EFFICIENT MINING OF WEIGHTED QUANTITATIVE ASSOCIATION RULES AND CHARACTERIZATION OF FREQUENT ITEMSETS

Arumugam G

Senior Professor and Head, Department of Computer Science
Madurai Kamaraj University
Madurai, Tamil Nadu, India
gurusamyarumugam@gmail.com

Vijayakumar V.K

Associate Professor and Head, Department of Computer Science
Sourashtra College
Madurai, Tamil Nadu, India
vijayakumar.sou.college@gmail.com

Abstract— In recent years, a number of association rule mining algorithms were developed. In these algorithms, two important measures viz., support count and confidence were used to generate the frequent itemsets and the corresponding association rules in a market basket database. But in reality, these two measures are not sufficient for efficient and effective target marketing. In this paper, a weighted frame work has been discussed by taking into account the weight / intensity of the item and the quantity of each item in each transaction of the given database. Apriori algorithm is one of the best algorithm to generate frequent itemsets, but it does not consider the weight as well as the quantity of items in the transactions of the database. This paper consists of two phases. In the first phase, we propose an algorithm Apriori-WQ, which extends the Apriori algorithm by incorporating the weight and quantity measures and generates Weighted Frequent Itemsets (WFI) and corresponding Weighted Association Rules (WAR). The rules are filtered based on a new measure called Minimum Weight Threshold (MWT), and then prioritized. Some itemsets may not be frequent but they satisfy MWT. Such sets are also generated. In the second phase we analyze the transactions {Ti}, which form the frequent itemsets and the customer characteristics (i.e., attributes) of those transactions {Ti}. Experiments are performed to establish a relationship between frequent itemsets and customer characteristics. 3D-graphical reports are generated, which helps the marketing leaders for making better predictions and planning their investment and marketing strategies.

Keywords: Weighted association rules, market basket database, apriori algorithm, customer characteristics.

I. INTRODUCTION

Data Mining is a process in the discovery of Knowledge from a very large database (VLDB). Association rule is one of the important techniques in data mining for extracting knowledge from a VLDB. Association rule finds the association or correlation between two sets of items. A typical example of association rule mining is market basket analysis. A market basket database is a transactional database containing Transaction Identifiers (TID) and a set of items bought by the customer. Association rule mining helps to find the buying pattern of customers, which will be very much useful to the sales managers in designing the catalog, target marketing, customer segmentation, planning the shelves and so on.

A. Basic Concepts

Let $I = \{i_1, i_2, i_3, \ldots, i_n\}$ be a set of 'n' *items*. Let D be a set of transactions, where each transaction T is a set of items such that $T \subseteq I$. Associated with each transaction is a unique identifier called *Transaction identifier* (TID). An *itemset* is defined as any nonempty subset of I. The *support count or support* of an itemset X, is the number of transactions in D that contain all the items in X. An itemset X is called a *frequent(or)large itemset (FIS),* if support $(X) \geq \alpha$ where α is a user specified minimum support threshold [3].

An *association rule* is an implication of the form $A \Rightarrow B$ [s , c], where A and B are two non-empty disjoint itemsets (i.e.,) $A \neq \phi$, $B \neq \phi$, $A \subset I$, $B \subset I$, and $A \cap B = \phi$. The *support 's'* is the percentage of

transactions in D that contain both A and B. The *confidence 'c'* is the percentage of transactions in D containing A that also contain B. An association rule $A \Rightarrow B$ [s , c] is said to be *strong* if $s \geq min\_sup$, and $c \geq min\_conf$, where *min_sup* and *min_conf* are the user specified *minimum support threshold (MST)* and *minimum confidence threshold(MCT)* respectively. Consider the following two transactions:

T1: {10 packets of Blades, 5 quantities of shaving cream}

T2: {1 packet of Blade, 1 quantity of shaving cream}

In the support-confidence frame work the above two transactions are considered to be the same, since the quantity of an item is not taken into account. But in reality, it is quite clear that the transaction T1 gives more profit than the transaction T2.

Thus to make efficient marketing we take in to account the quantity of each item in each transaction. In addition we also consider the intensity of each item, which is represented using a weight factor 'w'. In market basket analysis, 'w' may represent the retail price / profit per unit of an item. In stock-market analysis, 'w' may represent the share-price of a company and so on.

B. Problem Definition

The problem discussed in this paper consists of two phases. In the first phase, the apriori algorithm has been modified to incorporate the weight and quantity factors. we propose an algorithm Apriori-WQ, which generates Weighted Frequent Itemsets (WFI) and the corresponding Weighted Association Rules (WAR). The rules are filtered based on a new measure called Minimum Weight Threshold (MWT), and then prioritized. Some itemsets may not be frequent but they satisfy MWT. Such sets are also generated. In the second phase we analyze the transactions $\{T_i\}$, which form the frequent itemsets and the customer characteristics (i.e., attributes) of those transactions $\{T_i\}$. Textual and graphical reports between the frequent itemsets and customer characteristics are generated, which helps the corporate leaders for better planning and make intelligent predictions for an enterprise.

In the proposed work, the notations corresponding to the new structure is given below.

Universal Itemset = I = $\mathbf{\{ i_1 : w_1 , i_2 : w_2 , \ldots , i_m : w_m \}}$, where

$\{ i_1 , i_2 , \ldots , i_m \}$ represents 'm' items with respective weights $\{ w_1 , w_2 , \ldots , w_m \}$

The i^{th} transaction of the database D is of the form

$\mathbf{T_i = \{ TID_i , J_1 : Q_{i1} , J_2 : Q_{i2} , \ldots , J_t : Q_{it} \}}$, where

$\mathbf{TID_i}$ - Transaction Identifier of the i^{th} transaction

$\mathbf{\{ J_1 , J_2 , \ldots , J_t \}}$ - A subset of I

$\mathbf{Q_{ir}}$ - The quantities purchased by the customer for the r^{th} item in the i^{th} transaction.

The new algorithm called Apriori-WQ generates WFI and corresponding WAR. The rules are filtered using a new measure called Minimum Weight Threshold (MWT), and then they are prioritized based on their weights.

A frequent set may not be profitable to the marketers, whereas some of the itemsets are really profitable even though they are not frequent. In the traditional algorithms infrequent itemsets are not considered, but in this paper we consider candidate sets which are infrequent but satisfy MWT.

The remainder of this paper is organized as follows. In Section 2 we made a review of the earlier work and in Section 3 we propose our new algorithm APRIORI-WQ. Section 4 describes the implementation of APRIORI-WQ and the results of experiments on a market basket database. Finally, conclusions are drawn in Section 5 where we also indicate possible directions of future work.

II. BACKGROUND WORK

A. APRIORI algorithm

Apriori [2] is an influential algorithm for mining frequent itemsets for Boolean association rules. The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemset properties. Apriori is a step-wise algorithm. It generates the candidate itemsets to be counted in the pass by using only the itemsets found frequently in the previous pass. The key idea of the Apriori algorithm lies in the "downward-closed" property of the support, namely, if an itemset has minimum support, then all its subsets also have minimum support. Based on this property, we know that any subset of a frequent itemset must also be frequent while any superset of an infrequent itemset must also be infrequent. This observation motivates the step-wise idea to first generate frequent itemsets with only one item (called frequent 1-itemsets), then frequent 2-itemsets, and so forth.

During each iteration, only candidates found to be frequent in the previous iteration are used to generate a new candidate set during the next iteration. The candidate itemsets having k items (called candidate k-itemset) can be generated by joining frequent itemsets having k-1 items and deleting those itemsets that contain any subset that is not frequent. The algorithm terminates when there are no frequent k-itemsets. The notation is given in Table I while the detail of the Apriori algorithm is given in Table II.

TABLE I. NOTATION IN ASSOCIATION RULE MINING ALGORITHMS

Notation	Description
k-itemset	An itemset having k items
Ck	Set of candidate k-itemset (potentially frequent itemsets)
Lk	Set of frequent k-itemset (those with minimum support)

TABLE II. APRIORI ALGORITHM FOR DISCOVERING FREQUENT ITEMSETS FOR MINING BOOLEAN ASSOCIATION RULES

```
1: L1 = find_frequent_1-itemsets(D);
2: for (k = 2; Lk-1 ≠ϕ; k := k + 1) do
3: Ck = apriori_gen(Lk-1, min_sup);
4: for each transactions t ∈ D do
5: Ct = subset(Ck, t)
6: for each candidates c ∈ Ct do
7: c.count = c.count + 1
8: end
9: Lk := {c ∈ Ck | c.count ≥ min_sup}
11: end
12: return L= ∪k Lk
Procedure apriori_gen(Lk-1 : frequent (k-1) itemsets; min_sup: minimum support threshold)
For each itemset s1 ∈ Lk-1
      For each itemset s2 ∈ Lk-1 do
        If ((s1[1] = s2[1] ^ (s1[2] = s2[2] ^ . . . ^ (s1[k-2] = s2[k-2] ^(s1[k-1] < s2[k-1] ) then
           c = s1 X s2 ;
           If has_infrequent_subset(c, Lk-1 ) then
             Delete c;
           Else add c to Ck ;
       End
Return Ck ;
Procedure has_infrequent_subset(c:candidate k-itemset; Lk-1 : frequent (k-1)itemsets);
For each (k-1) subset s of c
   If s ∉ Lk-1 then
      Return TRUE;
Return FALSE;
```

A key observation exploited in the algorithm is that all subsets of a frequent itemset are also frequent. The first step when generating the frequent k-itemsets is therefore to join together frequent itemsets with k−1 items, to form C_k. This is the function of the apriori_gen function on line 3. The second step is then to delete all itemsets from C_k that have a subset that is not frequent. This is the job of the subset function on line 5. The algorithm for generating association rules using apriori algorithm is given in Table III.

TABLE III. ASSOCIATION RULES GENERATION USING FREQUENT ITEMSETS, GENERATED USING APRIORI ALGORITHM

```
for (k = 2; Lk !=∅; k++) do begin
 for each frequent k-item set S of Lk do begin
 generate association rule of the form A ⇒ B, for all subsets A, B ⊂ S
 such that A ∩ B = ∅ and A ∪ B = S
calculate conf (A ⇒ B) = {sup.count (A ∪ B) / sup.count (A) }
 if (conf (A ⇒ B) ≥ min_confidence_threshold (MCT) )
   add the rule A ⇒ B to the set of strong association rules R1
         end
end
return R1.
```

B. Weighted Association Rules

Paper [5] handles Weighted Association Rule Mining (WARM) problem. Here each item is given a weight. The goal is to find itemsets with significant weights. The problem of "Downward closure property" is solved and weighted downward closure property is found. A new algorithm WARM is developed based on the new model.

In paper [6], a weight is associated with each item in a transaction. Also a weight is associated with each item in the resulting association rule called Weighted Association Rule (WAR). It provides a mechanism to do more effective target marketing by segmenting the customers based on their potential volume of purchases.

C. Characterization of Association Rules

Paper [4] addresses the problem of mining characterized association rules from the market basket database. It discovers the buying patterns of customers and also discovers customer profiles by partitioning customer into disjoint classes. The algorithm presented in this paper combines the apriori algorithm and the AOG (Attribute Oriented Generalization) algorithm in large databases. It shows how the characterized itemsets can be generalized according to concept hierarchies associated with the characteristic attribute.

III. PROPOSED WORK AND METHODOLOGY

A. Proposed work

In the previous work, frequent itemsets [3] are generated using a variety of algorithms, But none of them have concentrated on infrequent itemsets which gives profit to the organization. Also most of the algorithms have not focused on prioritizing association rules. This paper highlights these problems and develops a new algorithm called Apriori-WQ. In addition, Apriori-WQ develops an interface between the itemsets and customer characteristics for efficient target marketing. The weight of a frequent itemset is defined as follows.

Weight of an itemset X in a transaction T_r: The weight of an itemset $X = \{x_1, x_2, x_3, \ldots x_t\}$ in a transaction T_r is defined as the sum of products of weights and quantities of items as given below:

$$W\{X, T_r\} = (q_{r1} * w_1 + q_{r2} * w_2 + q_{r3} * w_3 + \ldots + q_{rt} * w_t), \mathbf{X \subseteq T_r}$$

Where T_r - r^{th} transaction of the database

q_{ri} - The number of units purchased for the i^{th} item in the r^{th} transaction.

w_i - The weight per unit of the i^{th} item.

Weight of a frequent itemset S, with support count s : The weight of a frequent itemset $S = \{s_1, s_2, s_3, \ldots s_t\}$ with support count s is defined as

$$W\{S\} = \Sigma\, W\{S, T_r\}, \forall\, \mathbf{T_r} \supseteq S,$$ where the summation contains 's' transactions containing 'S'

Semi-frequent itemset: An itemset I is said to be *semi-frequent* if it satisfies the following conditions.

1. I is a candidate set and all subsets of which are frequent.

2. I is not frequent but satisfies MWT.(i.e.) Support count $\{I\} < MST$, and $W\{I\} \geq MWT$.

B. Objective

The aim of this paper is :

- To develop fast and efficient algorithms to find WFI and WAR
- To find the semi-frequent sets and its corresponding weight.
- To establish and visualize the relationship between the characteristics of customers and frequent itemsets.

C. APRIORI - WQ algorithm

It involves two phases.

Phase I: To identify the frequent & semi-frequent sets and to generate the corresponding association rules

It involves two steps.

- Join Step: C_k is generated by joining $L_{k-1}$with itself
- Prune Step: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset

The Apriori-WQ algorithm is given in Table IV and the algorithm for generating association rules using Apriori-WQ algorithm is given in Table V as shown below.

TABLE IV. THE APRIORI-WQ ALGORITHM

```
C1 = {1- item candidate sets};
Calculate the support count and the weight of each candidate set in C1
L1 = {1- item frequent sets };
M1 = {1- item semi-frequent sets};
for (k = 2; Lk-1 !=∅; k++) do begin
    Ck = candidates generated from Lk-1; i.e., Ck = Lk-1 X Lk-1 ;
   for each transaction t in database D do
      increment the count of all candidates in Ck that are contained in t;
      Calculate the support count and the weight of each candidate set in Ck;
   end
      Lk = {k-item freq.sets};
     Mk = { k-item semi-frequent sets};
      Arrange all freq.sets in Lk in the descending order of their weights.
    Arrange all candidate sets in Mk in the descending order of their weights.
end
return ∪k Lk and ∪k Mk;
```

TABLE V. ALGORITHM FOR GENERATING ASSOCIATION RULES USING APRIORI-WQ ALGORITHM

```
for (k = 2; Mk !=∅; k++) do begin
 for each frequent k-item set S of Mk do begin
 generate association rule of the form A ⇒ B, for all subsets A, B ⊂ S
 such that A ∩ B = ∅ and A ∪ B = S
calculate conf (A ⇒ B) = {sup.count (A ∪ B) / sup.count (A) }
 if (conf (A ⇒ B) ≥ min_confidence_threshold (MCT) )
   add the rule A ⇒ B to the set of strong association rules R2
        end
end
return R2.
```

Phase II: To identify the buying patterns of the customers based on their characteristics

1. For each frequent set S, find the TID's of the transaction 't' such that S ⊂ t and thus create a table called TID-table.
2. Create a customer table for each transaction containing the general characteristics of the customer.
3. Link the TID-table with the customer table for identifying the patterns among the customer characteristics.
4. Draw a 3-dimensional graphical chart showing the percentage of customers possessing the general characteristics.

The algorithm for generating a 3-dimensional graphical chart between customer characteristics and frequent itemsets is shown in Table VI.

TABLE VI. ALGORITHM FOR GENERATING A 3-DIMENSIONAL GRAPHICAL CHART BETWEEN CUSTOMER CHARACTERISTICS AND FREQUENT ITEMSETS

```
Create a Boolean CUST-TABLE for storing the various attributes of customer corresponding to each transaction;
for each frequent itemset S do begin
        initialize count;
for each transaction t in D do begin
                        if S ⊂ t  then S[count] = TID [t];
                            increment count;
        end
end
for each frequent itemset S do begin
            for each TID corresponding to S, link the TID in the CUST-TABLE;
                        for each attribute 'i' do begin
                            att[i]=att[i]+CUST-TABLE-TID[i];
                        end
             end
        for each attribute 'i' do begin
            if att[i] >= (sup.count [S] / 2 ) then
store the attribute along with its count for S
        end
end
Create a 2-dimensional table between frequent itemsets and customer characteristics
Draw a 3 dimensional chart showing the percentage of customers for each frequent itemset for each attribute of the customer.
 return
```

IV. EXPERIMENTS

A. Apriori algorithm

The algorithms given in this paper are implemented in C language on a Pentium-4 machine with Intel processor @1.7GHz speed , 4 GB main memory and 500 GB secondary memory. In order to highlight the difference between the two algorithms, we have applied the algorithms on a small market basket database say D1. Consider a market basket database with the Universal itemset I1 = { 1 , 2 , 3 , 4 , 5} and customer transactions as shown in Table VII.

TABLE VII. MARKET BASKET TRANSACTION DATABASE D1 WITH 5 DISTINCT ITEMS I1 = { 1 , 2 , 3 , 4 , 5}

S.No.	TID	Items bought
1	T_1	{1,2,5}
2	T_2	{2,4}
3	T_3	{2,3}
4	T_4	{1,2,4}
5	T_5	{1,3}
6	T_6	{2,3}
7	T_7	{1,3}
8	T_8	{1,2,3,5}
9	T_9	{1,2,3}

Let us assume that the minimum support threshold (i.e.,) **MST** = 2 = 2/9 * 100 = 22.22%

Using the Apriori algorithm given in Table II, we can generate the frequent itemsets as shown in Table VIII.

TABLE VIII. FREQUENT ITEMSETS WITH SUPPORT COUNT FOR THE DATABASE D1 USING APRIORI ALGORITHM

Sl.No.	*FREQUENT ITEMSETS*	*SUP.COUNT*
1	{1}	6
2	{2}	7
3	{3}	6
4	{4}	2
5	{5}	2
6	{1,2}	4
7	{1,3}	4
8	{1,5}	2
9	{2,3}	4
10	{2,4}	2
11	{2,5}	2
12	{1,2,3}	2
13	{1,2,5}	2

B. Apriori-WQ algorithm

Consider a market basket database D2 with the transaction based weights (quantity) for the itemset I2 = { 1 , 2 , 3 , 4 , 5} with respective item weights as shown in Tables IX and X respectively.

TABLE IX. TRANSACTION DATABASE D2 ALONG WITH QUANTITY

TID	Items bought with quantity				
	1	2	3	4	5
T_1	2	1	-	-	4
T_2	-	3	-	5	-
T_3	-	1	1	-	-
T_4	5	3	-	6	-
T_5	3	-	5	-	-
T_6	-	5	2	-	-
T_7	4	-	3	-	-
T_8	1	1	1	-	1
T_9	2	2	2	-	-

TABLE X. ITEMSET I2 = { 1 , 2 , 3 , 4 , 5} WITH RESPECTIVE ITEM WEIGHTS

Item	Weight
1	50
2	20
3	10
4	40
5	30

Let us assume that **MST** = 2 (or) 22.22% Using the Apriori-WQ algorithm given in Table IV, we can generate the frequent itemsets with their weights in descending order for each k-item set as shown in Table XI.

TABLE XI. FREQUENT ITEMSETS WITH SUPPORT COUNT FOR THE DATABASE D USING APRIORI-WQ ALGORITHM

S.No.	Frequent set	Support count	Weight
1	{1}	6	850
2	{4}	2	440
3	{2}	7	320
4	{5}	2	150
5	{3}	6	140
6	{1,2}	4	640
7	{1,3}	4	610
8	{2,4}	2	560
9	{1,5}	2	300
10	{2,3}	4	240
11	{2,5}	2	190
12	{1,2,5}	2	340
13	{1,2,3}	2	240

Let us assume that MWT= 300. The semi-frequent itemset for the database D2 is generated as shown in Table XII.

TABLE XII. SEMI-FREQUENT ITEMSETS FOR THE DATABASE D2 USING APRIORI-WQ ALGORITHM

S.No.	Semi-frequent itemset	Support count	Weight
1	{1,4}	1	490

Let us assume that MCT = 50%. The strong association rules are generated for both frequent and semi-frequent itemsets as shown in Tables XIII and XIV.

TABLE XIII. STRONG ASSOCIATION RULE GENERATION FOR FREQUENT ITEMSETS

2-itemsets
Freq.set: {1,2} Support count: 4 Weight:640
{1}==>{2} [4 66.00] {2}==>{1} [4 57.00]
Freq.set: {1,3} Support count: 4 Weight:610
{1}==>{3} [4 66.00] {3}==>{1} [4 66.00]
Freq.set: {2,4} Support count: 2 Weight:560
{4}==>{2} [2 100.00]
Freq.set: {1,5} Support count: 2 Weight:300
{5}==>{1} [2 100.00]
3-itemsets
Freq.set: {1,2,5} Support count: 2 Weight:340
{5}==>{1,2} [2 100.00] {1,2}==>{5} [2 50.00] {1,5}==>{2} [2 100.00] {2,5}==>{1} [2 100.00]

TABLE XIV. STRONG ASSOCIATION RULE GENERATION FOR SEMI FREQUENT ITEMSET

Semi-frequent itemset : {1,4} sup.count :1 Weight : 490
$\{4\} \Rightarrow \{1\}$ [1, 50]

The results show that frequent itemsets and strong association rules with weights give more meaning to the corporate managers for making better predictions and planning of their investment strategies. Moreover the semi-frequent itemsets gives an additional knowledge to the corporate managers about itemsets with more weight even though they are not frequent.

C. Characterization of Frequent Itemsets

This section analyzes the general behavior of the customers related to the frequent itemsets. The set of transactions related to the frequent itemsets are shown in Table 13. The customers are classified based on four important characteristics viz., *Income* {Low(A1), Middle(A2), High(A3)}, *Residence* {North(A4), South(A5), East(A6), West(A7)}, *Number of Children* {No child(A8), one child(A9), two children(A10), More than two-children(A11)}, *Working organization* {Small-scale industry(A12), Large-Scale industry(A13), Private organization(A14), Government organization(A15)}. We assume that the customers belonging to the database D1 with 9 transactions having the characteristics as shown in Table XV.

TABLE XV. SET OF TRANSACTIONS FOR THE FREQUENT ITEMSETS

Code	Frequent set	Support count	TID's
F1	{1}	6	T1,T4,T5,T7,T8,T9
F2	{4}	2	T2,T4
F3	{2}	7	T1,T2,T3,T4,T6,T8,T9
F4	{5}	2	T1,T8
F5	{3}	6	T3,T5,T6,T7,T8,T9
F6	{1,2}	4	T1,T4,T8,T9
F7	{1,3}	4	T5,T7,T8,T9
F8	{2,4}	2	T2,T4
F9	{1,5}	2	T1,T8
F10	{2,3}	4	T3,T6,T8,T9
F11	{2,5}	2	T1,T8
F12	{1,2,5}	2	T1,T8
F13	{1,2,3}	2	T8,T9

TABLE XVI. CUSTOMER CHARACTERISTICS TABLE

TID	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
T1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
T2	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1
T3	1	0	0	0	0	1	0	0	1	0	0	0	0	1	0
T4	0	0	1	0	0	0	1	0	1	0	0	0	1	0	0
T5	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1
T6	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0
T7	0	1	0	0	1	0	0	0	1	0	0	0	0	0	1
T8	0	0	1	0	1	0	0	0	1	0	0	0	0	0	1
T9	0	0	1	0	1	0	0	0	0	1	0	0	0	1	0

Consider the first frequent itemset F1={1} whose support count is 6 and the transactions corresponding to F1 are { T1,T4,T5,T7,T8,T9}(Refer Table XV) . Let us extract the customer characteristics from Table XVI corresponding to these transactions for the frequent itemset F1= {1} as shown in Table XVII.

TABLE XVII. CUSTOMER CHARACTERISTICS TABLE RELATED TO TRANSACTIONS FOR F1={1}

TID	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
T1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
T4	0	0	1	0	0	0	1	0	1	0	0	0	1	0	0
T5	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1
T7	0	1	0	0	1	0	0	0	1	0	0	0	0	0	1
T8	0	0	1	0	1	0	0	0	1	0	0	0	0	0	1
T9	0	0	1	0	1	0	0	0	0	1	0	0	0	1	0
Sum= n(A_i)	1	2	3	0	4	1	1	0	3	2	1	0	1	1	3

Now the percentage for an attribute A_i for a frequent itemset F_j is calculated using the formula:

percent(A_i) = n(A_i) * 100 / Support Count of F_j , where i=1 to 15 and j=1 to 13. For example in Table 16, we can calculate the attribute percentage as follows:

percent(A_1) = n(A_1) * 100 / Support Count of F_1 = 1 * 100 / 6 = 16.67%

percent(A_2) = n(A_2) * 100 / Support Count of F_1 = 2 * 100 / 6 = 33.33%

percent(A_3) = n(A_3) * 100 / Support Count of F_1 = 3 * 100 / 6 = 50%

percent(A_4) = n(A_4) * 100 / Support Count of F_1 = 0 * 100 / 6 = 0%

percent(A_5) = n(A_5) * 100 / Support Count of F_1 = 4 * 100 / 6 = 66.67%

percent(A_6) = n(A_6) * 100 / Support Count of F_1 = 1 * 100 / 6 = 16.67%

percent(A_7) = n(A_7) * 100 / Support Count of F_1 = 1 * 100 / 6 = 16.67%

percent(A_8) = n(A_8) * 100 / Support Count of F_1 = 0 * 100 / 6 = 0%

percent(A_9) = n(A_9) * 100 / Support Count of F_1 = 3 * 100 / 6 = 50%

percent(A_{10}) = n(A_{10}) * 100 / Support Count of F_1 = 2 * 100 / 6 = 33.33%

percent(A_{11}) = n(A_{11}) * 100 / Support Count of F_1 = 1 * 100 / 6 = 16.67%

percent(A_{12}) = n(A_{12}) * 100 / Support Count of F_1 = 0 * 100 / 6 = 0%

percent(A_{13}) = n(A_{13}) * 100 / Support Count of F_1 = 1 * 100 / 6 = 16.67%

percent(A_{14}) = n(A_{14}) * 100 / Support Count of F_1 = 1 * 100 / 6 = 16.67%

percent(A_{15}) = n(A_{15}) * 100 / Support Count of F_1 = 3 * 100 / 6 = 50%

Let us fix a user defined Minimum Attribute Percentage Threshold (MAPT) = 50% . In such case the significant attributes related to frequent itemset F1={1} are A3, A5,A9 and A15 whose MAPT is greater than or equal to 50% , which can be seen in the first row of Table XVIII related to frequent itemset F1={1}. In a similar way, we can calculate the attribute percentage for each frequent itemsets from F2 to F13 and the results are summarized as shown in Table XVIII. The graph in Fig.1 shows the relationship between the frequent itemsets and attributes.

TABLE XVIII. RELATIONSHIP BETWEEN CUSTOMER CHARACTERISTICS AND FREQUENT ITEMSETS

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
F1	0	0	50	0	66	0	0	0	50	0	0	0	0	0	50
F2	50	0	50	0	50	0	50	0	50	0	50	0	50	0	50
F3	0	0	57	0	57	0	0	0	0	0	0	0	0	0	0
F4	0	50	50	0	50	50	0	0	50	50	0	0	0	50	50
F5	0	0	50	0	83	0	0	0	50	0	0	0	0	0	50
F6	0	0	75	0	50	0	0	0	50	50	0	0	0	50	0
F7	0	0	50	0	100	0	0	0	50	0	0	0	0	0	75
F8	50	0	50	0	50	0	50	0	50	0	50	0	50	0	50
F9	0	50	50	0	50	50	0	0	50	50	0	0	0	50	50
F10	0	0	75	0	75	0	0	0	50	50	0	0	0	50	0
F11	0	50	50	0	50	50	0	0	50	50	0	0	0	50	50
F12	0	50	50	0	50	50	0	0	50	50	0	0	0	50	50
F13	0	0	100	0	100	0	0	0	50	50	0	0	0	50	50

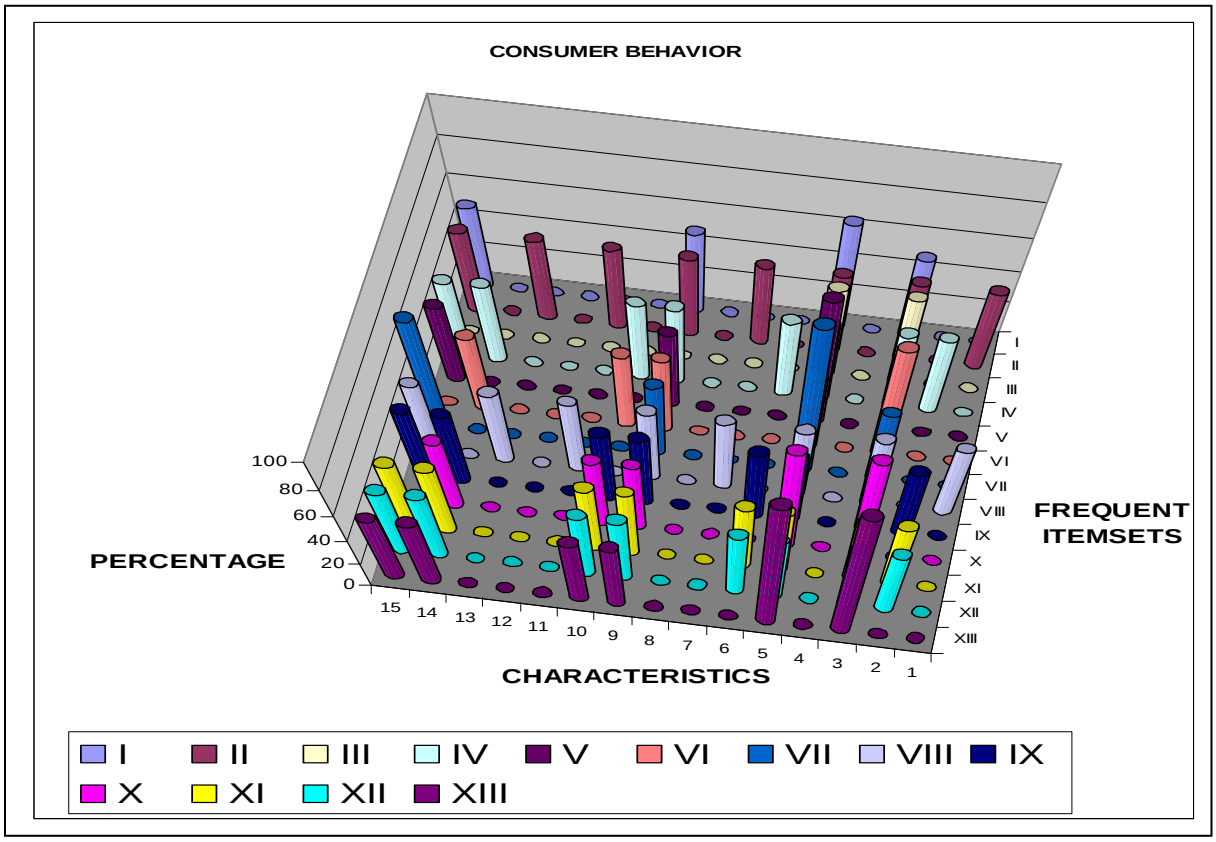

Figure 1. Relationship between frequent itemsets and attributes

D. Result Analysis

The graph in Fig.1 gives us the knowledge about the customer attributes and their buying habits.

- A minimum of 50% of people belong to **high income** family.
- A minimum of 50% of customers are coming from the **north** part of the city.
- A minimum of 50% of people are having **one child.**
- A minimum of 50% of people are working in **Govt. organization**.
- No customer is coming from **SOUTH** part of the city. In order to attract people from SOUTH, we can introduce a special attractive offer to them.
- Every customer is having atleast one child.

- 50% of the customer who purchase frequent itemets II & VIII are belong to low-income group, from West part of the city, having more than two-children and working in private companies

From the above observations one can devise marketing strategies for customers who belong to **HIGH-INCOME** group, residing in **NORTH** part of the city, having **ONE-CHILD** and working in **GOVT. ORGANIZATION**.

V. CONCLUSIONS AND FURTHER WORK

In this paper the following issues are highlighted.

- The weight and quantity factor given to each item in each transaction is used to find the weight of each frequent set that gives the importance of each frequent set. Two frequent sets may have same support count but the weight (or) profit of a frequent set decides which is more important. The marketers are more interested in the profit than the frequency of a set.
- Frequent itemsets and corresponding association rules were prioritized based on their weights.
- The semi-frequent sets may be useful to the sales managers as it satisfies MWT.
- The common characteristics for each frequent sets are identified and a three-dimensional graphical chart between the frequent sets and customer attributes is generated. Such graphical report will be of very much useful to the marketing leaders to promote their business by devising suitable marketing strategies.
- The proposed work in this paper can be applied on a large database. In apriori algorithm, frequent itemsets are generated using candidate generation phase, which is a time-consuming process. Instead other algorithms like FP-Growth can be used, which generates frequent itemsets without candidate generation and hence speed up the process.

REFERENCES

[1] R. Agrawal, T. Imielinski and A. Swami, "Mining Association Rules Between Sets of Items in Large Databases", *Proc. 1993 ACM SIGMOD,*Washington, DC, pp. 207–216, May 1993
[2] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules", *Proc. 20th International Conference on Very Large Databases* (VLDB'94), Santiago, Chile, pp. 487–499, Sept. 1994
[3] J. Han and Micheline Kamber, "Data Mining – Concepts and Techniques", Morgan Kaufmann Publishers, 2001
[4] Robert J. Hilderman, Colin L. Carter, Howard J. Hamilton, And Nick Cercone, "Mining Association Rules From Market Basket Data Using Share Measures and characterized Itemsets"
[5] Feng Tao, Fionn Murtagh, Mohsen Farid, "Weighted Association Rule Mining using Weighted Support and Significance framework"
[6] Wei Wang, Jiong Yang, Philip S.Yu, "Efficient Mining of Weighted Association Rules (WAR) "