ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Infrequent Weighted Item Set Mining Using Frequent Pattern Growth

Sahu Smita Rani
Assistant Professor, & HOD,
Dept of CSE,
Sri Vaishnavi College of Engineering.

D.Vikram Lakshmikanth
Assistant Professor,
Dept of CSE,
Sri Vaishnavi College of Engineering.

Yalla Sateesh Kumar
M.Tech,
Dept of CSE,
Sri Vaishnavi College of Engineering.

Abstract:

Association rule mining (ARM) identifies frequent item sets from databases and generates association rules by considering each item in equal value. However, items are actually different in many aspects in a number of real applications , such as retail marketing, network log, etc. The difference between items makes a strong impact on the decision making in these applications. Therefore, traditional ARM cannot meet the demands arising from these applications. By considering the different values of individual items as utilities, utility mining focuses on identifying the item sets with high utilities. As "downward closure property" doesn't apply to utility mining, the generation of candidate item sets is the most costly in terms of time and memory space. We propose two algorithms, namely utility pattern growth (UP-Growth) and UP-Growth+, for mining high utility item-sets with a set of effective strategies for pruning candidate item-sets. The information of high utility item-sets is maintained in a tree-based data structure named utility pattern tree (UP-Tree) such that candidate item-sets can be generated efficiently with only two scans of database. The performance of UP-Growth and UP-Growth+ is compared with the state-of-the-art algorithms on many types of both real and synthetic data sets. Experimental results show that the proposed algorithms, especially UP-Growth+, not only reduce the number of candidates effectively but also outperform other algorithms substantially in terms of runtime, especially when databases contain lots of long transactions.

Index Terms:

Candidate pruning, frequent item set, high utility itemset, utility mining, data mining.

1.INTRODUCTION:

Data mining is one of the best analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorizes it, and summarizes it into useful information.

Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases. Data mining commonly encompasses a variety of algorithms namely clustering, classification, association rule mining, regression, summarization and prediction. Among these algorithms, Association rules mining (ARM) is one of the most widely used techniques in data mining and knowledge discovery and has tremendous applications in business, science and other domains. The main objective of ARM is to identify frequently occurring patterns of item sets. It first finds all the item sets whose co- occurrence frequency are beyond a minimum support threshold, and then generates rules from the frequent item sets based on a minimum confidence threshold. Traditional ARM model treat all the items in the database equally by only considering if an item is present in a transaction or not.Discovering useful patterns hidden in a database plays an essential role in several data mining tasks, such as frequent pattern mining, weighted frequent pattern mining, and high utility pattern mining.

Among them, frequent pattern mining is a fundamental research topic that has been applied to different kinds of databases, such as transactional databases, streaming databases, and time series databases, and various application domains, such as bioinformatics, Web click-stream analysis, and mobile environments.Nevertheless, relative importance of each item is not considered in frequent pattern mining. To address this problem, weighted association rule mining was proposed. In this framework, weights of items, such as unit profits of items in transaction databases, are considered. With this concept, even if some items appear infrequently, they might still be found if they have high weights. However, in this framework, the quantities of items are not considered yet. Therefore, it cannot satisfy the requirements of users who are interested in discovering the item sets with high sales profits, since the profits are composed of unit profits, i.e., weights, and purchased quantities.In view of this, utility mining emerges as an important topic in data mining field. Mining high utility item sets from databases refers to finding the item sets with high profits.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Here, the meaning of item set utility is interestingness, importance, or profitability of an item to users. Utility of items in a transaction database consists of two aspects: 1) the importance of distinct items, which is called external utility, and 2) the importance of items in transactions, which is called internal utility. Utility of an item set is defined as the product of its external utility and its internal utility. An item set is called a high utility item set if its utility is no less than a user-specified minimum utility threshold; otherwise, it is called a low-utility item set.Mining high utility item sets from databases is an important task has a wide range of applications such as website click stream analysis business promotion in chain hypermarkets, cross-marketing in retail stores online e-commerce management, mobile commerce environment planning and even finding important patterns in biomedical applications. However, mining high utility item sets from databases is not an easy task since downward closure property in frequent item set mining does not hold. In other words, pruning search space for high utility item set mining is difficult because a superset of a low-utility item set may be a high utility item set. A naı¨ve method to address this problem is to enumerate all item sets from databases by the principle of exhaustion. Obviously, this method suffers from the problems of a large search space, especially when databases contain lots of long transactions or a low minimum utility threshold is set. Hence, how to effectively prune the search space and efficiently capture all high utility item sets with no miss is a crucial challenge in utility mining.

Two algorithms, named utility pattern growth(UP-Growth) and UP-Growth+ ,and a compact tree structure, called utility pattern tree (UP-Tree), for discovering high utility item sets and maintaining important information related to utility patterns within databases are proposed. High-utility item sets can be generated from UP-Tree efficiently with only two scans of original databases.Several strategies are proposed for facilitating the mining processes of UP-Growth and UP-Growth by maintaining only essential information in UP-Tree. 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 the search space. The proposed strategies can not only decrease the overestimated utilities of PHUIs but also greatly reduce the number of candidates.Different types of both real and synthetic data sets are used in a series of experiments to compare the performance of the proposed algorithms with the state-of-the-art utility mining algorithms.

Experimental results show that UP-Growth and UP-Growth+ outperform other algorithms substantially in terms of execution time, especially when databases contain lots of long transactions or low minimum utility thresholds are set.

2.RELATED WORK:

Extensive studies have been proposed for mining frequent patterns. Among the issues of frequent pattern mining, the most famous are association rule mining and sequential pattern mining. One of the well-known algorithms for mining association rules is Apriori, which is the pioneer for efficiently mining association rules from large databases. Pattern growth-based association rule mining algorithms such as FP-Growth were afterward proposed. It is widely recognized that FP-Growth achieves a better performance than Apriori-based algo-rithms since it finds frequent itemsets without generating any candidate itemset and scans database just twice. In the framework of frequent itemset mining, the importance of items to users is not considered. Thus, the topic called weighted association rule miningwas brought to attention. First proposed the concept of weighted items and weighted association rules. However, since the framework of weighted association rules does not have downward closure property, mining performance cannot be improved. To address this problem, Tao et al. proposed the concept of weighted downward closure property.

By using transaction weight, weighted support can not only reflect the importance of an itemset but also maintain the down-ward closure property during the mining process. There are also many studies that have developed different weighting functions for weighted pattern mining.Although weighted association rule mining considers the importance of items, in some applications, such as transac-tion databases, items' quantities in transactions are not taken into considerations yet. Thus, the issue of high utility itemset mining is raised and many studies have addressed this problem. Liu et al. proposed an algorithm named Two-Phase which is mainly composed of two mining phases. In phase I, it employs an Apriori-based level-wise method to enumerate HTWUIs. Candidate itemsets with length k are generated from lengthk-1 HTWUIs, and their TWUs are computed by scanning the database once in each pass. After the above steps, the complete set of HTWUIs is collected in phase I.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

In phase II, HTWUIs that are high utility itemsets are identified with an additional database scan. Although two-phase algorithm reduces search space by using TWDC property, it still generates too many candi-dates to obtain HTWUIs and requires multiple database scans. To overcome this problem, Li et al. proposed an isolated items discarding strategy (IIDS) to reduce the number of candidates. By pruning isolated items during level-wise search, the number of candidate itemsets for HTWUIsin phaseI-canbereduced However,this algorithm still scans database for several times and uses a candidate generation-and-test scheme to find high utility itemsets. To efficiently generate HTWUIs in phase I and avoid scanning database too many times, Ahmed et al. proposed a tree-based algorithm, named IHUP. A tree-based structure called IHUP-Tree is used to maintain the information about itemsets and their utilities.

Each node of an IHUP-Tree consists of an item name, a TWU value and a support count. IHUP algorithm has three steps: 1) con-struction of IHUP-Tree, 2) generation of HTWUIs, and 3) identification of high utility itemsets. In step 1, items in transactions are rearranged in a fixed order such as lexicographic order, support descending order or TWU descending order. Then the rearranged transactions are inserted into an IHUP-Tree. Fig. 1 shows the global IHUP-Tree for the database in Table 1, in which items are arranged in the descending order of TWU. For each node in Fig. 1, the first number beside item name is its TWU and the second one is its support count. In step 2, HTWUIs are generated from the IHUP-Tree by applying FP-Growth. Thus, HTWUIs in phase I can be found without generating any candidate for HTWUIs. In step 3, high utility itemsets and their utilities are identified from the set of HTWUIs by scanning the original database once.

UP-Growth algorithm

Input: UP-Tree T_x, Header Table HT_x, minimum utility threshold t, Item set $I = \{i_1, i_2, .., i_k\}$.

Process:

1. For each entry i_k in HT_x do
2. Trace links of each item. And calculate sum of node utilitynu_{sum}.
3. If $nu_{sum} \geq$ t
4. Generate Potential High Utility Itemset (PHUI) $Y = X \cup i_k$
5. Put Potential Utility of i_k as approximated utility of Y
6. Construct Conditional Pattern Based HT_Y.
7. Put local promising items intoHT_Y.
8. Apply Discarding Local Unpromising (DLU) to minimize path utilities of paths.
9. Apply DLU with *Insert_Recognized_Path* to insert path intoT_Y.
10. If $T_Y \neq \emptyset$ then call to UP-Growth.
11. End if
12. End for.

Output: All PHUI's in T_x

UP-Growth +:

•In UP-Growth, minimum item utility table is used to reduce the overestimated utilities.

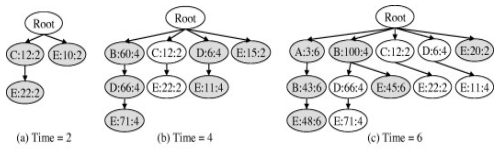
In UP-Growth+ algorithm we replace Discarding Local Unpromising (DLU) with Discarding Node Utility (DNU), DLN is replace with Decreasing local Node utilities for the nodes of local UP-Tree (DNN) and Insert_Recognized_Path is replace by Insert_Recognized_Path_miu When a path is retrieved, minimal node utility of each node in the path is also retrieved in the data mining process .

1.A novel algorithm, called UP-Growth (Utility Pattern Growth), is proposed for discovering high utility item sets. Correspondingly, a compact tree structure, called UP-Tree (Utility Pattern Tree), is proposed to maintain the important information of the transaction database related to the utility patterns. High utility item sets are then generated from the UP-Tree efficiently with only two scans of the database.

2.Four strategies are proposed for efficient construction of UP-Tree and the processing in UP-Growth. By these strategies, the estimated utilities of candidates can be well reduced by discarding the utilities of the items which are impossible to be high utility or not involved in the search space. The proposed strategies can not only efficiently decrease the estimated utilities of the potential high utility item sets but also effectively reduce the number of candidates.

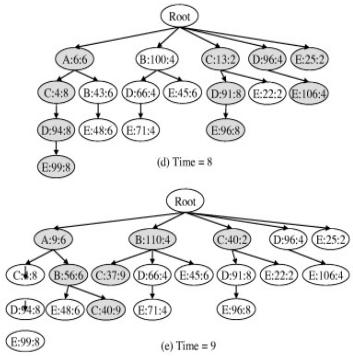
3.Both of synthetic and real datasets are used in experimental evaluations to compare the performance of UP-Growth with the state-of-the-art utility mining algorithms. The experimental results show that UP-Growth outperforms other algorithms substantially in terms of execution time, especially when the database contains lots of long transactions.

When data sets go beyond a single storage capacity, it is necessary to distribute them to multiple independent computers. Trans-computer network storage file management system is called distributed file system. A typical Hadoop distributed file system contains thousands of servers, each server stores partial data of file system.



(a) Time = 2 (b) Time = 4 (c) Time = 6

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal



System architecture:

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.

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 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 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.

Advantages:

1.Two algorithms, named Utility pattern growth(UP Growth)and UP-Growth+, and a compact tree structure, called utility pattern tree(UP-Tree),for discovering high utility item sets and maintaining important information related to utility patterns within databases are proposed.

2.High-Utility item sets can be generated from UP-Tree efficiently with only two scans of original databases. Several strategies are proposed for facilitating the mining process of UP-Growth+ by maintaining only essential information in UP-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.

Aims and Objectives:

In this project we have main aim is to present improved methods UP-Growth and UP-Growth+ with aim of improving its performance in terms of scalability and time:

-To present literature review different methods of frequent set mining over transactional datasets.

-To present the present new framework and methods.

-To present the practical simulation of proposed algorithms and evaluate its performances.

To present the comparative analysis of existing and proposed algorithms in order to claim the efficiency

The Proposed Mining Method:

UP-Tree, a basic method for generating PHUIs is to mine UP-Tree by FP-Growth. However too many candidates will be generated. Thus, we propose an algorithm UP-Growth by pushing two more strategies into the framework of FP-Growth. By the strategies, overestimated utilities of item sets can be decreased and thus the number of PHUIs can be further reduced. To address this issue, we propose two novel algorithms as well as a compact data structure for efficiently discovering high utility item sets from transactional databases. Major contributions of this work are summarized as follows:

1. Two algorithms, named utility pattern growth (UP Growth) and UP-Growth+ , and a compact tree structure, called utility pattern tree (UP-Tree), for discovering high utility item sets and maintaining important information related to utility patterns within databases are proposed. High-utility item sets can be generated from UP-Tree efficiently with only two scans of original databases.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

2. Several strategies are proposed for facilitating the mining processes of UP-Growth and UP-Growth by maintaining only essential information in UP-Tree. 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 the search space. The proposed strategies can not only decrease the overestimated utilities of PHUIs but also greatly reduce the number of candidates.

3. Different types of both real and synthetic data sets are used in a series of experiments to compare the performance of the proposed algorithms with the state-of-the-art utility mining algorithms. Experimental results show that UP-Growth and UP-Growth+ outperform other algorithms substantially in terms of execution time, especially when databases contain lots of long transactions or low minimum utility thresholds are set.

Pu(<ADC>;{B}-CPB)=10miu(A) X <ADC>:count =105X1=5 and
Pu(<ADC>;{B}-CPB)=32 miu(A) X <ADE>:count=325X1=27:

Improved Mining Method: UP-Growth+:

UP-Growth achieves better performance than FP-Growth by using DLU and DLN to decrease overestimated utilities of item sets. However, the overestimated utilities can be closer to their actual utilities by eliminating the estimated utilities that are closer to actual utilities of unpromising items and descendant nodes. In this section, we propose an improved method, named UP-Growth+, for reducing overestimated utilities more effectively.
In UP-Growth, minimum item utility table is used to reduce the overestimated utilities. In UP-Growth+, minimal node utilities in each path are used to make the estimated pruning values closer to real utility values of the pruned items in database.

UP-Growth algorithm

Input: UP-Tree , Header Table , minimum utility threshold , Item set = { 1, 2, . , }. Process:

1.For each entry in do
2.Trace links of each item. And calculate sum of node utility .
3.If ≥ t
4.Generate Potential High Utility Item set (PHUI)
5.Put Potential Utility of as approximated utility of
6.Construct Conditional Pattern Based .
7.Put local promising items into .
8.Apply Discarding Local Unpromising (DLU) to minimize path utilities of paths.
9.Apply DLU with _ _ h to insert path into .
10.If ≠ then call to UP-Growth.
11.End if
12.End for.

Output: All PHUI's in

3.Conclusion And Future Work

In this paper, we have proposed two efficient algorithms named UP-Growth and UP-Growth+ for mining high utility item sets from transaction databases. For maintaining the information of high utility item sets a data structure named UP-Tree was proposed. With only two database scans, from UP-Tree Potential high utility item sets can be efficiently generated. To perform a thorough performance evaluation both real and synthetic datasets were used in the experiments. Results show that the strategies considerably improved performance by reducing both the search space and the number of candidates. we have proposed two efficient algorithms named UP-Growth and UP-Growth+ for mining high utility item sets from transaction databases. For maintaining the information of high utility item sets a data structure named UP-Tree was proposed. With only two database scans, from UP-Tree Potential high utility item sets can be efficiently generated. To perform a thorough performance evaluation both real and synthetic datasets were used in the experiments. Results show that the strategies considerably improved performance by reducing both the search space and the number of candidate.

REFERENCES:

R[1] R. Agrawal and R. Srikant, "Fast Algorithms for MiningAssociation Rules,"Proc. 20th Int'l Conf. Very Large Data Bases(VLDB),pp. 487-499, 1994.

[2] R. Agrawal and R. Srikant, "Mining Sequential Patterns,"Proc.11th Int'l Conf. Data Eng.,pp. 3-14, Mar. 1995.

[3] C.F. Ahmed, S.K. Tanbeer, B.-S. Jeong, and Y.-K. Lee, "EfficientTree Structures for High Utility Pattern Mining

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

in IncrementalDatabases,"IEEE Trans. Knowledge and Data Eng., vol. 21, no. 12,pp. 1708-1721, Dec. 2009.

[4] C.H. Cai, A.W.C. Fu, C.H. Cheng, and W.W. Kwong, "Mining Association Rules with Weighted Items,"Proc. Int'l Database Eng.and Applications Symp. (IDEAS '98),pp. 68-77, 1998.

[5] R. Chan, Q. Yang, and Y. Shen, "Mining High Utility Itemsets,"Proc. IEEE Third Int'l Conf. Data Mining,pp. 19-26, Nov. 2003.

[6] J.H. Chang, "Mining Weighted Sequential Patterns in a SequenceDatabase with a Time-Interval Weight,"Knowledge-Based Systems,vol. 24, no. 1, pp. 1-9, 2011.

[7] M.-S. Chen, J.-S. Park, and P.S. Yu, "Efficient Data Mining for PathTraversal Patterns,"IEEE Trans. Knowledge and Data Eng.,vol. 10,no. 2, pp. 209-221, Mar. 1998.

[8] C. Creighton and S. Hanash, "Mining Gene Expression Databasesfor Association Rules," Bioinformatics,vol. 19, no. 1, pp. 79-86,2003.

[9] M.Y. Eltabakh, M. Ouzzani, M.A. Khalil, W.G. Aref, and A.K.Elmagarmid, "Incremental Mining for Frequent Patterns inEvolving Time Series Databases," Technical Report CSD TR#08-02, Purdue Univ., 2008.

[10] A. Erwin, R.P. Gopalan, and N.R. Achuthan, "Efficient Mining of High Utility Itemsets from Large Data Sets,"Proc. 12th Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining (PAKDD), pp. 554-561, 2008