Mining Weighted Association Rule using FP – tree

V.Vidya
Research scholar, Research and Development Centre, Bharathiar University,
Coimbatore, Tamilnadu, India
E-mail: pondymiraalfssa@gmail.com

Abstract

The main goal of association rule mining is to examine large transaction databases which reveal implicit relationship among the data attributes. Classical association rule mining model assumes that all items have same significance without assigning their weight within a transaction or record. This proposed method gives importance for the items and transactions while calculating weight on various algorithms have been represented by researchers. The proposed method combines w-support measure and the essential features of the FP-tree to reduce the time complexity. The experimental result shows that the proposed method performs better than existing method.

Keywords: Data mining, Association rules, HITS, link analysis

1. Introduction

Data mining can be defined as exploration and analysis of large quantities of data in order to discover unknown or hidden information. Data mining is applied in numerous applications like cross- marketing, market-basket problem and text mining [1] [2]. 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. The antecedent and consequent are a set of items from the domain I. Thus X∩Y=Φ. If an item set contains K items, and then the item set is called K-item set.

The support of an item set is defined as the ratio of number of transactions containing the item set 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.

$$P_r(X/Y) = P_r(X \cup Y) / P_r(X) \qquad (1)$$

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 number of transaction containing both X and Y divided by the number of transactions containing X [4]. In classical association rule mining algorithm treat each and every transaction equally, these measures are preassigned weights. Suppose data containing frequent item set I = {I1, I2, I5}. The non-empty subsets of I are {I1, I2},{I1, I5},{I2, I5},{I1},{I2}and {I5}.The resulting association rules can be listed as

I1^I2	Confidence 2/4=50%
I1^I5	Confidence 2/2=100%
I1^I5	Confidence 2/2=100%
I1→I2^I5	Confidence 2/6=33%
I2→I1^I5	Confidence 2/7=29%
I5→I1^I2	Confidence 2/2 =100%

Ke sun and Fengshan Bai[1] proposed HITS(Hyperlink Induced Topic Search) which is a combination of apriori- gen algorithm that generates candidate set and frequent item set by scanning and pruning the database. Mainly HITS is based on employing hubs and authorities for generating item set and searches for frequent occurring data set item which reinforces another item set. According to HITS all items are weighted differently and for each item weight support is generated. However, apriori –gen algorithm assumes that transaction database is memory resident. It also requires many database scans.

This paper introduces a method to use FP-Growth algorithm instead of apriori- gen algorithm to reduce space and time complexity. FP-Growth is a tree like structure which scans in descending order to produce nodes

from its root node as there is a relationship between the root node and child node transaction in an item set the combination of HITS with FP-Growth algorithm increases the speed of performance and reduce wastage of time.

2. Previous Research

Data mining is one of the problems solving technique that has made a great deal of attention. Association rule is an important technique in data mining. According to Sergey Brin et al, provided a measure called DIC (Dynamic Item set Counting) with apriori algorithm that builds large item set and make its subset also large by increasing memory size and time complexity . The advantage of DIC measure is flexible to add, delete the counted item sets, extended to parallel and incremental update versions by J.M.Kleinberg[3].A new efficient algorithm for mining binary association rule has been proposed by a measure called Apriori-gen. In apriori-gen, if an item set is large all the subsets of item set must be large. Traditional association rule model assumes that items have the same significance without taking account of their attribute within a transaction or within the whole item space.

According to Ke sun and Fenshan Bai, the improved model is weighted association rule mining. In weighted association rule mining, item sets are no longer simply counted as appeared in the transaction. This change of counting mechanism makes it necessary to adapt traditional support to weighted support. The goal of using weighted support is to make use of the weight in the mining process and prioritize the selection of target item sets according to their significance in the data set rather than their frequency alone [4] [5]. Weighted support of an item set can be defined as the product of the total weight of item set (sum of the weights of its items) and the weight of the fraction of transaction in the item set occurs. HITS are combined for calculating weighted support.

3. Ranking Transaction with HITS

The fact of transaction describes bipartite graph without any loss of information. Let $D = \{T_1, T_2, ...T_m\}$ be a list of transactions and $I = \{i_1, i_2, ...i_n\}$ be the corresponding set of items *D* is equal to the bipartite graph $G = (D, I, E)$, where $E = \{(T, I) : i \in T, T \in D, i \in I\}$.

The relationship between transaction and item is just like the relationship between hubs and authorities [1]. By considering the transaction of pure hubs and pure authorities which shows the statement or expression are equal through iteration process such as

$$auth(i) = \sum_{T:i \in T} hub(T), hub(T) = \sum_{i:i \in T} auth(i) \qquad (2)$$

TID	TRANSACTION
100	{A,B,C,D,E}
200	{C,F,G}
300	{A,B}
400	{A}
500	{C,F,G,H}
600	{A,G,H}

Table 1: The Bipartite representation of a database

When HITS model is applied all transactions are obtained, some weights represent high-value item, transaction with few items have good hub. If all sub-items are top-ranked transactions with more ordinary item have low hub weight.

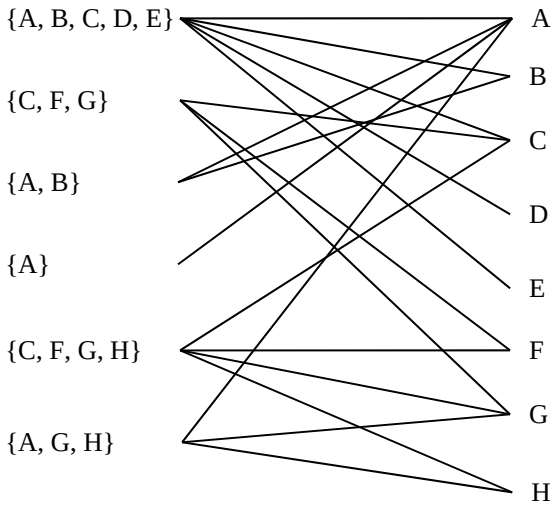

Figure 1: Bipartite graph

3.1. W-Support: A New Measuring weight

In classical association rule [2] the values are found for a group or a separate list of item on calculation. By familiarizing, a new measure called W-support which expresses weight for each and every item and transaction.

In the previous section, we have showed the application of HITS algorithm [4] is the order of processing. Step by step process focuses on the authority weight $auth(i) = \sum_{T:i \in T} hub(T)$ pointing to the "significance " of an item i.

Definition 1: The W-support of an item set X is defined as

$$W_{\sup}(X) = \frac{\sum_{T:X \in T^{T \in D}} hub(T)}{\sum_{T:T \in D} hub(T)} \qquad (3)$$

$hub(T)$ is the weight of proceeding T. An item is said to be important if its w-support is larger than a user defined value.

$$W_{\sup} = \frac{\begin{array}{l}\text{Add hub weights of transactions in}\\ \text{which item W - support need to be}\\ \text{calculated is available}\end{array}}{\text{Total hub weights}}$$

e.g

$$W_{\sup}(A) = \frac{0.518 + 0.233 + 0.148 + 0.412}{2.291}$$

$$= 0.57$$

By placing all *hub(T)* with one on the right hand side of equation (2) gives sup*(X)*.So $W_{\sup port}$ is considered as a common inference of support which employs weights of all transactions into account. The weights are not aimed by sharing values to items but there is a wider link constructed from a database.

TID	TRANSACTION	HUB WEIGHT
100	{A,B,C,D,E}	0.518
200	{C,F,G}	0.436
300	{A,B}	0.233
400	{A}	0.148
500	{C,F,G,H}	0.544
600	{A,G,H}	0.412

ITEM SET	SUPPORT	W- SUPPORT
{A}	0.67	0.57
{B}	0.33	0.33
{C}	0.50	0.65
{D}	0.17	0.23
{E}	0.17	0.23
{F}	0.33	0.43
{G}	0.50	0.61
{H}	0.33	0.42

Table 2: Hubs and $W_{\sup}$ for Database

A database in tabulation provide weights for each (transaction) proceedings items which are described in HITS iteration and W-support for individual one- item set. In table 2, finding the best hub between (transactions 500). Here CFGH is one with more item number and the most important item set (|C|) is not the largest support. This example brings out the difference between link-based and counting based proceedings in 200 and 500 T200(C, F, G), T500(C, F, G, H) there exists strong relationship between two transaction i.e., (C, F, G) in a bipartite graph as these items are to be evaluated highly since it needs to strengthen itself and adds other transaction items to be found united. Hence A has more support, which rarely shows with other valuable items, and A is graded in lower level.

W-support finds and differentiate the values eg., item sets B,F,H have support of 0.33, and F is graded first since it occurs in both the proceedings T200 and T500 with (C,G) i.e., along with good items.

Definition: 2

The W-support of an association rule X→Y is defined as

$$W_{\sup}(X \rightarrow Y) = (X \cup Y) \qquad (4)$$

For example

$$W_{sup}(A \rightarrow B) = \frac{0.518 + 0.2333}{2.291}$$

$$=0.3279$$

and the W-Confidence is

$$W_{conf}(X \rightarrow Y) = \frac{W_{\sup}(X \cup Y)}{W_{\sup}(X)} \qquad \{5)$$

$$W(C \cup F) = \frac{0.436 + 0.544}{0.518 + 0.436 + 0.233 + 0.148 + 0.544 + 0.412}$$

$$= 0.43$$

$$W_{conf}(C \rightarrow F) = \frac{0.43}{0.65}$$

$$=0.66$$

The w-confidence can be inferred as the expression to show the relation by dividing the hub weights accepted by X along with Y to the sum of hub weights accepted by X. w-support extents the importance of X and Y that appear united, if (X→Y) is big, it reflects that more good hubs that prefer for X also preferred for Y. Since the part of a whole of these hubs may be small suitably, association rule mining is to find out all rules with w-support and w-confidence with some given range of values.

3.2 A Fast Mining Algorithm

One of the mining association rule that comply with lowest possible W-support and W-confidence that can be separated into two small uncertainties.

1. To find all important items set with W-support overhead the given range of values.
2. Getting rules from item sets in step1. The first step is more important. Overlooked from the step first is very costly. The solution given in this step is to carry out that if item set fulfill some lowest possible W-support, then all its subsets comply with the smallest possible quantity. This is called downward closure property.

Proof:

Let X be an item set that fulfill $W_{\sup}(X) \geq \min(W_{\sup})$ and Y be a subset of X, we shall verify the $W_{\sup}(Y) \geq \min(W_{\sup})$ any proceedings that handles X must also handle Y ie.,

$$\{T : X \subset T, T \in D\} \subset \{T : Y \subset T, T \in D\} \subset \cup$$

In addition to hub weights in all proceedings are untrue. Therefore,

$$\sum_{T:X \subset T^{T \in D}} hub(T) \leq \sum_{T:Y \subset T^{T \in D}} hub(T)$$

Distribute each sides by $\sum_{T:T \in D} hub(T)$ then we have $W_{\sup}(X) \leq W_{\sup}(Y)$ .This gives the outcome of the item set. Based on characteristics, we can summarize the importance of item set in a level wise, behavior as the Apriori like algorithm shown in figure 3.

Initialize $auth(i)$ to I for each item i
for $(i-0;i<num\_it;i++)$
do begin
$auth'(i)=0$
for each item i
for all transactions $t \in D$ do begin
$hub(t) = \sum_{i:i \in t} auth(i)$
$auth'(i) += hub(t)$
for each item
$i \in t$
end
$auth(i) = auth'(i)$
for each item I,
normalize $auth$
end
$L_i = \{\{i\} : w_{sup}(i) \geq \min(w_{sup})\}$
for $(k=2;L_{k-1} \neq 0;k++)$
do begin
C_k =FP-Growth (L_{k-1})
for all transactions $t \in D$ do begin
C_t =subset(C_k ,t)
for all candidates $c \in C_t$ do
$c.w_{sup} += hub(t)$
$H += hub(t)$
end
$L_k = \{c \in C_k \mid c.w_{sup}/H \geq \min(w_{sup})\}$
end
Output=$U_k L_k$

Figure 2. An Algorithm for Mining Significant Item sets

3. Proposed Work

In order to mine the frequent items from the database the associations rules are performed all these rules are useful for the decision maker in order to know which items occurs more frequently in the database based on rules by specifying minimum threshold specified all these items can be extracted. The different algorithms like apriori, FP-Growth, dynamic FP-Growth and weighted associations are used. In order to determine the minimum threshold value for the apriori and FP-Growth we calculate the support and confidence and by using these values the minimum threshold value is set. To overcome the disadvantages of apriori: too much of database scanning to calculate the frequent items, to increase the performance of apriori , FP-Growth is proposed.

Calculating weight value for each item using hub and authority, using HITS algorithm to reduce number of scanning process. In which root node is created with null node, if more frequent item set transaction then child node and any other existing item node are added to the parent node to continue the process of finding the best transaction item. Again it returns to the candidate item set for item set calculation. FP-Growth is constructed and implemented as proposed work in order to speed up the memory.

Resembling various algorithms for frequent item set mining for example apriori or éclat, FP-Growth preprocesses the proceeding database as follows: In first scan, the repetition of items (support of single element item sets) are determined, and infrequent items are removed items that appear in small proceedings is not a

portion of calculating frequent item set. In addition, the items in every transaction are separated and represented in descending order with right to their repetition in the database. Though the algorithm does not rely on precise order, tests are displayed that leads to less performance than an indefinite order. A frequent pattern tree is a tree structure defined below.

1. It consists of one root labeled as "root", a set of item prefix sub-trees as the children of the root, and a frequent-item header table.

2. Each node in the item prefix sub-tree consists of three fields: item-name, count, and node-link, where item-name registers which item this node represents, count registers the number of transactions represented by the portion of the path reaching this node, and node-link links to the next node in the FP-tree carrying the same item-name , or null if there is none.

3. Each entry in the frequent-item header table consists of two fields, (1) item-name and (2) head of node-link, which points to the first node in the FP-tree carrying the item-name.

C_k =FP-Growth (L_{k-1})
Create a root node of FP- tree and label it as null
do while t is empty
insert (t,root)
link the new node to other node with similar label links originating from header list
end do
insert (t,any_node)
return FP-tree
do while t is not empty
if any_node has child node with label head_ t then increment the link count between any_node and head_t by 1
else create a new child node of any_node with label; head_t with link count 1
call insert (body_t,head_t)
end do
return C_k

Figure 3. FP-Growth Algorithm

TID	Items
1	{a,b}
2	{b,c,d}
3	{a,c,d,e}
4	{a,d,e}
5	{a,b,c}
6	{a,b,c,d}
7	{a}
8	{a,b,c}
9	{a,b,d}
10	{b,c,e}

Table 3: Transaction Data set

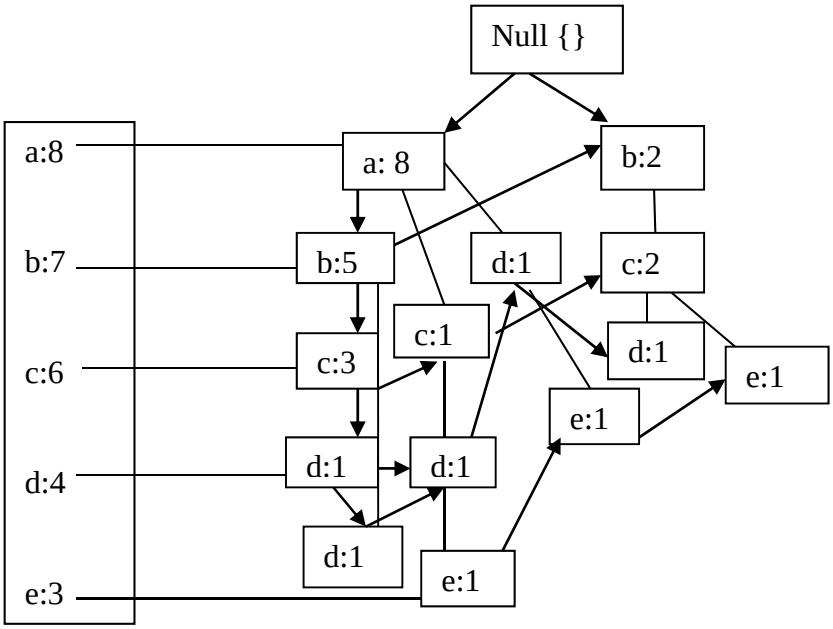

Figure 4. Sample FP-tree Construction

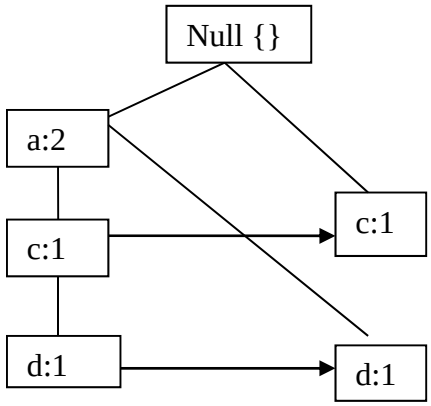

Figure 5 Conditional FP-Tree on e

Suffix	Frequent item sets
e	{e}, {d,e},{a,d,e},{c,e},{a,e}
d	{d}, {c,d},{b,c,d}, {b,d}, {a,b,d},{a,d}
c	{c}, {b,c}, {a,b,c},{a,c}
b	{b} {a,b}
a	{a}

Table 4:Frequent item sets

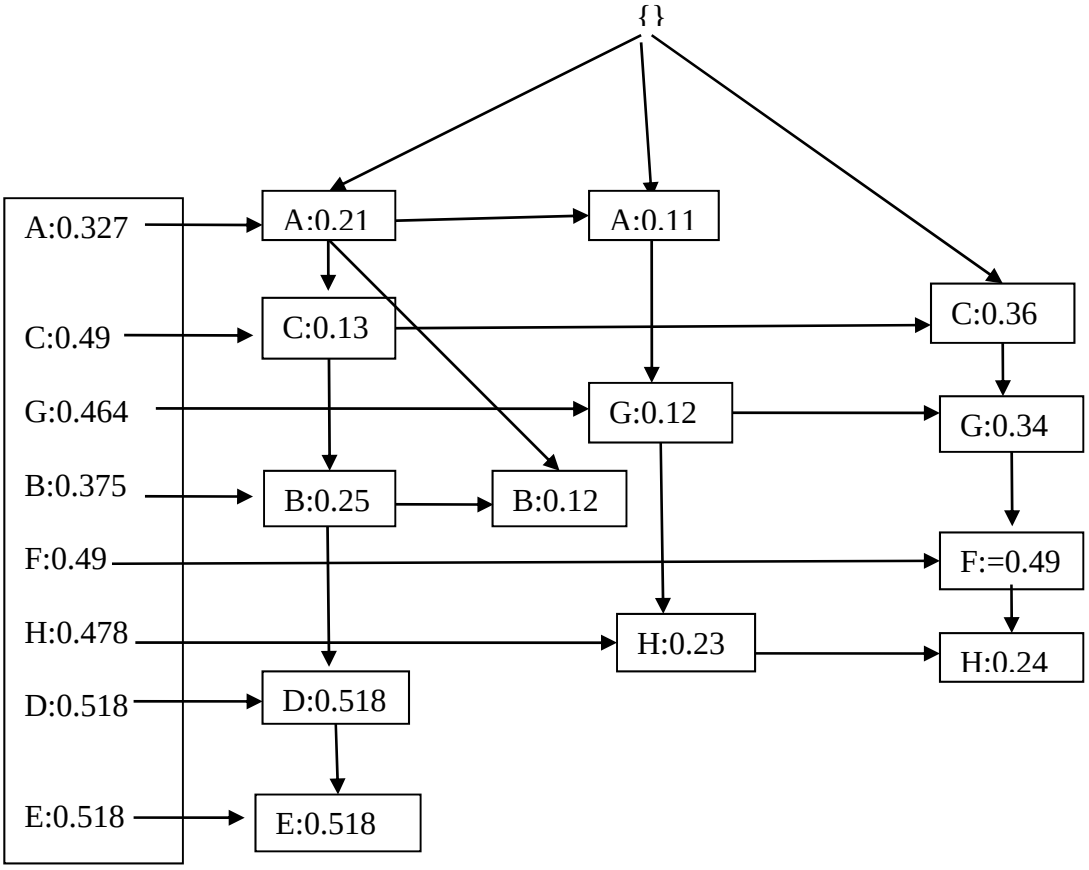

Figure 6: Example of FP-Growth Algorithm with Weight value

Conditional tree - C consists of two path A,B,C,D,E and C,F,G,H

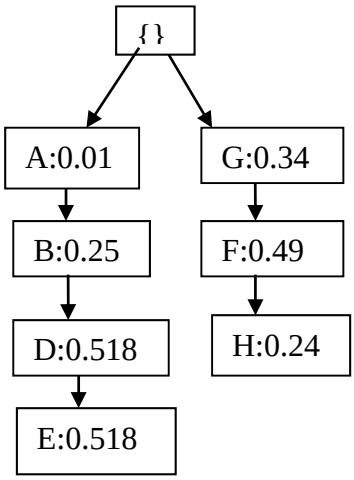

Figure 7: Prefix tree for C

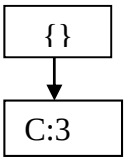

Figure 8: Conditional FP-tree for C

Suffix	Frequent item sets
H	{H}, {A,H},{A,G,H}
G	{G},{A,G},{C,F,G},{C,F,G,H}
F	{F},{C,F,G},{C,F,G,H}
E	{E},{A,B,C,D,E}
D	{D}, {A,B,C,D,E}
C	{C},{A,B,C,D,E},{C,F,G,H},{C,F,G}
B	{B},{A,B}
A	{A}, {A,B},{A,B,C,D,E},{A,G,H}

Table 5: Frequent Pattern Item set for weight value

5. Experimental Results

Instead of apriori algorithm, FP-Growth algorithm which gives effect to W-support and W-confidence as the collection which begin with various tests to capture some classical datasets. Example for FP-Growth is the Food mart dataset as input. Classify the dataset into two ways by taking the real dataset that is food mart as an input. Food marts, unit profits for items in utility tables are generated and quantities of items also generated from dataset. The experiments were conducted on a 2.84-GHz Intel Core 2 dual machine with 2 Gigabyte of RAM running Windows XP using java version 1.6.

In regard to the experimental testing database, its source was a FoodMart2000 retail transaction database embedded in a Microsoft SQL Server 2000. Since there are different kinds of transaction databases in FoodMart2000, by select sales fact 1997 data table for assessment. The number of product items in this data table is 1560. In order to effectively mine meaningful association rules, this experiment categorizes the products into groups according to the product category provided by the data table. Thus, products are classified into 34 categories, each with a corresponding product category id. In regard to data selection, 6000 customers are randomly selected along with their corresponding transaction data at different times. After arrangement, there are a total of 12,100 transaction records for these 6000 customers.

5.1 Number of rules Vs Weighted confidence

By analyzing and comparing the performance offered by Apriori with HITS and FP growth with HITS. The technique based on two parameters is number of rules and weighted confidence is compared. Here if the weighted confidence is increased the number of rules is decreased linearly. But if the number of rules generated by proposed system is high when compared with existing system. Based on the comparison and the results from the experiment show that proposed approach works better than the other existing systems. The table value is given below in Table 3.

S.No	Weighted confidence	Apriori with HITS	FP growth with HITS
1	20	80	85
2	30	75	80
3	40	70	75
4	50	65	72
5	60	60	65
6	70	53	60

Table 6: Number of rules Vs Weighted confidence

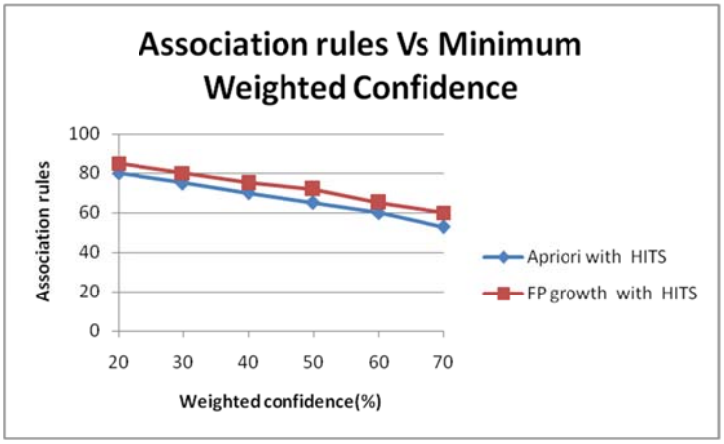

Figure 9: Number of Rules Vs Weighted confidence

5.2 Frequent item set Vs support

The performance offered by Apriori with HITS and FP- Growth with HITS are analyzed and compared. The technique is based on two parameters, frequent item set and support. Here if the support value is increased the frequent item sets decreased linearly. But the frequent item sets of proposed system producing the high frequent item set when compared with existing system. Based on the comparison and results from the experiment show the proposed approach works better than the other existing systems. The table value is given below in Table 4.

S.NO	Support (%)	Apriori with HITS	FP growth with HITS
1	10	396	485
2	20	280	370
3	30	220	280
4	40	185	220
5	50	140	180
6	60	120	150

Table 7: Frequent item set Vs support

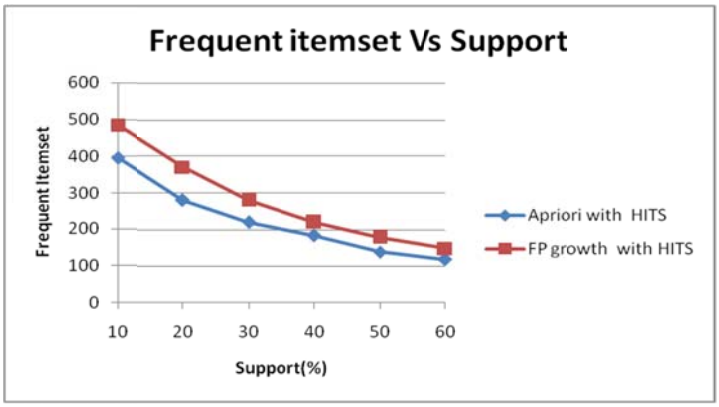

Figure 10: Frequent item set Vs support

6. Conclusion

By declaring an outline on frequent pattern growth depending on association rule and weighted association rule overcome the rule is a specimen to generate frequent item sets with large merits in cost and performance. WARM rule is to calculate frequent item set with large number of iteration to produce an outcome to be generated. So FP-tree algorithm with increased HITS providing a study is more useful to produce frequent item set. The experimental results were measured with real-time dataset foodmart between the WARM and FP growth. Proposed FP growth system performs better than the existing system with real-time dataset Foodmart. From the above paper further enhancement, based FP- tree can also refined by implementing genetic algorithm to decrease the pattern growth of an database.

7. References

[1] Ke Sun and Fengshan Bai, 2008. Mining Weighted Association Rules without Preassigned Weights. *IEEE Transaction on Knowledge and Data Engineering*, Vol 20, No.4, 2008.

[2] R.Agrawal, T.Imielinski, and A.Swami, 1993. Mining Association Rules between Sets of Items in Large Database. *Proc.ACM SIGMOD '93* pp .207-216.

[3] J.M.Kleinberg, 1999. Authoritative Sources in a Hyperlinked Environment. *J.ACM*, vol.46, no.5,pp.604-632.

[4] G.D.Ram Kumar, S.Rankaand S.Tsur, 1998. Weighted Associaton Rules: Model and algorithm. *Proc. ACM SIGKDD*.

[5] F.Tao, F.Murtagh and M.Farid,2003. Weighted Association Rules using Weighted Support Significance Framework. *Proc.ACM SIGKDD '03*,pp661-666.

[6] J.S.Park, M.Chen and P.S.Yu,1995. An effective Hash Based Algorithm for Mining Association Rules. *Proc.ACM SIGMOD*.

[7] http://tech.panorama.com/index.php/kb/57-olap-java-iis-aspnet/32

8. Biographical Notes

Vidya.V, received M.Sc degree in Computer Science from Bharathiyar University and M.Phil degree from Bharathidasan University. She is currently pursuing her doctoral degree at Bharathiar University,Coimbatore, Tamil Nadu, India.