ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 2, Special Issue 12, April 2016.

Rare utility itemset mining without candidate generation

T.Padmavathy[1], Wisely Joe[2]

[1]*PG Student, Department of CSE, St Peter's College of engineering and technology, Chennai.*

[2]*Assistant Professor, Department of CSE, St Peter's College of engineering and technology, Chennai.*

[1]padhu29@gmail.com

[2] wiselybritto@gmail.com

ABSTRACT : *Frequent itemset mining focuses on mining itemsets which occur often in a transactional database. These itemsets are profitable and useful in business decision making and satisfies minimum support threshold set by the user. However infrequent itemset mining is a variation of frequent itemset mining where it finds the uninteresting patterns. This paper deals with the problem of determining rare utility itemsets. The term utility refers to the importance or the usefulness of the appearance of the itemset in transactions and is quantified in terms like profit, sales, quantity or any other user preferences. Many algorithms have been proposed which produces large number of candidate itemsets which in turn degrades the performance. FP-Growth algorithm offers for efficient and effective mining of frequent itemset from databases without candidate generation. In this paper based on this FP-Growth algorithm a new algorithm is defined which takes both frequency and utility as parameter to identify rare high utility itemsets.*

Keywords: ARM, FP-growth, Infrequent itemset mining, RUI-Miner , Utility mining,

1. INTRODUCTION

1.1 Data Mining

Data mining is the process of finding correlations or patterns among dozens of fields in large relational databases. Data mining techniques can be categorized according to the objectives they follow and the results they offer, which obtains computer as a tool and makes use of the skill and knowledge significance to comprehend and explain the problem. Various data mining techniques such as, decision trees, association rules, and neural networks are already presented and become the point of attention for several years. Association rule mining technique is the most efficient data mining technique to search hidden or desired pattern among the huge amount of data. It is responsible to get correlation relationships among various data attributes in a large set of items in a database. Itemset mining is an investigative data mining technique widely used for discovering important correlations among data. The first challenge to perform itemset mining [1] was focused on discovering frequent itemsets, i.e., patterns whose observed frequency of occurrence in the source data (the support) is above a given threshold. many traditional approaches ignore the influence/interest of each item/transaction within the analyzed data. To allow treating items/transactions differently based on their relevance in the frequent itemset mining process, the notion of weighted itemset has also been introduced [4],

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 2, Special Issue 12, April 2016.

[5], [6]. A weight is associated with each data item and characterizes its local significance within each transaction.

1.2 Association Rule Mining

ARM is primarily focused on finding frequent co-occurring associations among a collection of items. It aims at extracting interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories. Association rules are widely used in various areas such as telecommunication networks, market and risk management, inventory control. There are two important basic measures for association rules, support(s) and confidence(c). Support(s) of an association rule is defined as the percentage/fraction of records that contain X ∪ Y to the total number of records in the database. Confidence of an association rule is defined as the percentage/fraction of the number of transactions that contain X∪Y to the total number of records that contain X. Association rule mining is to find out association rules that satisfy the predefined minimum support and confidence from a given database. The problem is decomposed into two sub problems. One is to find those itemsets whose occurrences exceed a predefined threshold in the database and those itemsets are called frequent or large itemsets. The second problem is to generate association rules from those large itemsets with the constraints of minimal confidence.[1] When mining association rules there are mainly two problems to deal with. First there is a algorithmic complexity. The number of rules grows exponentially with the number of items. Second interesting rule must be picked from the set of generated rule. This is quite costly because the generated rule sets are quite large.[2]

1.3 Frequent Pattern Mining

Frequent itemsets play an essential role in many data mining tasks that try to find interesting patterns from databases, such as association rules, correlations, sequences, episodes, classifiers, clusters and many more of which the mining of association rules is one of the most popular problems. This is first proposed by Agrawal, Imielinski, and Swami [AIS93] in the context of frequent itemsets and association rule mining. The application of frequent pattern mining is Basket data analysis, cross-marketing, catalog design, sale campaign analysis, web log analysis, and DNA sequence analysis. FPM discloses an intrinsic and important property of data mining tasks. The unique challenges in discovering frequent patterns are in three-fold. First, frequent pattern mining needs to search space with an exponential number of patterns. Second frequent pattern mining relies on the downward closure property to prune infrequent patterns and generate the frequent ones.

1.4 Infrequent Pattern Mining

An infrequent pattern is an item set or a rule whose support is less than the minimum support threshold. The extraction of infrequent patterns is called infrequent pattern mining. In frequent pattern mining, only frequent patterns are returned while infrequent patterns are simply discarded without further consideration. Since the most valuable information is carried by the frequent patterns and the infrequent patterns cannot adequately reflect the typical characteristics from the data because of their rare occurrence. Mining infrequent patterns is a

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 2, Special Issue 12, April 2016.

challenging endeavor because there is an enormous number of such patterns that can be derived from a given data set. More specifically, the key issues in mining infrequent patterns are: (1) how to identify interesting infrequent patterns, and (2) how to efficiently discover them in large data sets[3]. Infrequent patterns can be used in many applications like text mining - indirect associations can used to find synonyms, antonym or words that are used in different contexts. , market basket domain - indirect associations can be used to find competing items. Infrequent patterns can be used to detect errors. Infrequent itemset has important usage in n (i) mining of negative association rules from infrequent itemsets (ii) statistical disclosure risk assessment where rare patterns in anonymous census data can lead to statistical disclosure (iii) fraud detection where rare patterns in financial or tax data may suggest unusual activity associated with fraudulent behavior and (iv) bioinformatics where rare patterns in microarray data may suggest genetic disorders. To detect an infrequent and interesting association rule, we need a specific measure, other than a minimum support threshold. It is by using this measure, in place of an arbitrary support threshold, we can i) detect also infrequent and interesting association rules, ii) calculate for its strength of interestingness, and iii) search for all (i.e. both frequent and infrequent) interesting rules directly from a database using properties on this measure.

1.5 Utility Pattern Mining

Utility mining emerges as an important topic in data mining field. The main objective of Utility Mining is to identify the itemsets with highest utilities, by considering profit, quantity, cost or other user preferences. Mining High Utility itemsets from a transaction database is to find itemsets that have utility above a user-specified threshold. Itemset Utility Mining is an extension of Frequent Itemset mining, which discovers itemsets that occur frequently. In many real-life applications, high-utility itemsets consist of rare items. 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 itemset is defined as the product of its external utility and its internal utility[14]. Mining high utility itemsets 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. Frequent pattern mining techniques treat all items in the database equally by taking into consideration only the presence of an item within a transaction. However, the customer may purchase more than one of the same item, and the unit price may vary among items. High utility pattern mining approaches have been proposed to overcome this problem. The usefulness of an itemset is characterized as a utility constraint[16]. That is, an itemset is interesting to the user only if it satisfies a given utility constraint. The utility of itemset in transaction is the sum of the utilities of all the itemset in transaction in which itemet is contained.

2. Related work

Frequent item set mining is a widely used data mining technique that has been introduced in. In the traditional item set mining problem items belonging to transactional data are treated equally. To allow differentiating items based on their interest or intensity within each transaction, in the authors focus on

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 2, Special Issue 12, April 2016.

discovering more informative association rules, i.e., the weighted association rules (WAR), which include weights denoting item significance. It addresses the discovery of infrequent and weighted item sets, i.e., the infrequent weighted item sets, from transactional weighted data sets. To address this issue, the IWI-support measure is defined as a weighted frequency of occurrence of an item set in the analyzed data. Occurrence weights are derived from the weights associated with items in each transaction by applying a given cost function. Two different IWI-support measures are illustrated. The IWI-support- min measure, which relies on a minimum cost function, i.e., the occurrence of an item set in a given transaction is weighted by the weight of its least interesting item, The IWI-support-max measure, which relies on a maximum cost function, i.e., the occurrence of an item set in a given transaction is weighted by the weight of the most interesting item. Note that, when dealing with optimization problems, minimum and maximum are the most commonly used cost functions. Hence, they are deemed suitable for driving the selection of a worthwhile subset of infrequent weighted data correlations. Specifically, the following problems have been addressed: Two novel algorithms, namely Infrequent Weighted Item set Miner (IWI Miner) and Minimal Infrequent Weighted Item set Miner (MIWI Miner), which performs IWI and MIWI mining driven by IWI-support thresholds. IWI Miner and MIWI Miner are FPGrowth-like mining algorithms, whose main features may be summarized as follows: Early FP-tree node pruning driven by the maximum IWI-support constraint. The early discarding of part of the search space thanks to a novel item pruning strategy, and cost functionindependence, i.e., they work in the same way regardless of which constraint (either IWI-support-min or IWI-support-max) is applied, early stopping of the recursive FP-tree search in MIWI Miner to avoid extracting non-minimal IWIs. Mining of infrequent itemsets only takes the presence and absence of items into account. Other information about items is not considered, such as the independent utility of an item and the context utility of an item in a transaction. Typically, in a supermarket database, each item has a distinct price/profit, and each item in a transaction is associated with a distinct count which means the quantity of the item one bought.

3. Proposed work

In the infrequent weighted itemset mining weights of items, such as unit profits of items in transaction databases, are considered. With this even if some items appear infrequently they might still be found if they have high weights. However, in this structure, the quantities of items are not considered yet. Therefore it cannot satisfy the requirements of users who are paying attention in discovering the itemsets with high sales profits. I propose utility mining which refers to finding the itemsets with high profits. The meaning of itemset utility is interestingness, importance, or profitability of an item to users. Our proposed FP-tree-based utility mining utilizes the pattern growth to avoid the costly generation of a large number of candidate sets in which dramatically reduces the search space. In this paper, we have designed an efficient tree structure for mining the high utility itemsets efficiently. Here, we have proposed a novel utility FP- tree, an extended tree structure for storing essential information about utility frequent patterns. In addition to, we have utilized the mining technique used in the standard FP-growth algorithm for mining the complete set of utility patterns. The efficiency of the high utility pattern mining is realized by considering the two important thoughts. One is, a large database is compressed into a compact data structure as well as the FP-tree avoids repeated database scans

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 2, Special Issue 12, April 2016.

and the other one is our proposed FP-tree-based utility mining utilizes the pattern growth method to avoid the costly generation of a large number of candidate sets in which it dramatically reduces the search space. The experimentation is carried out on different datasets in order to find the efficiency of the proposed approach in mining of high utility itemsets when compared with the standard FP-Growth algorithm.

4. Background

The infrequent itemset are mined by considering the support and weight parameter. If the support is less than or equal to a predefined maximum support threshold then that item is considered as infrequent item. For calculating utility consider the tables in table 4.1 and 4.2. There are seven items in the utility table and seven transactions in the transaction table in the database. To calculate support, an algorithm only makes use of the information of the first two columns in the transaction table, the information of both the utility table and the other columns in the transaction table are discarded. However, an itemset with high support may have low utility, or vice versa[16]. For example, the support and utility of itemset {bc} appearing in T1, T2, and T6 are 3 and 18 respectively and those of itemset {de} appearing in T2 and T5 are 2 and 22. In some applications, such as market analysis, one may be more interested in the utility rather than support of itemsets. Traditional frequent itemset mining algorithms cannot evaluate the utility information about itemsets.

TABLE 4.1 Utility Table

Item	a	b	c	d	e	f	g
Utility	1	2	1	5	4	3	1

TABLE 4.2 Transaction Table

Tid	Transaction	Count
T1	{b,c,d,g}	{1,2,1,1}
T2	{a,b,c,d,e}	{4,1,3,1,1}
T3	{a,c,d}	{4,2,1}
T4	{c,e,f}	{2,1,1}
T5	{a,b,d,e}	{5,2,1,2}
T6	{a,b,c,f}	{3,4,1,2}
T7	{d,g}	{1,5}

TABLE 4.3 Transaction Weighted Utility

Item	{a}	{b}	{C}	{d}	{e}	{f}	{g}
TWU	69	68	66	71	49	27	10

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 2, Special Issue 12, April 2016.

Table 4.3 shows the transaction-weighted utilities of all 1- itemsets. For example, itemset {f} is contained in T4 and T6, and thus twu({f}) = tu(T4) + tu(T6) = 9 + 18 = 27.

5. Methodology

There are four main methods used for mining high utility itemsets from transactional databases that are given as follows:

4.1. Data Structure

A compact tree structure, Utility Pattern (UP) -Tree, is used for facilitate the mining performance and avoid scanning original database repeatedly[5]. It will also maintain the transactions information's and high utility itemsets.

4.2. UP-Growth Mining Method

After construction of global UP tree, mining UP-Tree by FP-Growth for generating PHUIs will generate so many candidates in order to avoid that UP-Growth method is used with two strategies: One is discarding unpromising items during constructing a local UP-Tree. Another is discarding local node utilities[13].

4.3. Efficiently Identify High Utility Itemsets

After finding all PHUIs, the third step is to identify high utility itemsets and their utilities from the set of PHUIs by scanning original database once. However, in previous studies, two problems in this phase occur: 1) number of HTWUIs is too large; and (2) scanning original database is very time consuming. In our framework, overestimated utilities of PHUIs are smaller than or equal to TWUs of HTWUIs since they are reduced by the proposed strategies. Thus, the number of PHUIs is much smaller than that of HTWUIs. Therefore, in phase II, our method is much efficient than the previous methods. Moreover, although our methods generate fewer candidates

6. Implementation Details

FP-tree structure for the first time proposed to store information on frequent items, thus transforming the issue of mining frequent itemsets into that of mining FP-trees. In an FP-tree, every node is composed of three domains: an item name, designated as item_name, a node count, designated as count, and a link, designated as node_link; besides, in order to facilitate the traversing of the FP-trees, an item header table is created to make every time point to its presence in the tree through a node link, and the header table is made up of two domains: an item name, designated as item_name, and a node link head, designated as head, and the head points to the first node in the FP-tree with the same name with it[15]. A tree structure in which all items are arranged in descending order of their frequency or support count. After constructing the tree, the rare items can be mined using FP-growth.

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 2, Special Issue 12, April 2016.

6.1 Initial utility tree

In our algorithm, each itemset holds a utility tree. Initial utility-tree storing the utility information about a mined database can be constructed by two scans of the database. Firstly, the transaction-weighted utilities of all items are accumulated by a database scan. If the transaction-weighted utility of an item is less than a given minutil, the item is no longer considered. For the items whose transaction-weighted utilities exceed the minutil, they are sorted in transaction-weighted-utility-ascending order. For the database in table 4.1 and table 4.2, suppose the minutil is 30, and then the algorithm no longer takes items f and g into consideration after the first database scan. The remaining items are sorted: e<c<b<a<d. When scanning the database again, the algorithm revises each transaction for constructing initial utility-tree. Each element in the utility-list of itemset X contains three fields: tid, iutil, and rutil.

• Field tid indicates a transaction T containing X.

• Field iutil is the utility of X in T, i.e., u(X, T).

• Field rutil is the remaining utility of X in T, i.e., ru(X, T).

The sums of the iutils and rutils in the utility-tree of an itemset can be computed by scanning the utility-tree. To avoid utility-tree scan, in the process of constructing a utility tree, accumulate the iutils and rutils in the utility-tree. In addition, there is also no need to bind each itemset to its utility-tree. The itemsets represented by all child nodes of a node in a set-enumeration tree have the same prefix itemset. Therefore, for a 1-extension, its extended item can be separated from its prefix itemset.

6.2 RUI-Miner algorithm

For each utility-tree X in ULs (the second parameter), if the sum of all the iutils in X exceeds minutil, and then the extension associated with X is high utility and outputted. Only when the sum of all the iutils and rutils in X exceeds minutil should it be processed further. When the initial utility-tree are constructed from a database, they are sorted and processed in transaction-weightedutility-ascending order. Therefore, all the utility-tree in ULs are ordered as the initial utility-tree are. To explore the search space, the algorithm intersects X and each utility-tree Y after X in ULs. Suppose X is the utility-tree of itemset Px and Y that of itemset Py, and then construct(P.UL, X, Y) is to construct the utility-tree of itemset Pxy. Finally, the set of utility-tree of all the 1-extensions of itemset Px is recursively processed. Given a database and a minutil, after the initial utility-tree IULs are constructed, RUIMiner(Ø, IULs, minutil) can mine all rare utility itemsets.

7. Conclusion

This paper faces the issue of discovering rare itemsets by using weights for differentiating between relevant items and not within each transaction.In this paper, a novel data structure has been proposed, utility-tree, and developed an efficient algorithm, RUIMiner, for Rare utility itemset mining. Utility-tree provide not only utility information about itemsets but also important pruning information for RUI-Miner. Previous algorithms have to process a very large number of candidate itemsets during their mining processes. However, most candidate itemsets are not high utility and are discarded finally. RUI-Miner can mine high utility itemsets without candidate generation, which avoids the costly generation and utility computation of candidates.

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 2, Special Issue 12, April 2016.

8. References

[1] R. Agrawal, T. Imielinski, and Swami, "Mining Association Rules between Sets of Items in Large Databases," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '93), pp. 207-216, 1993.

[2] W. Wang, J. Yang, and P.S. Yu, "Efficient Mining of Weighted Association Rules (WAR)," Proc. Sixth ACM SIGKDD Int'l Conf. Knowledge Discovery and data Mining (KDD '00), pp. 270-274, 2000.

[3] D.J. Haglin and A.M. Manning, "On Minimal Infrequent Itemset Mining," Proc. Int'l Conf. Data Mining (DMIN '07), pp. 141-147, 2007.

[4] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 1-12, 2000.

[5] A. Gupta, A. Mittal, and A. Bhattacharya, "Minimally Infrequent Itemset Mining Using Pattern-Growth Paradigm and Residual Trees," Proc. Int'l Conf. Management of Data (COMAD), pp. 57-68, 2011.

[6] Frequent Itemset Mining Dataset Repository. http:// fimi.ua.ac.be/, 2012.

[7] NU-MineBench: A Data Mining Benchmark Suite. http://cucis.ece.northwestern.edu/projects/ DMS/MineBench.html, 2012.

[8] C. F. Ahmed, S. K. Tanbeer, B.-S. Jeong, and Y.-K. Lee. Efficient tree structures for high utility pattern mining in incremental databases. IEEE Transactions on Knowledge and Data Engineering, 21(12):1708–1721, 2009.

[9] Y.-C. Li, J.-S. Yeh, and C.-C. Chang. Isolated items discarding strategy for discovering high utility itemsets. Data & Knowledge Engineering, 64(1):198–217, 2008.

[10] G. Liu, H. Lu, J. X. Yu, W. Wang, and X. Xiao. Afopt: An efficient implementation of pattern growth approach. In Proc. IEEE Int'l Conf. Data Mining Workshop Frequent Itemset Mining Implementations, 2003.

[11] Y. Liu, W.-K. Liao, and A. Choudhary. A fast high utility itemsets mining algorithm. In Proc. UtilityBased Data Mining Workshop, pages 90–99, 2005.

[12] V. S. Tseng, B.-E. Shie, C.-W. Wu, and P. S. Yu. Efficient algorithms for mining high utility itemsets from transactional databases. IEEE Transactions on Knowledge and Data Engineering, 2012, doi: 10.1109/TKDE.2012.59.

[13] V. S. Tseng, C.-W. Wu, B.-E. Shie, and P. S. Yu. Upgrowth: An efficient algorithm for high utility itemset mining. In Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pages 253–262, 2010.

[14] H. Yao, H. J. Hamilton, and C. J. Butz. A foundational approach to mining itemset utilities from databases. In Proc. SIAM Int'l Conf. Data Mining, 2004.

[15] A. JalpaA Varsur , Nikul G Virpariya. Mining Rare Itemset Based On Fp-Growth Algorithm. In International Conference on Information Engineering, Management and Security 2015 [ICIEMS 2015]

[16] Mengchi Liu, Junfeng Qu," Mining High Utility Itemsets without Candidate Generation", In CIKM'12, October 29–November 2, 2012, Maui, HI, USA. Copyright 2012 ACM 978-1-4503-1156-4/12/10 ...$15.00.

