MINING FREQUENT ITEMSETS WITH DIVERSE ASSOCIATION RULE MINING: A SURVEY

Dr. P.N. Chatur*

Prof. R.V. Mante*

D.S. Asudani*

A.R. Khobragade*

Abstract:

Frequent itemset mining leads to the discovery of association and correlation among items in large transactional or relational data sets. Massive amount of data continuously being collected and stored, many industries are becoming interested in mining such patterns from their databases. The main purpose of data mining provides superior result for using knowledge base system. Researchers presented a lot of approaches and algorithms for determining association rules. This paper discusses few approaches for mining association rules. Association rule mining approach is the most efficient data mining method to find out hidden or required pattern among the large volume of data. It is responsible to find correlation relationships among various data attributes in a huge set of items in a database. Apriori algorithm is an illustration of an enhanced association rule mining algorithm, which supports to avoid the replication of same items. Weighted Association Rule Mining (WARM) makes use of the importance of each itemset and transaction. WARM requires each item to be given weight to reflect their importance to the user. This paper also discusses several variations to the Apriori algorithm for improved efficiency and scalability. An improved frequent pattern growth adapts a divide and conquers strategy that compresses the database representing frequent items into a frequent pattern tree and compressed into a set of conditional databases.

Keywords: Association Rule Mining, Apriori, Weighted Association Rule Mining, FP-Tree.

* Government College of Engineering, Amravati

I. INTRODUCTION

Data mining is the process of extracting interesting, non-trivial, implicit, previously unknown and potentially useful information or patterns from large information repositories such as relational database, data warehouses, XML repository, etc. Frequent itemset mining plays an essential role in the mining of various patterns (e.g. association rules, correlation, sequences, episodes, maximal patterns, closed patterns) and is in demand for many real-life applications. Association rule mining aims to explore large transaction databases. Classical Association Rule Mining (ARM) model assumes that all items have the same significance without taking their weight into account. It also ignores the difference between the transactions and importance of each and every itemsets [1]. Weighted Association Rule Mining (WARM) does not work on databases with only binary attributes. It makes use of the importance of each itemset and transaction. WARM requires each item to be given weight to reflect their importance to the user. The weights may correspond to special promotions on some products, or the profitability of different items [2]. A typical example of the association rules mining is the market basket analysis. Association rules research assists to find the relationship among different products (items) in transaction databases and to find out the customer buyer behaviors, such as the purchase of a commodity impact on other goods. The results can be applied to goods shelf layout, storage arrangements, and classification of user according to buying patterns. Association Analysis is the detection of hidden pattern or condition that occurs frequently together in a given data. Association Rule mining techniques finds interesting associations and correlations among data set. An association rule is a rule, which entails certain association relationships with objects or items, for example the interrelationship of the data item as whether they occur simultaneously with other data item and how often. These rules are computed from the data and, association rules are calculated with help of probability. It has a mentionable amount of practical applications, including classification, XML mining, spatial data analysis, and share market and recommendation systems. This rule measure with support to ensure every dataset treated equally in classical model. The perception of association rule mining suggests the support confidence level outline and condensed association rule mining to the discovery of frequent item sets. Rule support and confidence are two measures of interestingness. Association rules are regarded as appealing if a minimum support and a minimum confidence threshold is satisfied. Boolean association rule mining is more extensively used than other kinds of association rule mining [3].

Apriori algorithm can control the excessive growth of the number of items of the candidate item sets by using pruning techniques bases on the minimum supporting degree. Supermarkets decision-supporting system uses multidimensional data table to save and manipulate the data. Find the inner link of the goods customers have bought and then first divide commodities in to big categories and each has a uniform weighted value, by using this method, it can avoid that each product has a power value which is inconvenient to set and adjust weights. Divide commodities into big categories and then set value according the big categories, due to fewer categories, it is convenient to set and adjust weights. Then calculate the weighted supporting degree, to obtain the association rules [4].

II. RELATED WORK

Association rule mining is an important technique or mechanism in data mining. Association rule is an implication expression of the form X→Y where X is antecedent and Y is consequent. The antecedent and consequent are set of item from item domain I. Thus X∩Y=□. If an itemset contains K items then it called as K-itemset. The support of an itemset is defined as the ratio of number of transactions containing the itemset to the total number of transactions. The confidence of the association rule X→Y is the probability that Y exists given that a transaction contain X i.e. P (X / Y) = P (XUY) / P (X). In large databases, the support of X→Y is taken as the fraction of transaction that contains XUY. The confidence of X→Y is the numbers of transaction containing both X and Y divided by the number of transactions containing X [5].

Apriori Algorithm:

Apriori approach is based on the anti-monotone Apriori heuristic, if any length k pattern is not frequent in the database, its length (k + 1) super-pattern can never be frequent. The essential idea is to iteratively generate the set of candidate patterns of length (k+1) from the set of frequent-patterns of length k (for k ≥ 1), and check their corresponding occurrence frequencies in the database. The Apriori heuristic achieves good performance gained by reducing the size of candidate sets. With a large number of frequent patterns, long patterns, or quite low minimum support thresholds, an Apriori algorithm may suffer from the following two nontrivial costs, first it is costly to handle a huge number of candidate sets. For example, if there are 104 frequent 1-itemsets, the Apriori algorithm will need to generate more than 107 length-2 candidates and accumulate and test their occurrence frequencies. To discover a frequent pattern of size 100, such

as $\{a_1, \ldots, a_{100}\}$, it must generate $2^{100} - 2 \approx 10^{30}$ candidates in total. This is the inherent cost of candidate generation. Second it is tedious to repeatedly scan the database and check a large set of candidates by pattern matching, which is especially true for mining long patterns [6].

Frequent-Pattern Tree:

Frequent-pattern tree (FP-tree) is constructed, which is extended prefix-tree structure storing crucial, quantitative information about frequent patterns. To ensure that the tree structure is compact and informative, only frequent length-1 items will have nodes in the tree, and the tree nodes are arranged in such a way that more frequently occurring nodes will have better chances of node sharing than less frequently occurring ones. Stud show that such a tree is compact, and it is sometimes orders of magnitude smaller than the original database. Subsequent frequent-pattern mining will only need to work on the FP-tree instead of the whole data set.

Alternatively an FP-tree-based pattern-fragment growth mining method is developed, which starts from a frequent length-1 pattern (as an initial suffix pattern), examines only its conditional-pattern base (a "sub-database" which consists of the set of frequent items co-occurring with the suffix pattern), constructs its (conditional) FP-tree, and performs mining recursively with such a tree. The pattern growth is achieved via concatenation of the suffix pattern with the new ones generated from a conditional FP-tree. The major operations of mining are count accumulation and prefix path count adjustment, which are usually much less costly than candidate generation and pattern matching operations performed in most Apriori-like algorithms [7].

Partitioning Based:

In partitioning-based, divide and conquer method, rather than Apriori-like level-wise generation of the combinations of frequent itemsets. This dramatically reduces the size of conditional-pattern base generated at the subsequent level of search as well as the size of its corresponding conditional FP-tree. It transforms the problem of finding long frequent patterns to looking for shorter ones and then concatenating the suffix. It employs the least frequent items as suffix, which offers good selectivity and contribute to substantial reduction of search costs [8].

Weighted Association Rule Mining on Dynamic Content:

WARM requires for each item to be given weight to reflect their importance to the user. The weights may correspond to the profitability of different items. In case of dynamic content, as more data is gathered, which are frequently getting updated, the construction of the graph should be dynamic instead of static. Using Online Hits algorithm, the graph can be constructed

dynamically and the cost can be reduced by postponing updates whenever possible, by calculating Eigen values we can enforcing the mutual reinforcement relationship between the items. HITS algorithm is suitable only for static content, where no dynamic updation is possible and it fail to capture the rich information's that lie within the patterns of user access or in the structure that can be defined by user group implicitly. HITS algorithm is replaced with Online HITS algorithm which reduces the cost by postponing the updates whenever possible and makes it more suitable for dynamic environments. HITS algorithm normally used for web pages, but it can also be used for transactional datasets from which we can calculate the hub and authorities, based on which the graph is constructed. The general HITS algorithm is too costly to run on every update. When the updates are accumulated run online HITS once. This way cost is reduced [2].

III. FREQUENT PATTERN TREE : DESIGN AND CONSTRUCTION

Let $I = \{a_1, a_2, \ldots, a_m\}$ be a set of items, and a transaction database $DB = (T_1, T_2, \ldots, T_n)$, where T_i ($i \in [1 \ldots n]$) is a transaction which contains a set of items in I . The support of a pattern A, where A is a set of items, is the number of transactions containing A in database. A pattern A is frequent if A's support is no less than a predefined minimum support threshold, ξ. With the transaction database, DB, shown in Table 1, and the minimum support threshold be 3 (i.e., $\xi = 3$). A compact data structure can be designed as follows:

i. Scan transaction database DB to identify the set of frequent items (with frequency count obtained as a by-product).
ii. Set the frequent items of each transaction in some compact structure, it may avoid repeatedly scanning the original transaction database.
iii. If multiple transactions share a set of frequent items, it may be possible to merge the shared sets with the number of occurrences registered as count. It is easy to check whether two sets are identical if the frequent items in all of the transactions are listed according to a fixed order.
iv. If two transactions share a common prefix, according to some sorted order of frequent items, the shared parts can be merged using one prefix structure as long as the count is registered properly. If the frequent items are sorted in their frequency descending order.

TID	Items bought
100	f, a, c, d, g, i, m, p
200	a, b, c, f, l, m, o

300	b, f, h, j, o
400	b, c, k, s, p
500	a, f, c, e, l, p, m, n

Table 1 : Transaction Database with Frequent items

Frequent pattern tree can be constructed as follows: First, a scan of DB derives a list of frequent items, ((f:4), (c:4), (a:3), (b:3), (m:3), (p:3)), the number after : indicates the support, in which items are ordered in frequency descending order. This ordering is important since each path of a tree will follow this order. Second, the root of a tree is created and labeled with "null". The FP-tree is constructed as follows by scanning the transaction database DB the second time.

i. The scan of the first transaction leads to the construction of the first branch of the tree: ((f :1), (c:1), (a:1), (m:1), (p:1)). The frequent items in the transaction are listed according to the order in the list of frequent items.
ii. For the second transaction, since its (ordered) frequent item list (f, c, a, b, m) shares a common prefix (f, c, a) with the existing path (f, c, a, m, p), the count of each node along the prefix is incremented by 1, and one new node (b:1) is created and linked as a child of (a:2) and another new node (m:1) is created and linked as the child of (b:1).
iii. For the third transaction, since its frequent item list (f, b) shares only the node (f) with the f -prefix subtree, f 's count is incremented by 1, and a new node (b:1) is created and linked as a child of (f :3).
iv. The scan of the fourth transaction leads to the construction of the second branch of the tree, ((c:1), (b:1), (p:1)).
v. For the last transaction, since its frequent item list (f, c, a, m, p) is identical to the first one, the path is shared with the count of each node along the path incremented by 1.

To facilitate tree traversal, an item header table is built in which each item points to its first occurrence in the tree via a node-link. Nodes with the same item-name are linked in sequence via such node-links. After scanning all the transactions, the tree, together with the associated node-links is formed.

IV. CONCLUSION

It is very important to have a data mining algorithm with high efficiency because transaction database usually are very large. Association rule mining has a wide range of applicability such as market basket analysis, medical diagnosis/ research, Website navigation analysis, homeland security and so on. Frequent itemset mining plays an essential role in the mining of various

patterns and is in demand in many real life applications. Most of the existing algorithms find frequent itemsets from traditional transaction databases consisting of precise data. Finding the weights for categories of goods by considering the authorities and hubs, where authorities are number of items and relation between transactions and items referred as hubs. The correlation between transactions and items, calculating one item is repeating in how many transactions, and in one transaction how many items are present, which helps to identify weights. This paper discusses the list of existing association rule mining techniques, and the algorithms used in mining the training data set, which can discover implicit and potential useful knowledge from large preprocessed databases. This paper also discusses enhanced variations of Apriori algorithm and method to design and construct Frequent Pattern Tree which reduces the space complexity and time complexity.

REFERENCE

[1] Carson Kai-Sang Leung, Boyu Hao, "Mining of Frequent Itemsets from Streams of Uncertain Data", IEEE International Conference on Data Engineering, pp. 1663-1670, 2009.

[2] P. Velvadivu, Dr. K. Duraisamy, "An Optimized Weighted Association Rule Mining on Dynamic Content", IJCSI, International Journal of Computer Science Issues, Vol. 7, Issue 2, No 5, ISSN (Online): 1694-0784, pp. 16 – 19, March 2010.

[3] Rachna Somkunwar, "A study on Various Data Mining Approaches of Association Rules", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 9, ISSN: 2277 128X, pp. 141-144, September 2012.

[4] Padmaja V., Poongodai A., "Mining Weighted Association Rules", IJAEST, International Journal of Advanced Engineering Sciences and Technologies, Vol No. 11, Issue No. 1, pp. 153 - 156, 2011.

[5] V. Vidya, R. Nedunchezhian, "A Robust Weighted Association Rule Mining using FP - Tree", European Journal of Scientific Research, ISSN 1450-216X, Vol.66 No.4, pp. 600 – 609, 2011.

[6] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A.I. Verkamo, "Fast discovery of association rules in Advances in Knowledge Discovery and Data Mining", AAAI/MIT Press, pp. 307–328.

[7] R. Agarwal, C. Aggarwal, V.V. Prasad, "A tree projection algorithm for generation of frequent itemsets", Journal of Parallel and Distributed Computing, pp. 61:350–371, 2001.

[8] T. Imielinski, R. Agrawal, A. Swami, "Mining association rules between sets of items in large databases", In Proc. ACM-SIGMOD, Int. Conf. Management of Data (SIGMOD'93), Washington, DC, pp. 207-216, 1993.

[9] Jiawei Han, Jian Pei, Yiwen Yin, Runying Mao, "Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach", Data Mining and Knowledge Discovery, Kluwer Academic Publishers, Manufactured in The Netherlands, 8, pp. 53–87, 2004.

[10] Sotiris kotsiantis, Dimitris Kanellopoulos, "Association Rule Mining : A Recent Overview", GESTS, International Transactions on Computer Science and Engineering, Vol. 32(1), pp.71-82, 2006.

[11] Qiankun Zhao, Sourav S. Bhowmick, "Association Rule Mining: A Survey", Technical Report, CAIS, Nanyang Technological University, Singapore, No. 2003116, 2003.

[12] Ke Sun, Fengshan Bai, "Mining Weighted Association Rules without Pre-assigned Weights", IEEE Transaction on Knowledge and Data Engineering, Vol 20, No.4, 2008.

[13] Parvinder S. Sandhu, Dalvinder S. Dhaliwal, S. N. Panda, "Mining utility-oriented association rules: An efficient approach based on profit and quantity", International Journal of the Physical Sciences Vol. 6(2), pp. 301 - 307, 18 January, 2011.

[14] Jianyong Wang, Jiawei Han, Ying Lu, Petre Tzvetkov, "TFP: An Efficient Algorithm for Mining Top-K Frequent Closed Itemsets", IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 5, May 2005.

[15] Filip Karel, "Quantitative Association Rules Mining", Doctoral Thesis, Faculty of Electrical Engineering, Department of Cybernetics, Czech Technical University in Prague, April 2008.

[16] Bin Zeng, Xiao-Li Jiang, Wei Zhao, Chao Luo, "The Improvement of Weighted Association Rules Arithmetic Based on FP-Tree", 3rd International Conference on Advanced Computer Theory & Engineering (ICACTE), pp. V4-549, V4-552, IEEE - 978-1-4244-6542-2, 2010.

[17] B. Chandra, Shalini Bhaskar, "Patterned Growth algorithm using Hub - Averaging without pre - assigned weights", IEEE, ISSN - 978-1-4577-0653-0/11, pp. 3518-3523, 2011.

[18] Wei Xie, Jing Wu, "Mining Positive and Negative Weighted Association Rules in Medical Records without User-Specified Weights Based on HITS Model", 3rd International Conference on Biomedical Engineering and Informatics (BMEI), pp. 2325-2329, IEEE, ISSN - 978-1-4244-6498-2/10, 2010.