ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

AN ENHANCED UPGROWTH ALGORITHM FOR TEMPORAL HIGH UTILITY ITEM MINING

Ms. S. Gomathi*, Mrs. M. Suganya
* M.Phil Research Scholar, School of Computer Studies, Rathnavel Subramaniam College of Arts & Science,Sulur, Coimbatore - 402, TN, India
Assistant Professor, School of Computer Studies, Rathnavel Subramaniam College of Arts & Science,Sulur, Coimbatore - 402, TN, India

ABSTRACT

High utility item set mining from a transactional database helps to discover the items with high utility based on profit, cost and quantity. Even though many numbers of significant algorithms have been proposed in recent years they experienced the problem of producing a large number of candidate itemsets for high utility itemsets. Such a huge number of candidate item sets degrades and reduces the mining performance in terms of storage space requirement and execution time. The situation may become worse when the database contains lots of datasets, long transactions or long high utility itemsets. The proposal introduces two algorithms which are temporal utility pattern growth (TUP-Growth) and temporal UP-Growth+, for mining high utility itemsets with a set of effective strategies for pruning candidate item sets rapidly. The information of high utility itemsets is maintained in a tree-based data structure named utility pattern tree (TUP-Tree) such that candidate itemsets can be generated efficiently with only two scans of database, then that will be segmented into multiple clusters for fast computation. The proposed algorithms reduce the number of candidates and database scans effectively. This also outperforms best than the existing algorithms and significantly reduces the runtime and memory and storage overhead, especially when databases contain lots of high and long transactions.

KEYWORDS: Data Mining, Association Rules,TUP Growth Algorithm, Apriori Algorithm.

INTRODUCTION

Data mining area can be defined as efficiently discovering interesting rules from large databases. A new data mining issue, discovering sequential patterns from large scale databases, the input data is in form like set of sequences, called data-sequences. Each and every data sequence may have the list of transactions, where each transaction is a set of entities, called items. Naturally there is a transaction time allied with each transaction. A sequential pattern also consists of a list of sets of items. The issue is to find all sequential patterns with a user-specified minimum support, where the support of a sequential pattern is the certain percentage level of data-sequences that contain the pattern.[1].

Basic Issue in analyzing frequent items, Collection of events occurring close to each another one, Development of data mining techniques for time series data is an important problem of current interest, A frequent episode is one whose frequency exceeds a user specified threshold, The main computationally intensive step in frequent episode discovery is that of calculating the frequency of sets of candidate episodes.

In existing works Given a set of sequences, where each and every sequence consists of a list of elements and each element consists of a set of items, and given a user-specified minimum no of support threshold, sequential pattern mining is to find all frequent subsequences, that mean, the subsequences whose occurrence frequency in the set of sequences is no less than minimum support. Many previous studies contributed to the efficient mining of sequential patterns or other frequent patterns in time-related data, generalized from this definition of sequential patterns in to include time constraints, sliding time window, and user-defined taxonomy, and presented an a priori-based, in another work proposed inter transaction association rules that are implication rules whose two sides are totally-ordered episodes with timing-interval restrictions. [2].

In another work proposed the use of regular expressions as a flexible constraint specification tool

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

that enables user-controlled focus to be incorporated into the sequential pattern mining process. Some other studies extended the scope from mining sequential patterns to mining partial periodic patterns. In another journal introduced cyclic association rules that are essentially partial periodic patterns with perfect periodicity in the sense that each pattern reoccurs in every cycle, with 100 percent confidence. In another work developed a frequent pattern mining method for mining partial periodicity patterns that are frequent maximal patterns where each pattern appears in a fixed period with a fixed set of offsets and with sufficient support.

EXISTING SYSTEM

Mining high utility item sets from databases refers to finding the itemsets with high profits. Here, the meaning of item set utility is interestingness, importance, or profitability of an item to users.

Existing studies applied overestimated methods to facilitate the performance of utility mining. In these methods, potential high utility itemsets (PHUIs) are found first, and then an additional database scan is performed for identifying their utilities.

However, a huge set of PHUIs are generated and their mining performance is degraded consequently by existing methods.

When databases contain many long transactions or low thresholds are set, the situation may become worse.

A challenging problem to the mining performance is the huge number of PHUIs since more the PHUIs the algorithm generates, the higher processing time it consumes.

DISADVANTAGES:

1. Existing methods often generate a huge set of PHUIs and their mining performance is degraded consequently.
2. The huge number of PHUIs forms a challenging problem to the mining performance since the more PHUIs the generates, the higher processing time it consumes.

PROPOSED METHOD

The purpose of the proposed systems is towards finding high utility item set. If the items utility is no less than a user specified minimum utility threshold, then that item is known as high utility item set. Otherwise, the item is called a low-utility item set. The goal of frequent item set mining is to find items that co-occur in a transaction database above a user given frequency threshold, without considering the quantity or weight such as profit of the items. However, quantity and weight are significant for addressing real world decision problems that require maximizing the utility in an organization. The high utility item set mining problem is to find all item sets that have utility larger than a user specified value of minimum utility.

LITERATURE REVIEW

In existing work many patterns are identified by frequent sequential pattern mining algorithms. Most of them may not be informative to business decision-making, since they do not show the business value and impact. In some cases, such as fraud detection, some truly interesting sequences may be filtered because of their low frequencies. For example, in retail business, selling a car generally leads to much higher profit than selling a bottle of milk, while the frequency of cars sold is much lower than that of milk. In online banking fraud detection, the transfer of a large amount of money to an unauthorized overseas account may appear once in over one million transactions, yet it has a substantial business impact. Such problems cannot be tackled by the frequency/support framework. In the related area, utility is introduced into frequent pattern mining to mine for patterns of high utility by considering the quality (such as profit) of item sets. This has led to high utility pattern mining, which selects interesting patterns based on minimum utility rather than minimum support.

We identify the main challenge of adapting traditional association rule mining model in a weighted setting as the invalidation of the downward closure property, which is used to justify the efficient iterative process of generating and pruning large item sets from its subsets. In order to tackle this challenge, we made adaptation on the traditional association rule mining model under the significant weighted support metric framework instead of the large – support framework used in previous works. In this new proposed model, the iterative generation and pruning of significant item sets is justified by a weighted downward closure property.[3].

Earlier association works identify the limitation of the traditional Association Rule Mining model, in particular, its incapacity for treating units differently. We proposed that weight can be integrated in the mining process to solve this problem. We identify the

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

challenge faced when making improvement towards using weight, in particular the invalidation of downward closure property. A set of new concepts are proposed to adapt weighting in the new setting. Among them is the proposal of using "weighted downward closure property" as a replacement of the original downward closure property. This is proved as valid and justifies the effective mining strategy in the new framework of weighted support significant. The new framework is designed to replace the original "support – large" framework in order to tackle the problem in weighted settings. [9]

A high utility item set mining [4] identifies item sets whose utility satisfies a given threshold. It allows users to quantify the usefulness or preferences of items using different values. Thus, it reflects the impact of different items. High utility itemsets mining is useful in decision-making process of many applications, such as retail marketing and Web service, since items are actually different in many aspects in real applications. However, due to the lack of "downward closure property", the cost of candidate generation of high utility itemsets mining is intolerable in terms of time and memory space. This paper presents a Two-Phase algorithm which can efficiently prune down the number of candidates and precisely obtain the complete set of high utility itemsets. The performance of our algorithm is evaluated by applying it to synthetic databases and two real-world applications. It performs very efficiently in terms of speed and memory cost on large databases composed of short transactions, which are difficult for existing high utility item sets mining algorithms to handle. Experiments on real-world applications demonstrate the significance of high utility item sets in business decision-making, as well as the difference between frequent item sets and high utility item sets.

This paper addresses the discovery of temporal utility and weighted item sets from transactional weighted data sets. To address the high utility item mining issue, the TUI-support measure is defined as a weighted frequency of occurrence of an item set in the analyzed data for fast.[13]

CONTRIBUTION OF THE PROPOSED SYSTEM

The Proposed strategies can not only decrease the overestimated utilities of PHUIs but greatly reduce the number of candidates. Different types of both real and synthetic data sets are used in a series of experiments to the performance of the proposed algorithm with state-of-the-art utility mining algorithms. Experimental results show that UP-Growth [5] and UP-Growth+ outperform other algorithms substantially in term of execution time, especially when databases contain lots of long transactions or low minimum utility thresholds are set.[10]

ADVANTAGES:

1. Two algorithms, named Temporal Utility pattern growth(TUP Growth)and TUP-Growth+, and a compact and segmented tree structure, called temporal utility pattern tree(TUP-Tree),for discovering high utility item sets based on the segmented dataset and maintaining important information related to utility patterns within databases are proposed.
2. High-Utility item sets can be generated from TUP-Tree efficiently with only two scans of original databases. Several strategies are proposed for facilitating the mining process of TUP-Growth+ by maintaining only essential information in TUP-Tree.
3. By these Strategies, overestimated utilities of candidates can be well reduced by discarding utilities of the items that cannot be high utility or are not involved in search space.
4. Reduces the cost and time by performing less number of scans in the database.

TUP – GROWTH TREE ALGORITHM

Steps

a. Initially assign 0 for F.
   i. F=0
b. For each item I in the header tree table UPTree.
   I=prefix U{i}-generate a new itemset I by joining prefix and I with TUI support set to the TUI support item i
c. If I is utility
   i. Store I.
d. End if
e. If TUI-support(I) >= St then
   i. F=FU{I}
f. End if
g. Conditional_pattern (P)=generate(UPTree, ,I)
h. $UPTree_I$=createUP -tree(Conditional_pattern)
i. Perform pruning

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

i. Prune=identify(UPTree I, St)
ii. Htee=remove(UPTreeI, prune)

j. If UPTreeI #0 then
i. F=F U TUIMining(UPTree, St, I)

k. End
l. Return the output F.

PERFORMANCE EVALUATION

To evaluate the efficiency, four measures were used to evaluate the effectiveness. One is the number of modified entries, indicating how much the content of the original database is preserved. The other measures are defined as follows: Let H and U are the sets of all frequent items and all strong rules in the original database, respectively. After frequent pattern mining, let the set of all strong rules in the modified database be denoted as U'. Moreover, let SR, LR, and FR, respectively, denote the sets of all the frequent items that fail to be recalculated, all the loss rules, and all the false rules. The number of rules in any notation R is denoted as |R|.[7][8].

For all the ratios, the lower they are, the better this approach performs. The scalability of this approach is first evaluated in terms of the database size, the number of frequent items, and the number of strong rules, respectively. After that, the frequent items are selected in such a way that all of them have at least one item in common to evaluate the effectiveness of this approach on the four measures. Finally, the overlapping degree of a rule is defined and experiments were made to observe how the correlation among rules influences the performance.[12]

SCALABILITY APPROACH

The processing time reported includes the CPU time consumed in the preprocessing steps (after frequent items have been selected), the template generation, and the complete process for calculating frequent items. The I/O time spent on the index construction and the database modification is excluded in order to highlight the impact of the database scale on this indexing mechanisms and the proposed method for frequent pattern mining. The proposed experiments uses 25688 entries.[6].

Transactions(hundreds)	UP Growth	TUP Growth +
1	1.30	0.5
3	2.03	1.02
5	6.0	4.5
10	9.0	6.5

Table 8.a Scalability on the database

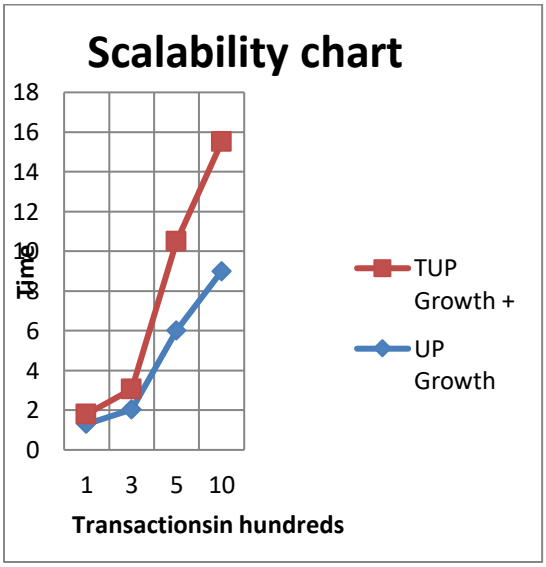

Fig 8.b Scalability on the database size

The above chart describes the comparison of TUP growth and UP growth with the consideration of scalability. The system compares the number of transaction considered by the algorithm for high utility item finding. The dataset taken for the experiments are given in thousands. For 1000 transaction the system takes 2 seconds for computing. The chart concludes that the proposed work can handle more number of transactions than existing system.

Transactions (thousands)	UP Growth	UP Growth +	TUP Growth +
1	**1.30**	0.5	**0.3**
3	**2.03**	1.02	**1.00**
5	**6.0**	4.5	**3.9**
10	**9.0**	6.5	**5.4**

Fig 8.c Time based on the database

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

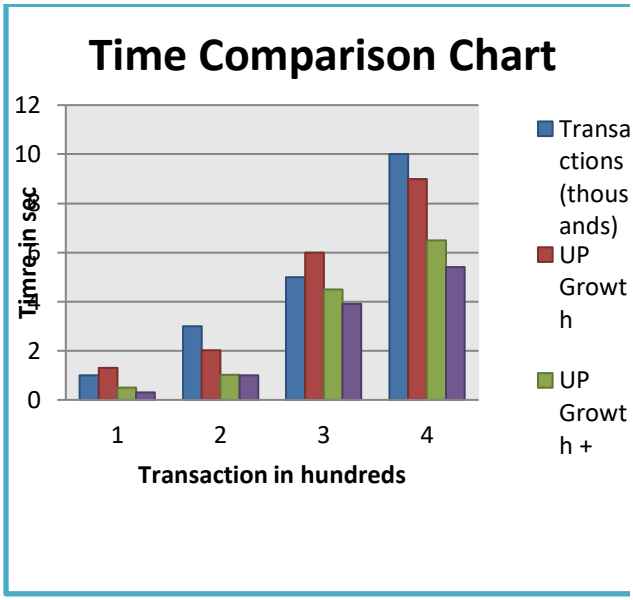

Fig8. d Time based on the database size

The above chart describes the comparison of TUP growth+, UP growth TUP growth and UP growth+ with the consideration of time. The system compares the proposed system with the existing system by the time. The dataset taken for the experiments are given in thousands. For 1000 transaction the system takes 0.7 seconds for computing. The chart concludes that the proposed work can handle more number of transactions in less time than existing system.

CONCLUSION

The proposed system applies the Apriori algorithm for effective frequent item set mining. The main drawback of the Apriori is the lack of memory and slow. So instead of showing the dataset in apriori alone the proposed system uses UP growth based implementation. The system implemented the UP-growth method and effective warehousing concepts. This studied through the performance in comparison with several existing utility pattern mining algorithms in large databases. The performance study shows that the proposed method mines both short and long patterns efficiently in large databases and also concentrated on high dimensional data storage problem. The system outperforms with the current candidate pattern generation-based algorithms effectively.

FUTURE ENHANCEMENTS

The current study proposed two definitions to capture the effects of the noise in the data. This pointed out possible scenarios where the mining of these patterns is central as well as the challenges in developing efficient mining algorithms. Future works include the extension of the temporal utility pattern tree to mine noisy patterns, and developing more efficient techniques to handle genomic data.

REFERENCES

1. Tseng, Vincent S., et al. "Efficient algorithms for mining high utility item sets from transactional databases." Knowledge and Data Engineering, IEEE Transactions on 25.8 (2013): 1772-1786.
2. H. F. Li, H. Y. Huang, Y. C. Chen, Y. J. Liu and S. Y. Lee, “Fast and Memory Efficient Mining of High Utility Item sets in Data Streams,” in Proc. of the 8th IEEE Int'l Conf. on Data Mining, pp. 881-886, 2008.
3. B.-E. Shie, V. S. Tseng and P. S. Yu, “Online mining of temporal maximal utility itemsets from data streams,” in Proc. of the 25th Annual ACM Symposium on AppliedComputing, Switzerland, Mar., 2010.
4. S.Shankar, T.P.Purusothoman, S. Jayanthi,N.Babu, “A fast algorithm for mining high utility itemsets” ,in :Proceedings of IEEE International Advance Computing Conference (IACC 2009), Patiala, India, pp.1459-1464.
5. R. Agrawal and R. Srikant. “Fast algorithms for mining association rules,” in Proc. of the 20th VLDB Conf., pp. 487-499, 1994.
6. J. Han, J. Pei, 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.
7. V. S. Tseng, C.-W.Wu, B.-E.Shie and P. S. Yu, “UPGrowth: An Efficient Algorithm for High Utility Itemsets Mining,” in Proc. of the 16th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD 2010), pp. 253-262, 2010.
8. J. Hu, A. Mojsilovic, “High-utility pattern mining: Amethod for discovery of high-utility item sets”, PatternRecognition 40 (2007) 3317 – 3324.
9. Chen,“Mining frequent itemsets from data streams with a time sensitive sliding window,” in Proc. of the SIAM Int’lConference on Data Mining (SDM 2005), 2005

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

10. Y. Liu, W. Liao and A. Choudhary, "A fast high utility itemsets mining algorithm," in Proc. of the Utility-BasedData Mining Workshop, 2005
11. B.-E. Shie, V. S. Tseng and P. S. Yu, "Online mining of temporal maximal utility itemsets from data streams," in Proc. of the 25th Annual ACM Symposium on AppliedComputing, Switzerland, Mar., 2010
12. K. Sun and F. Bai, "Mining Weighted Association Rules without Preassigned Weights," IEEE Trans. OnKnowledge and Data Engineering, Vol. 20, No. 4, 2008.
13. S. K. Tanbeer, C. F. Ahmed, B.-S.Jeong and Y.-K.Lee,"Efficient frequent pattern mining over data streams," in Proc. of the ACM 17th Conference on Information andKnowledge Management, 2008.