Eng., vol.28,. 2012, pp. 55 –64.

[4] Y.-C. Li, J.-S. Yeh, and C.-C. Chang, "Isolated Items Discarding Strategy for Discovering High Utility Itemsets," Data and Knowledge Eng., vol. 64, no. 1, pp. 198-217, Jan. 2008.

[5] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," Proc. ACM SIGMOD Int. Conf. Manage. Data , 2000, pp. 1 –12.

[6] L. Geng and H. J. Hamilton, "Interestingness measures for data mining: A survey" ACM Comput. Surveys, vol. 38, no. 3, p. 9, 2006.

[7] H. Yao, H. J. Hamilton, and L. Geng, "A unified framework for utility-based measures for mining itemsets," Proc. ACMSIGKDD 2nd Workshop Utility-Based Data Mining, 2006, pp. 28 –37.

[8] J. Han and Y. Fu, "Discovery of Multiple-Level Association Rules from Large Databases," Proc. 21th Intl Conf. Very Large Data Bases, pp. 420-431, Sept. 1995.

[9] C. F. Ahmed, S. K. Tanbeer,B. S. Jeong, and Young-Koo Lee "Efficient Tree Structures for High Utility Pattern Mining in Incre-mental Databases," IEEE Trans. on Knowl. and Data Engineering, VOL NO. 12, DECEMBER 2009 pp. 1708-1721, Sept. 1995.

[10] V. S. Tseng, B.-E. Shie, C.-W. Wu, and P. S. Yu, "Efficient algorithms for mining high utility itemsets from transactional databases," IEEE Trans. on Knowl. and Data Engineering, VOL. 25, NO. 8 pp. 1772-1786, Aug. 2013.