Mining Weighted Frequent Patterns using 'Weighted_FPGrowth'- A modified FP-Growth

Deepak Sinwar
Research Scholar (CS),
deepak.sinwar@gmail.com
Dr. V. S. Dhaka
Professor
Jaipur Nationl University, Jagatpura, Jaipur, Rajasthan (INDIA)
vijaypal.dhaka@gmail.com

----------ABSTRACT----------

Mining Frequent Patterns is one of the primary step in Association Rule Mining (ARM). ARM always aims to produce relationships between different attributes of a database. Sometimes we may require including the weights (or significance) of different attributes in the ARM process; but such type of mining cannot be handled using traditional ARM approaches. To facilitate the concept of weighted attributes this paper has implemented the concept of Weighted Frequent Patterns by modifying the well known FP-growth algorithm. The modified algorithm is named as 'Weighted_FPGrowth'. We know that some patterns are not frequent at all, but they may be significant enough in some cases. Theoretical analysis and **experimental work have shown that the modified approach is able to detect the relative importance of attributes in terms of their weights. The attributes which were less frequent are now frequent due to their weights.**

1. INTRODUCTION

Association Rule Mining (ARM) is a well established Data Mining technique for finding hidden patterns called Association Rules between different attributes/items of a transaction database (i.e. Market-Basket database). A *frequent itemset* typically refers to a set of items that frequently appear together in a transactional data set, such as milk and bread [6]. Many algorithms have been proposed to obtain useful and invaluable information from huge databases in research literature. In general data mining tasks can be classified into two categories: descriptive and predictive [4]. Descriptive mining tasks (Association rule discovery, Clustering etc.) characterize the general properties of the data in the data base. Predictive mining tasks (Classification and Regression etc.) perform inference on the current data in order to make predictions. The association rule $A \Rightarrow B$ holds in transaction set D with support *s*, where *s* is the percentage of transactions in D that contain $A \cup B$ (i.e., the union of sets A and B, or say, both A and B) and with confidence *c,* where *c* is the percentage of transactions in D containing A that also contains B. This is taken to be the conditional probability, P(B|A). Rules that satisfy both a minimum support threshold (*min*_sup) and a minimum confidence threshold (*min_conf*) are called strong. Association rule mining generally a two step approach: first, find all frequent patterns; second, generate association rules through frequent patterns. In most cases the algorithms which are used to generate association rules are either based on Aprori [1] algorithm or may follow FP Growth approach [5]. However, these traditional association rule mining (ARM) models assume that items have the same significance without taking account of their weight/attributes within a transaction or within the whole item space. But this is not always the case as some items may be more important than others for one or other reasons.

For example, [wine → salmon, 1%, 80%] may be more important than [bread → milk, 3%, 80%] even though the former holds a lower support. This is because items in the first rule bring more profit per unit sale, but the standard ARM simply ignores this difference [9]. In order to tackle this problem we have modified the well known association rule mining approach i.e., FP-growth in order to include the concept of weights of the attributes. The new algorithm works by considering a weight value for every item (attribute) so that the attribute (s) can reflect their significance.

The rest of this paper is organized as follows. In section 2, some work related to Association Rule Mining has been discussed. Section 3 proposes the concept of mining weighted association rules using Weighted_FPGwoth. In section 4, we have analysed the experimental results. The conclusion and future works are made in section 5.

2 RELATED WORK

Plenty of work has been done for mining simple as well as weighted association rule mining. The principle of weighted association rule mining was first given by Cai *et al.*, [2]. Their aim is to produce such association rule mining algorithm which does not take all items of a basket database uniformly. They generalize this case where items are given weights to reflect their importance to user. Most of the work in association rule mining focused to determine strong association rules which satisfy the minimum support (*min_sup*) and minimum confidence (*min_conf*) thresholds. But the motive of association rule mining is to

maximize/increase the profit of organizations and also to make business decisions.

An improved method of weighted association rule mining has been proposed by Feng Tao *et al.* [9] called WARM (Weighted Association Rule Mining), where each item is allowed to have a weight as per its importance. The goal of using weighted support in WARM is to make use of the weight in the mining process and prioritize the selection of target itemsets according to their significance in the dataset, rather than their frequency alone. Another weighted association rule mining algorithm called Weighted Frequent Itemset Mining (WFIM) has been proposed in [11], using global and conditional FP-trees, which generates concise and important weighted frequent itemsets in large databases, particularly dense databases with low minimum support, by adjusting a minimum weight and a weight range.

Lots of Apriori-like algorithms [2, 8, 10 and 12] have proved to achieve good performance. However, it is costly to handle a large number of candidate sets and must scan the database repeatedly. So, to perform fast weighted association rule mining using well establish Apriori algorithm, Zhou *et al.*, [12] uses Lucene index which is a high-performance, full-featured text search engine library written entirely in Java. The Analytical Hierarchy process is the new concept introduced by Jian and Ming [6] which consists of weight setting, judgement matrix building and calculating the weight of items. Their new algorithm WARM enables the weight of alarms in communication networks to be more flexible, natural and understandable. To maintain the popularity of Apriori algorithm, Swargam and Palakal [8] produces three variants of Apriori algorithm called adaptive Apriori to discover item sets with low and high frequency. Wang et al. [10] has introduced the concept of ALocating Pattern (*ALP*) which is a special form of weighted association rule (WAR) mining, where each item is associated with a weighting score between o and 1, and the sum of all rule item scores is 1 [8]. This special case of weighted association rule mining is introduced as the "one-sum" WAR.

3. WEIGHTED_FPGROWTH

As discussed in section-1 that Weighted_FPGrowth is a modified version of FP-grwoth algorithms [5]. The modified algorithm considers a weight as a parameter to reveal the importance of an attribute. The motivation behind this modification is to extract those associations which are infrequent in nature, but sometimes more significant from others. The significance is a general parameter whose value varies from person to person. Some attributes are important for some set of persons but some are not. In this modified version we have used random weight values (from 0.0 to 1.0) using random number generation method of Java language.

The working principle of Weighted_FP Growth implements a concept of building Frequent-Pattern tree by pruning those attributes / items from the database whose weights are below the minimum weight threshold and then inserting the remaining items to the tree and also having more support count than the minimum support threshold. We have simulated the modified version in Java Language using some modifications in the already developed version of FP-growth in WEKA [3]. Let us see the pseudo code of the modified algorithm:

3.1 Algorithm: Weighted_FPGrowth)

Input: *A* transaction database *D* in the form of binary values

Output: A list of weighted association rules

Method:

1. *S*can the database once to find out the support count (*m*) of each item *a* of the database.
2. Build the frequent pattern tree (after assigning a random weight to all attributes) by inserting only those items of a transaction database which satisfies the following conditions:

*Condition 1: Weight (*Item *i) > min_weight*
*Condition 2:Support_count (*Item *i) > min_suppport*

3. The FP-Tree is mined by calling *FP-growth*(*FP-tree, null*) [4].
4. Generate association rules from weighted frequent itemsets generated in step 3.
5. End.

Figure-1 shows the actual coding of building FP Tree.

```
protected FPTreeRoot buildFPTree
(ArrayList<BinaryItem> singletons,
    Instances data, int minSupport)
{
FPTreeRoot tree = new FPTreeRoot();
for(int a=0; a<data.numAttributes();a++)
{data.attribute(a).setWeight(Math.random());
for (int i = 0; i < data.numInstances(); i++) {
Instance current = data.instance(i);
ArrayList<BinaryItem> transaction = new
ArrayList<BinaryItem>();
if (current instanceof SparseInstance){
for (int j = 0; j < current.numValues(); j++) {
int attIndex = current.index(j);
if (singletons.get(attIndex).getFrequency() >=
minSupport && data.attribute(a).weight()>=0.4) {
transaction.add(singletons.get(attIndex)); } }
Collections.sort(transaction);
tree.addItemSet(transaction, 1); }
else {
for (int j = 0; j < data.numAttributes(); j++) {
if (!current.isMissing(j)) {
if (current.attribute(j).numValues() == 1 ||
current.value(j) == m_positiveIndex - 1) {
if (singletons.get(j).getFrequency() >= minSupport &&
data.attribute(a).weight()>=0.4) {
transaction.add(singletons.get(j));} } } }
Collections.sort(transaction);
tree.addItemSet(transaction, 1); } } }
return tree;
}
```

Figure 1: Actual coding of building FPTree in Java Language

We have implemented the code as shown in figure-1 in Java language and then the algorithm has been added to the GUI of WEKA so that we can compare the results. The GUI after the addition of Weighted_FPGrowth is shown in figure-3.

4. EXPERIMENTAL WORK AND ANALYSIS

We have performed a set of experiments to validate our modified algorithm i.e. Weighted_FPGrowth. One real world dataset (dataset 01: Pima Indians Diabetes Database) obtained from UCI machine learning repository and one synthetic dataset (dataset 02: LED24 generated from WEKA) has been used in the experimental work. All experiments were performed on Intel(R) Core (TM) i3 CPU M370 2.4 GHz with 2 GB of main memory running on windows 7(32 bit).

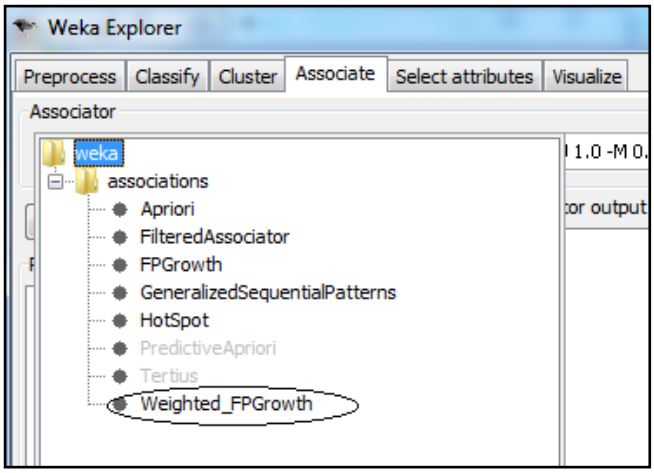

Figure 3: The GUI of WEKA after addition of new algorithm (i.e., 'Weighted_FPGrowth')

As we know that before mining the frequent patterns the database must be in binary form. In the absence of binary datasets we may use filters to convert a non-binary data to the binarized dataset. We have used unsupervised attribute filter on Pima Indians Diabetes Database to convert the numeric values into binary values so that we are able to perform our experiments. Firstly we run the already developed FPGrowth algorithm on both datasets and then Weighted_FPGrowth. A comparative study has been made to validate the modified algorithm using the 6 experiments as:

Experiment1: Running FPGrowth on first dataset
Experiment2: Running Weighted_FPGrowth on first dataset
Experiment3: Running Weighted_FPGrowth on first dataset again (to check the randomness of the weights, because we have used the concept of assigning random weights)
Experiment4: Running FPGrowth on second dataset
Experiment5: Running Weighted_FPGrowth on second dataset
Experiment6: Running Weighted_FPGrowth on second dataset again (to check the randomness of the weights as done in Experiment 3).

The default configurations as supplied in WEKA have been used in all the experiments except the minimum weight threshold, which is a newly added feature in Weighted_FPGrowth algorithm. A minimum threshold value for the weight parameter is set to 0.4 in the modified algorithm as shown in Figure-1.

Analysis of Experiments:

As shown in figure-2 that in experiment-1, the FPGrowth algorithm has generated 60 association rules whereas Weighted_FPGrowth has generated 180 rules in both experiments 2 and 3. But in 4th experiment FPGrowth algorithm has generated 20 association rules whereas Weighted_FPGrowth has generated 329 rules in experiment-5 and 263 rules in experiment-6. We have seen a rapid growth in the rules generated by the modified algorithm due to the weight factor. As discussed in section-3 that we have supplied weight values to different attributes of both the datasets using random number generation method of Java to check the significance of weights. But in actual environment the actual weights are different from random weights. If we know the actual weights then we can also assign the actual weights in the modified algorithm.

After analysing the results of various experiments we can say that the modified algorithm has generated more association rules due to the weights supplied only. If we have the actual weights then we can also see their significance using the modified algorithm.

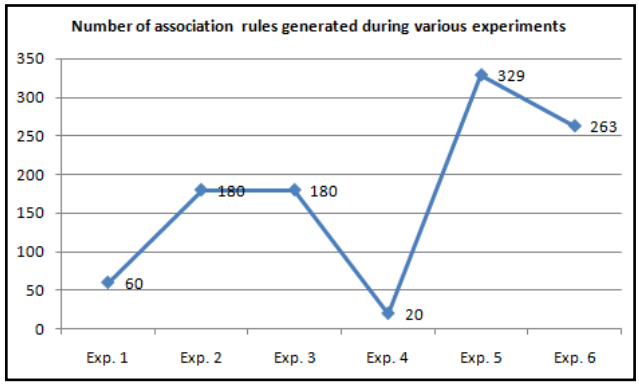

Figure: 2 Number of association rules generated by both algorithms on two datasets

5. CONCLUSION AND FUTURE WORK

This paper has proposed a modified FPGrowth algorithm that takes into account the weights of the attributes of the datasets. One real and one synthetic dataset were used in six different experiments to validate the comparative study. Experimental results and their analysis have shown that the modified algorithm is scalable to the real/ actual weights. The performance of Weighted_FPGrowth has been compared with FP-growth algorithm in terms of number of rules generated after supplying random weights in Weighted_FPGrowth. We found that Weighted_FPGrowth algorithm outperforms the traditional FPGrowth in terms of number of rules generated; but when we talk about the conciseness the performance of FPGrowth was better. This work may be extended by considering actual weights from real environment on some large and real world databases.

REFERENCES

[1] R. Agrawal, R. Srikant, "Fast Algorithms for Mining Association Rules," in *Proc. 20th Intl. Conf. on Very Large Data Bases (VLDB'94)*, Santiago de Chile, Chile, pp. 487-499, 1994.

[2] C. H. Cai, A. W. C. Fu, C. H. Cheng and W. W. Kwong, "Mining Association Rules with Weighted Items," in *Proc. of Intl. Database Engineering and Applications Symposium*, Cardiff, Wales, pp. 68-77, 1998.

[3] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and H. Ian, "The WEKA Data Mining Software: An Update", SIGKDD Explorations, Volume 11, Issue 1, 2009.

[4] J. Han, M. Kamber, *Data Mining: Concepts and Techniques,* Morgan Kaufmann Publisher, San Francisco, USA, 2001.

[5] J. Han , J. Pei and Y. Yin ," Mining frequent patterns without candidate generation", in proc. Intl. conference on Management of Data, 2000, pp. 1-12

[6] W. Jian, L. X. Ming, "An Effective Mining Algorithm for Weighted Association Rules in Communication Networks," *Journal of Computers,* Vol. 3, pp. 20-27, 2008.

[7] T. Li, X. Li and H. Xiao, "An Effective Algorithm for Mining Weighted Association Rules in Telecommunication Networks," in *Proc. of Intl. Conf.* on Computational Intelligence and Security *Workshops*, Harbin, pp. 425-428, 2007.

[8] R. J. Swargam, M. J. Palakal, " The Role of Least Frequent Item sets in Association Discovery," in *Proc. of 2nd Intl. Conf. on Digital Information Management, ICMD'07,* Lyon, pp. 217-223, 2007.

[9] F. Tao, F. Murtagh and M. Farid, "Weighted Association Rule Mining using weighted support and signigificance framework," in *Proc. 9th ACM-SIGKDD'03,* Washington, D.C., pp. 661-666, 2003.

[10] Y. J. Wang, X. Zheng, F. Coenen and C. Y. Li, "Mining Allocating Patterns in One-sum Weighted Items," in *Proc. of IEEE Intl. Conf. on Data Mining Workshops*, Pisa, pp. 592-598, 2008.

[11] U. Yun U., Leggett J. J., "WFIM: Weighted Frequent Itemset Mining with a weight range and a minimum weight," in *Proc. of Intl. conf. on Data Mining, SIAM,* pp. 636-640, 2005.

[12] Zhou N., J. Wu, S. Zhang, H. Chen and X. Zhang, "Mining Weighted Association Rules with Lucene Index," in *Proc. of Intl. Conf. on Wireless Communications, networking and Mobile Computing*, Shanghai, pp. 3697-3700, 2007.