Profit Maximizing Approach in Data Mining using Modified Weighted Association Rule Mining Algorithm

Ramesh Yadav[1], Snehal[2], Aafaq Wali[3]
[1,2]Department of IT & Software Engineering, BITS, Bhiwani (Haryana), India
[3]Department of Computer Science & Engineering, BITS, Bhiwani (Haryana) India
[1]ramesh_sqt1@rediffmail.com, snehal123_1986@rediffmail.com[2], aafaqw@gmail.com[3]

***Abstract* – This research paper proposes a Modified Frequent Pattern-Growth algorithm, a new type of Weighted Association Rule Mining algorithm, which takes into account the weights of items as well as their support counts while discovering weighted association rules. This is generalization of the weighted association rule mining. The main issues of weighted association rules mining investigated in this work are – how good are weights as compared to support counts, and the problem of reducing the amount of association rules for achieving conciseness. The experimental results show a series of six experiments in which Weighted Association Rule Mining (WARM) method outperforms traditional Associated Rule Mining (ARM) method in all confidence levels.**

Keywords – Data Mining, ARM, WARM, FP-Growth algorithm, Weighted FP-Growth algorithm.

I. INTRODUCTION

Association Rule Mining (ARM) is a well established data mining technique used for the extraction of hidden and interesting patterns called association rules. ARM works well by considering uniform importance of items. But in real life scenario different items play different role in our daily life. So nowadays, it is vital requirement to introduce such association rule mining algorithm which understands the importance of items while mining day to day market-basket transaction records. This type of ARM is called Weighted Association Rule Mining (WARM). Plenty of work has been done for mining simple as well as weighted association rule mining, but the most of work of traditional ARM focuses on the well established ARM method i.e., Apriori. On the hand different researchers adopt different scenarios for Weighted Association Rule mining.

A. Traditional ARM

As we know that association rule mining is the building block for weighted association rule mining, we briefly review some of the published research related to ARM. Generally ARM was primarily proposed for market basket analysis to understand consumer purchasing patterns in retailing industry. Because analysis of past data is a commonly used approach in order to improve quality of such decisions. Until recently, however, only global data about the cumulative sales during some time period (a day, a week, a month, etc.) was available on the computer. Progress in bar-code technology has made it possible to store the so called basket data that stores items purchased on a per-transaction basis. Basket data type transactions do not necessarily consist of items bought together at the same point of time. It may consist of items bought by a customer over a period of time. Examples include monthly purchases by members of a book club or a music club. Several organizations have collected massive amounts of such data. These data sets are usually stored on tertiary storage and are very slowly migrating to database systems. One of the main reasons for the limited success of database systems in this area is that current database systems do not provide necessary functionality for a user interested in taking advantage of this information. So, for efficiently mining a large collection of basket data type transactions produces and efficient algorithm for this purpose. After this efficient methodology they have proposed a solution to the problem of "generating all association rules that have support and confidence greater than the user specified minimum support and minimum confidence thresholds respectively. Most of the research in this area during that decade moves around the so called Apriori algorithm developed by Agrawal and Srikant in 1994 at 'IBM Almaden research centre'. But due to some limitations of Apriori algorithm like needs several iterations over the data, use of uniform minimum support threshold etc, there was a need to generate such efficient ARM method which saves time by reducing the number of passes over the database as well as without the help of candidate generation. So in this way many researchers have proposed different variations of the Apriori method. A combination of both Apriori algorithm and FP-growth has been proposed in many research publications. One of them is a new method Apriori-Growth has been presented in very effective manner by Wu et al., 2008 in [WZL+08]. Using computational results shown by them, we can easily see the performance of Apriori-Growth is much faster than Apriori algorithm, and it is almost as fast as FP-Growth, but it needs smaller memory. Because ordinary association rule mining algorithms considers same function of each attribute, they can not effectively discover the association rules which can reveal cause and effects relationships between attributes. Secondly another combination of Apriori and FP-Growth is merged in APFT algorithm [LZW09]. The advantage of APFT (APriori FP-Tree) is that it does not need to generate conditional pattern bases and sub-conditional pattern tree recursively. It means that the compressed FP-tree is partitioned off a set of conditional subtree, each of the conditional subtree associated with a frequent item. If there are n 1-frequent items I_i (i=1,2,…,n), then the FP-tree can be divided into n conditional subtree FPT_i (i=1,2,…,n), and FPT_i is the conditional subtree associating with frequent item I_i. Then use the Apriori algorithm to mine each conditional subtree, and gain all the frequent itemsets with the first prefixI_i. The AFPT algorithm includes two steps, the first step is to construct the FP\-tree as FP-growth does, the second step is to use of the Apriori algorithm to mine the FP-tree.

B. Work related to WARM

There have been tremendous works in the area of weighted association rule mining, different approaches are adopted by different researches. But we can group these approaches whose basic structures are same, for example Apriori like structure, pattern growth structure, fuzzy weighted approaches, positive and negative weighted rule approaches, direct indirect rule approaches, and finally others with a combination of either Apriori and pattern growth approaches or either a totally different perspectives.

C. Apriori like Structure

The principle of Weighted Association Rule Mining was first given by Cai et al., in [CFC+98]. Their aim was 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. The weights may correspond to special promotions on some products, or the profitability of different items. They propose two new algorithms to mine weighted binary association rules namely MINWAL(O) and MINWAL(W) which uses a metric called the k-support bound. Plenty of work has been done in the area of weighted association rule mining, in which most work is based on well established Apriori algorithm which is less efficient in terms of time and candidate generation. Some work is also based on pattern growth structure but they have not apply high preference to weights only, a combined preference of both support counts as well as weights has been treated. So that, our first aim is to generate fast as well as efficient weighted association rule mining algorithm based on pattern growth structure towards profit maximization by assigning higher preference to weights of items, secondly the conciseness of results.

II. METHODOLOGY USED

A. Modified Weighted FP-Growth

A new type of Weighted Association Rule Mining algorithm has been designed to produce weighted association rules by making some minimal changes to the Frequent Pattern-Growth algorithm. Our main objective is to extract such relationships between those items which are infrequent in nature but are more significant from others. The significance is based on the percentage of profit earned per unit of sale of that item. With the help of weight vector, it is possible for customers/users to supply weights according to his/him choice, thus making Weighted Frequent-Pattern (WFP) more effective than other WARM algorithms. WFP works on the principle of building Frequent-Pattern tree by firstly pruning/eliminating those items from the transaction database D whose weight is below the minimum weight threshold and secondly and also having more support count than the minimum support threshold, as shown in below:

Algorithm: WFP (Weighted FP-Growth)

Input: A transaction dataset D, min_wt (minimum weight of an item),
min_sup (minimum support), min_conf (the minimum confidence threshold).

Output: A list of weighted association rules.

Method: Firstly, scan the data set once to find out the support count(m) of each item a of transaction dataset.
Build the frequent pattern tree by inserting only those items of a transaction which satisfies both conditions:

(a) If weight of that item is greater than minimum weight threshold i.e.,

Weight(Item i) > min_wt

(b) Support count of that item is greater than minimum support threshold i.e.,

Supp_count(Item i) > min_sup

The FP-Tree is mined by calling FP-Growth (FP-tree, null)
Generate association rules form weighted frequent itemsets generated in Step 3.

End.

The Weighted FP-Growth algorithm

B. RWC (Rule Weight Computation) Procedure

The RWC procedure is designed to calculate the total weight of a rule generated by WFP and FP-Growth algorithms. Further these weights will be used to calculate average weight of top rules, which are obtained at specified minimum confidence threshold value. The procedure is defined below:
Procedure: RWC (Rule Weight Computation)

Input: An association rule

Output: Total weight of the rule.

Method: 1. If A be an association rule comprises of premises and consequences as:

$A = L_p + L_c$ where,

L_p = A List of n premise items {1,2,…,n}

Lc: A List of m consequence {1,2,…,n}

and then

Calculate weight of rule A using the following formula:

$Wt(A) = wt(L_p) + wt(L_c)$

[where, $wt(L_p) = (wt(a_1) + wt(a_2) + \ldots + wt(a_n)) * s_1$

$wt(L_p) = (wt(a_1) + wt(a_2) + \ldots + wt(a_m)) * s_2$

and s_1 & s_2 are the support of premise and consequence respectively.

I. Return wt(A).

Procedure to compute weights of association rules

We will compute average weight of all A's generated by both WFP and FP-Growth algorithms on a specified confidence level in descending form from 0.9 to 0.5 for taking the difference between them.

C. Generation of synthetic datasets

In our experiments we have used two experimental synthetic datasets which are generated by BayesNet and LED24 data generator programs of WEKA - 3.7 (Waikato Environment for Knowledge Acquisition) Data Mining software. The notations first data set and second data set are used for BayesNet and LED24 respectively throughout this Research paper. The first dataset comprises of 15 attributes and one hundred instances, whereas second dataset comprises of 24 attributes and same number of instances as of first dataset. The idea behind generating the first dataset is formed by generating random instances based on a Bayes network and the data produced by second dataset is data for display with 7 LEDs. We have used the default options for generation of both datasets except for the first dataset, where we have supplied value 15 to option 'numAttributes'. Let us see those different options provided along with the data generator programs which makes data generation more scalable and efficient.

1) The options with first dataset are:

Cardinality: The cardinality of the attributes, incl the class attribute.

Debug: Whether the generator is run in debug mode or not.

NumArcs: The number of arcs in the bayesian net, at most: n * (n - 1) / 2 and at least: (n -1); with n = numAttributes

NumAttributes: The number of attributes the generated data will contain (including class attribute), i.e., the number of nodes in the bayesian network

NumExamples: The number of examples to generate. RelationName: The relation name of the generated data (if empty, a generic one will be

supplied).

Seed: The seed value for the random number generator.

2) The options with second dataset are:

Debug: Whether the generator is run in debug mode or not.

NoisePercent: The noise percent: 0 <= perc <= 100.

NumExamples: The number of examples to generate.

RelationName: The relation name of the generated data (if empty, a generic one will be supplied).

Seed: The seed value for the random number generator.

D. Distribution of Weights

Weights to attributes are distributed in a variety of ways to produce weighted association rules. In our experiments we have not applied any unique formula to calculate and assign weights, because we assign this responsibility upon the end user or customer. The customer on his behalf provides a list of weights corresponding to attributes which further will be assigned to each attribute present in the market. But for checking the strength of our algorithm we have assigned weights in three different ways:

- According to support counts (flat weights)
- All flat weighted except for one attribute
- On the basis of real item prices and unique profit percentage

III. RESULT

We present the effect of 5 different values of minimum confidence (min_conf) on average weights of rules generated by both algorithms i.e., WFP and FP-growth. A series of six experiments are performed to analyze their performance. Every experiment will summarize the average_weight (obtained by RWC procedure explained in section 2.2) of association rules along with an extra parameter num_rules(number of association rules) for performance evaluation. The num_rules parameter gives us the exact amount of association rules generated on a prescribed confidence level. Every time we execute our program, it generates a list of association rules and we have considered only top 10 out of them. Generally, we will see a higher difference in values of average weights of rules generated by both algorithms. Because it is mere fact that there will be less population above the average level, and level below average always comprises of equal number of population for each test. The selection criterion for top 10 rules is based on support of a rule. These 10 results are selected from a list of all rules generated on the supplied min_conf scale, for example: if we have supplied min_conf value 0.9 to the environment, then the resultant association rules will be generated for a range of min_conf between 0.8 and 0.9; if we provides value 0.8 then the rules will be extracted from level 0.7 to 0.8. For that reason, we have supplied only 5 values of min_conf ranging from 0.9 to 0.5 to the running environment, because for values lower than 0.5 we have not found any change in results.

A. Performance Evaluation

The performance evaluation is analyzed by supplying 5 different values of min_conf along with an extra parameter num_rules. The performance will be evaluated on the basis of summary of six experiments, which are discussed from section 3.2 to 3.7.

B. Experiment 1*: Flat/ Equi-Support distribution on first dataset*

Our first experiment considers the distribution of flat weights. By Flat weights, we mean that weights are symmetrical to support count values of attributes.

Table I

SUMMARY STATISTICS of FIRST EXPERIMENT COMPRISES of AVERAGE_WEIGHT and NUMBER of RULES of WFP and FP-GROWTH at 5 SPECIFIED CONFIDENCE LEVELS.

Min_conf	FP-Growth		WFP	
	Avg_wt	Num_rules	Avg_wt	Num_rules
0.9	24.78	23	44.16	1
0.8	34.58	19	42.20	12
0.7	46.87	16	50.14	11
0.6	47.32	14	47.31	12
0.5	58.06	11	58.06	11

The minimum weight (min_wt) threshold value for this experiment is set to 0.45. From Table I we can see that WFP generates association rules which are having average_weight (avg_wt) value 44.16 at confidence level 0.9, but FP-Growth produces rules which were having an average_weight value 27.78 at same level of minimum confidence. Please refer table I for other values of avg_wt and num_rules at remaining confidence levels. It is clear that WFP generates less number of rules for each confidence level then FP-Growth. We know that for obtaining higher profit we have to take effective decisions, but generally in market-basket scenario we have no such enough time to test all decisions. Thus, we require only a small amount of effective rules, which can maximize our profit as well as save the costly time. So, we can conclude from first experiment that for this type of environment, WFP outperforms the ordinary model (FP-Growth).

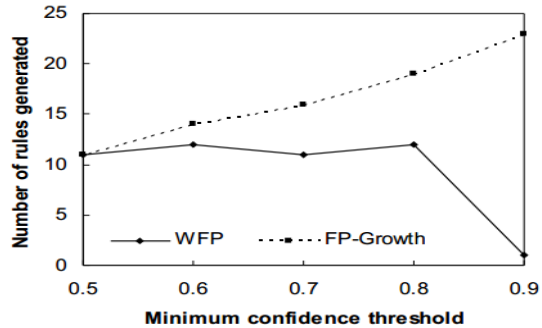

Figure 1 Number of rules generated by both algorithms at specified confidence levels for first experiment.

C. Experiment 2*: Flat/ Equi-support distribution on second dataset*

Our second experiment considers the same situation as considered in first experiment, where flat distribution of weights according to support count of attributes is applied.

Table II

SUMMARY STATISTICS of SECOND EXPERIMENT COMPRISES of AVERAGE_WEIGHT and NUMBER of RULES of WFP and FP-GROWTH at 5 SPECIFIED CONFIDENCE LEVELS.

Min_conf	FP-Growth		WFP	
	Avg_wt	Num_rules	Avg_wt	Num_rules
0.9	67.30	20	68.44	19
0.8	87.59	13	85.53	10
0.7	89.68	13	89.68	12
0.6	89.68	18	89.68	17
0.5	89.68	20	89.68	18

The minimum weight (min_wt) threshold value for this experiment is set to 0.44. From table II we can see that WFP generates association rules which are having average_weight (avg_wt) value 68.44 at confidence level 0.9, but FP-Growth produces rules which were having an average_weight value 67.3 for same level. Please refer table II for other values of avg_wt and num_rules at remaining confidence levels. Therefore, we conclude from second experiment that the both algorithms on the basis of rule weights is identical but on the basis of extra parameter i.e., num_rules WFP performs better than FP-Growth algorithm.

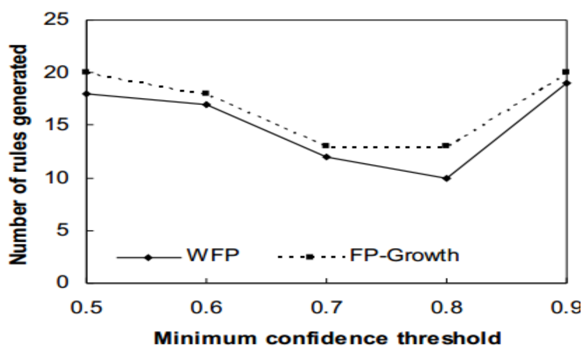

Figure 2 Number of rules generated by both algorithms at specified confidence levels for second experiment's environment.

C. Experiment 3*: All flat weighted except for one attribute on first dataset*

In this distribution we select an attribute arbitrarily which is having lower support count. A higher weight value has been assigned to it to test the performance in the sense that, we can achieve better performance by setting higher weights to more than one attributes, if we see good results by only setting higher weight to a single attribute. After setting the weights in this fashion, we found that our model produces such association rules contains that item whose weight is manually set by us. We can also view this approach as, "All Flat or Equi-weighted except for one".

Table III

SUMMARY STATISTICS of THIRD EXPERIMENT COMPRISES of AVERAGE_WEIGHT and NUMBER of RULES of WFP and FP-GROWTH at 5 SPECIFIED CONFIDENCE LEVELS.

Min_conf	FP-Growth		WFP	
	Avg_wt	Num_rules	Avg_wt	Num_rules
0.9	25.42	23	29.21	5
0.8	35.12	19	42.84	10
0.7	47.41	16	48.63	12
0.6	47.32	14	50.06	12
0.5	58.06	11	58.06	11

The minimum weight (min_wt) threshold value for this experiment is set to 0.44. From Table III we can see that WFP generates association rules which are having average_weight (avg_wt) value 29.21 at confidence level 0.9, but FP-Growth produces rules which were having an average_weight value 25.42 for same level. Please refer Table III for other values of avg_wt and num_rules at remaining confidence levels. Therefore, we conclude this experiment that the performance of WFP algorithm remains higher as compared to FP-Growth on the basis of rule weights as well as extra parameter i.e., num_rules for this type of environment. Now we can say that, we can achieve better results by setting higher weight value to more than one attributes.

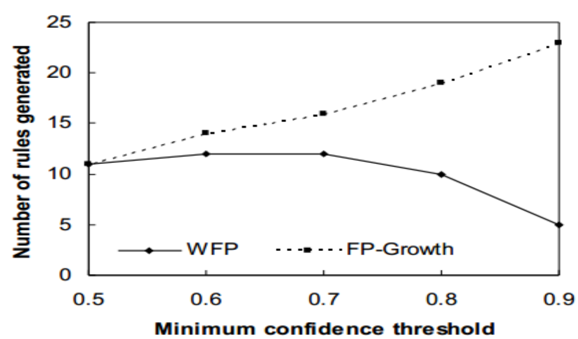

Figure 3 Number of rules generated by both algorithms at specified confidence levels for third experiment's environment.

D. Experiment 4*: All flat weighted except for one attribute on second dataset*

Our previous experiment select an attribute arbitrary which is having lower support count for assigning a higher weight to it. The same approach is applied here but the difference is of only the dataset which is used to perform the experiment. We may also view this approach as, "All Flat/Equi-weighted except for one". We apply a higher weight to a non-higher supporting attribute by considering a real life scenario, in which sometimes we see that some of the items available in market are purchased occasionally/ rarely which may cost higher then others which are purchased on regular basis.

Table IV

SUMMARY STATISTICS of FOURTH EXPERIMENT COMPRISES of AVERAGE_WEIGHT and NUMBER of RULES of WFP and FP-GROWTH at 5 SPECIFIED CONFIDENCE LEVELS.

Min_conf	FP-Growth		WFP	
	Avg_wt	Num_rules	Avg_wt	Num_rules
0.9	67.04	20	68.49	11
0.8	75.95	13	75.28	12
0.7	87.59	13	88.52	12
0.6	87.59	18	88.52	16
0.5	87.59	20	88.52	18

The minimum weight (min_wt) threshold value for this experiment is set to 0.5. From Table IV we can see that WFP generates association rules which are having average_weight (avg_wt) value 68.49 at confidence level 0.9, but FP-Growth produces rules which were having an average_weight value 67.04 for same level, which is less then WFP. Please refer table IV for other values of avg_wt and num_rules at remaining confidence levels. Therefore, we conclude this experiment that the performance of WFP algorithm remains higher as compared to FP-Growth on the basis of rule weights as well as extra parameter i.e., num_rules for this type of environment.

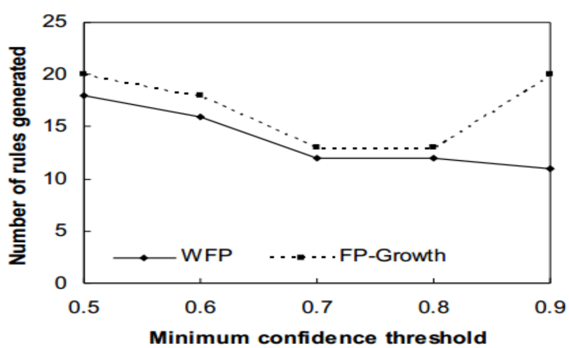

Figure 4 Number of rules generated by both algorithms at specified confidence levels for fourth experiment's environment.

E. Experiment 5*: Arbitrary distribution of real weights on first dataset*

Our next experiment considers the prices of books from amzon.com [A10]. These prices are assigned to attributes of first dataset randomly. We use this type of distribution for evaluating performance of WFP algorithm in the sense that what will be the effect on result if item prices are from real world. After setting real prices we have applied weights to the attributes using the formula (Final Weight = Support Count * Profiting Amount). We may also view this approach as, "Arbitrary distribution of real weights". The prices assigned to the attributes of first dataset are ranged with a lot of variations from $8.95 to $68.68.

TABLE V

SUMMARY STATISTICS of FIFTH EXPERIMENT COMPRISES of AVERAGE_WEIGHT and NUMBER of RULES GENERATED by WFP and FP-GROWTH at 5 SPECIFIED CONFIDENCE LEVELS.

.Min_conf	FP-Growth		WFP	
	Avg_wt	Num_rules	Avg_wt	Num_rules
0.9	40.16	23	44.07	1
0.8	42.06	19	95.83	7
0.7	66.86	16	72.81	11
0.6	104.17	14	130.14	15
0.5	104.17	11	124.66	15

The minimum weight (min_wt) threshold value for this experiment is set to 0.59. From table V we can see that WFP generates association rules which are having average_weight (avg_wt) value 44.07 at confidence level 0.9, but FP-Growth produces rules which were having an average_weight value 40.16 for same level, which is less then WFP. Please refer the same table V for other values of avg_wt and num_rules at remaining confidence levels. Therefore, we conclude this experiment that the performance of WFP algorithm also remains higher as compared to FP-Growth on the basis of rule weights as well as extra parameter i.e., num_rules for this type of environment.

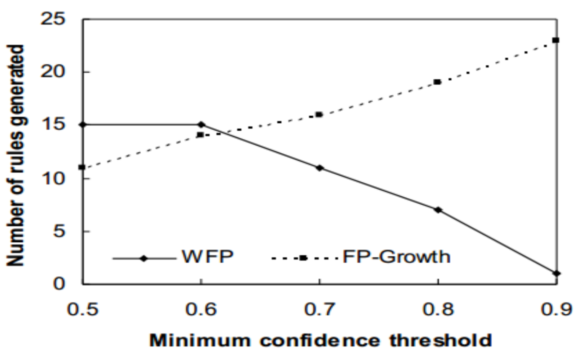

Figure 5 Number of rules generated by both algorithms at specified confidence levels for fifth experiment's environment.

F. Experiment 6*: Arbitrary distribution of real weights on second dataset*

In this experiment, we have again considered the prices of books from amzon.com [A10]. These prices are assigned to attributes of second dataset randomly. Same approach of calculating final weights to attributes is applied in the experiment 5. The benefit of assigning real weights in this order brings us a real life application of WFP. We may also view this approach as, "Arbitrary distribution of real weights".

TABLE VI

SUMMARY STATISTICS of SIXTH EXPERIMENT COMPRISES of AVERAGE_WEIGHT and NUMBER of RULES GENERATED by WFP and FP-GROWTH at 5 SPECIFIED CONFIDENCE LEVELS.

Min_conf	FP-Growth		WFP	
	Avg_wt	Num_rules	Avg_wt	Num_rules
0.9	110.46	20	127.73	13
0.8	125.54	13	191.47	10
0.7	134.68	13	169.52	10
0.6	134.68	18	169.52	16
0.5	134.68	20	163.77	12

The prices assigned to the attributes of the dataset have many variations, which are ranged from $10.88 to $165.88. The minimum weight (min_wt) threshold value for this experiment is set to 1.0. From Table VI we can see that WFP generates association rules which are having average_weight (avg_wt) value 127.73 at confidence level 0.9, but FP-Growth produces rules which were having an average_weight value 110.46 for same level, which is less then WFP; and for confidence level 0.8 WFP produces average weight value 191.47 of top 10 association rules as compared to average weight value 125.54 of FP-Growth. Because for this type of environment WFP generates such association rules in which out of 10 rules 9 rules comprises of more than one premise item, whereas for such environment FP-Growth generates all 10 rules of single premise item. Please refer Table VI for other values of avg_wt and num_rules at remaining confidence levels.

Therefore, we conclude this experiment that the performance of WFP algorithm also remains higher in this experiment as compared to FP-Growth on the basis of rule weights as well as extra parameter i.e., num_rules for this type of environment.

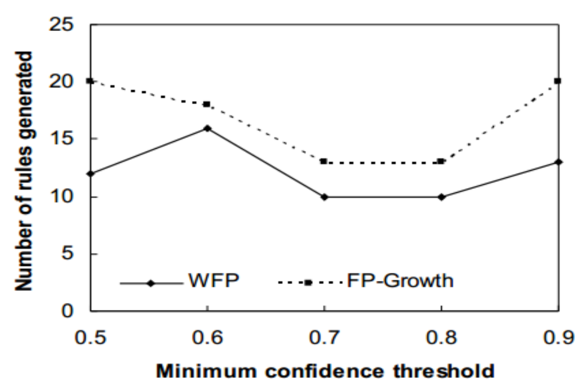

Figure 6 Number of rules generated by both algorithms at specified confidence levels for fifth experiment's environment.

Out of six experiments discussed above, WFP performs better than the traditional association rule mining method i.e., FP-Growth algorithm in four cased (Experiment 1,3,5,6), but for remaining two experiments we have seen that WFP performs better in almost all cases except for one or two confidence levels. Three types of distribution for two datasets are used whether it may be flat, manually or according to real prices, the performance of the proposed method WFP remains higher in all cases then the FP-Growth algorithm.

IV. CONCLUSION

This Research Paper proposes a modified Frequent Pattern-growth algorithm which takes into account the weights of items as well as their support counts while discovering weighted association rules. This is a generalization of the weighted association rule mining. The main issues of weighted association rules mining investigated in this work are – how good are weights as compared to support counts, and the problem of reducing the amount of association rules for achieving conciseness. The experimental results show a series of six experiments in which WARM method outperforms traditional ARM method in all confidence levels. Let us summarize the contributions in each of these areas below.

***A. Issue 1**: How weights are effective then support counts*

A variety of algorithms have been proposed in the recent past for the efficient mining of weighted as well as traditional association rules, each in turn claiming to outperform its predecessors on a set of standard databases. In this dissertation, our approach was to generate such an effective method which understands the importance of weights of items purchased from a supermarket. For this purpose, we have proposed a modified version of well established Frequent Pattern-Growth algorithm named WFP i.e., Weighted Frequent Pattern-growth. In this concept the items of transactions are assigned weights to reflect their significance. The working principle of WFP is almost same except the procedure of building the FP-Tree. We modify the ordinary method in such a way that both weights as well as support counts must take into account while mining weighted association rules, but we assign a higher priority to weights rather than support counts. Two synthetic datasets are involved to perform a set of experiments. We have shown that weights are more effective in mining process as compared to support counts, because it is up to user's choice to assign attribute weights according to his/her satisfaction. The performance of WFP has been evaluated using different combinations of weight distributions. For first two experiments we have used flat weights for items according to their support count values. But for rest of four experiments we have tried a different approach. For next two experiments i.e., third and fourth, we have tested a single attribute by assigning it a higher weight whose support count is not so higher as its weight; and for rest of two experiments we have computed real weights of attributes from real prices of items. The experimental results show that the proposed modified WFP is able to produce more versatile association rules as compared to traditional method. It is observed that WFP outperforms FP-Growth in almost all of the cases, but simple FP-Growth can performs better in some special cases like the cases in which items having more support count values then their weights.

B. Issue 2: Conciseness of Results

Generally in a supermarket scenario, the market managers have no such extra time to test all of the decisions produced by any association rule mining algorithm. They are always willing for only effective yet selected decisions, on which attraction have to be devoted. So that we have considered this issue to reduce the amount of association rules which are generated by association rule mining algorithms. It is up to the manger that what should be the exact amount of decisions he/ she requires. By setting higher value of min_wt threshold we can reduce the amount of rules in very effective manner. We can also see this effect also from our experimental results, which uses an extra parameter i.e., num_rules(number of rules) to measure the conciseness of results. Results in research work, shows that a greater amount of decisions have been pruned in very effective manner. The effectiveness is measured in terms of the weight value of top association rules. Some of our experiments i.e., first, third and fifth showed a very less amount of such relationships between attributes, and we can see the weights produced by them is more then that of weights of traditional ARM method, where as traditional methods generally generated a large amount of relationships.

V. FUTURE WORK

The work that we have presented in this Research Paper can be extended in the following ways:

A. Datasets

So far we have considered only men made datasets; the work can be extended to large and real world datasets for taking effective market-basket decisions like what to put first on store, what promotional schemes have to be effected when, how to gain higher profit from those items whose daily sales is not so higher and many more.

B. Transaction Weights

Transaction weight is another category of setting attribute weights. The work can be extended in the way of setting transaction weights either along with attribute weights (integration of attribute as well as transaction weights) or without attribute weights. The criteria of setting transaction weights may be based on the assumption that, "Good transactions contains good items", for this purpose a baseline have to be prepared first on the basis of individual weights of items.

REFERENCES

[1] Han. J and Kamber. M (2001), "Data Mining: Concepts and Techniques". New York: Morgan Kaufmann Publishers.

[2] Pang-Ning Tan, Micheal Steinbach, Vipin Kumar: Introduction to Data Mining, Addison-Wesley.

[3] U.M. Fayyad, G. Piatetsky-Shapiro, P.Symth and R. Uthurusamy, eds. Advances in Knowledge Discovery and Data Mining. AAAI Press / MIT Press, Cambridge, CA, USA 1996.

[4] F. Bonchi and B.Goethals, FP-Bonsai: The Art of Growing and Prunning Small FP-tress. Proc. 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'04, Sydney, Australia), 155-160. Springer-Verlag, Heidelberg, Germany 2004.

[5] Qiu-Yu, Cao Hua, Apriori algorithm based on Weighted Association Rule Mining, Lanzhou University of Technology, 2007, 33(6) : 69-71.

[6] G.D.Ramkumar, Sanjay Ranka and Shalom Tsur, "Weighted Association Rules: Model and Algorithm" KDD 1998, 1998.

[7] W. Wang, J. Yang and P. Yu "Efficient mining of weighted association rules (WAR)", Proc. of the ACM SIGKDD Conf. on Knowledge Discovery and Data Mining, 270-274, 2000.

[8] Sun Ke, Fengshan Bai (2008), "Mining Weighted Association Rules without Pre-assigned Weights" Proceedings of IEEE Transactions on Knowledge and Data Engineering." Vol. 20, No. 4, pp. 489-495.

[9] Tao, F., Murtagh, F. and Farid, M. (2003) Weighted Association Rule Mining using weighted support and significance framework, SIGKDD.

[10] Cai, C.H., Fu, A.W.C., Cheng, C.H. and Kwong, W.W. (1998) Mining Association Rules with Weighted Items, IDEAS.