AnalyzeMarket Basket Data using FP-growth and Apriori Algorithm

Ankur Mehay
Department of Computer Science
Punjabi University
Patiala, India
ankur.mehay@yahoo.com

Dr. Kawaljeet Singh
University Computer Centre
Punjabi University
Patiala, India
singhkawaljeet@rediff.com

Dr. Neeraj Sharma
Department of Computer Science
Punjabi University
Patiala, India
sharmaneeraj@hotmail.com

***Abstract*— In this paper we find the association rules among the large dataset. To find association rules we use two algorithms i.e. FP-growth and Apriori algorithms. First we find frequent itemsets using Weka tool and Rapid-miner tool. Then we generate association rules from the frequent itemsets. We have analyzed that as per this research FP-tree much faster than Apriori algorithm to generate association rules when we use large dataset.**

Index Terms — *Data mining,apriori, FP-growth, FP-tree, market basket analysis, association*

*****

I. INTRODUCTION

Data mining is the process of analyzing or extracting large amount of data from different perspectives and summarizing it into useful information. That information can used to increase revenue, cost, or both. Data mining is also known as knowledge discovery from data. Data Mining helps experts to understand the data and lead to good decisions. Central Statistical Organization (CSO) classification of the services sector consists of four broad categories viz., (i) trade, hotels and restaurants; (ii) transport, storage and communication; (iii) financing, insurance, real estate and business services (iv) community, social and personal services. In alignment with global trends, the Indian services sector has witnessed major boom. Services sector in India today accounts for more than half of India's GDP [8]. There has been a marked acceleration in services sector growth. Availability of quality services is vital for the well-being of the economy. Academic circles are very seriously covering all important aspects of the services sector, the challenges ahead and the strategies that need to be adopted for further strengthening the sector to help achieve more inclusive and balanced growth.

Market basket analysis is one of the most common and useful technique of data analysis for marketing and retailing. The main purpose of Market basket Analysis is to determine what products are usually bought together by the customer. Market basket analysis identifies customers purchasing habits. It provides insight into the combination of products within a customer's 'basket'. The term 'basket' normally applies to a single order. However, the analysis can be applied to other variations. However, we often compare all orders associated with a single customer.

In Market Basket Analysis we can analyze, which type of person wants which type of product and this will be helpful for both retailer as well as manufacturing company in order to maintain the inventory. A store could use this analyzed information to place products frequently sold together into the same area, so that store product selling gets increased. This information will enable the retailer to understand the buyer's needs and rewrite the store's layout accordingly. [3][4] Suppose a manger of an electronics store wants to learn about the buying habits of their customers so that one can determine "which groups or sets of items are customer's likely to purchase on a given trip to the store?" To do this, market basket analysis may be performed on the retail data of customer transactions. This will help to plan marketing, designing new catalog so that sale of the store increases. The strategy for designing is that, items that are frequently purchased together are placed into the same area. If a customer who is purchasing a computer gets an antivirus within the same vicinity, then there is high probability that he'll buy the antivirus and such planning's tend to increase the sales. Market basket analysis can also help retailers plan which items to put on sale at reduced prices. [5]

II. ASSOCIATION

Suppose set of items are available at the electronics store. We represent each of the item's with one of the two Boolean variables. This tells the presence or absence of the item. So we can represent each item in the basket by Boolean vector of the values assigned to these variables. The Boolean vectors then analyzed for buying patterns reflect the items that are frequently associated or purchased together. These patterns can be represented in the form of association rules. Support and confidence are the two measures of rule interestingness. From the example of electronics store. If a customer buys a computer he also tends to buy antivirus software at the same time. If support is 2% and confidence is 60%. It tells that 2% of the transactions under analysis show that computer and software are purchase together. A confidence 60% means that 60% of the customers who purchased computer also bought the software. Association rules are considered interesting if they satisfy both minimum support threshold and minimum confidence threshold. Users or domain experts can set these thresholds. [5] We have dataset in which different attributes represents that data item is present or not by using binary value 0 or 1. 0 represents value missing and 1 represent value present. Let I= $\{I_1,I_2,I_3..Im\}$ be a set of items in the store. Let D is the database which contains transactionT.TransactionT=$\{TID_1,TID_2,TID_3\ldots TIDn\}$ is the set of items such that $T\subseteq I$. Let X and Y are the two transactions. A transaction T is said to contain X if and only if $X\subseteq I$. An association rule is an implication of the form X Y, where $X\subset I$, $Y\subset I$ and $X\cap Y=\phi$. In this association rule, X is call as antecedent and Y is call as consequent. X and Y are sets of items and the rule means that customers who buy X are likely to buy Y. The rule X Y holds in the transaction set D with support s, where s is the percentage of transaction in D that contain $X\cup Y$. This is taken as the probability, $P(X\cup Y)$. The rule has confidence c in the transaction D, where c is the percentage of transactions in D containing X that also contain Y. This is taken to be conditional probability, P (Y|X). Association rule mining can be solve in two steps:

A. **Find all frequent item sets**: Each of the item set will occur at least as frequently as minimum support.

$$\text{Support } (X \Rightarrow Y) = P(X\cup Y)$$

Support tells us that if we have X and Y item collectively, and its value is above the minimum support value after applying in all the transactions. Then X and Y are frequent item sets. Minimum support will be specified by the user.

B. **Generate strong association rules from the frequent item sets:** These rules must satisfy minimum support and minimum confidence.

$$\text{Confidence}(X \Rightarrow Y) = P\ (X|Y)$$

$$\text{Where, } P\ (X|Y) = \frac{\text{Support } (X\cup Y)}{\text{Support } (X)}$$

Confidence can be count after generating the frequent itemsets. It determines how frequently items in Y appear in transactions that contain X.

Rules that satisfy both a minimum support threshold and minimum confidence threshold are called **association rule.** These threshold values are set by the experts. Both threshold and confidence values lies from 0% to 100%.

III. APRIORI ALGORITHM OVERVIEW

Apriori Algorithm is famous algorithm used by most of the researcher. It was proposed by R.Agrawal and R. Srikant in 1994 for mining the frequent itemsets. Apriori is an iterative process where k-itemsets are used to explore (k+1) itemsets. The algorithm terminates when frequent itemsets cannot be extended any more. But it has to generate a large amount of candidate itemsets and scans the data set as many times as the length of the longest frequent itemsets. Apriori algorithm can be written by pseudo code as follows.[6]

Input: number of transactions (t) i.e. dataset, minimum support(minsup).

Suppose $L_1,L_2,L_3\ldots\ldots L_k$ is the itemsets where L_1 is the 1-itemset, L_2 is the 2-itemset and L_k is the k-itemset. D is the dataset. C_k is the K candidate itemset. [1][2]

```
First of all find the frequent of 1-itemset i.e.$L_1$ from the dataset D.
For(k=2;$L_{k-1} \neq \phi$;k++)
{
$Ck$ = Apriori_gen( $Lk-1$ , minsup);
For each transaction t ε D.
{
    $C_t$=subset($C_k$,t); //candidates contained in t
     For each candidate c ε $C_t$ do.
     c.count++;
}
Lk={c ε $C_k$ |c.count≥minsup};
}
```

13. Return L={L_1UL_2UL_3….L_k}.

IV. FP-GROWTH ALGORITHM

FP stands for frequent pattern. It generates frequent itemsets without the use for candidate generation. FP-growth adopts the divide and conquer strategy. FP-Growth algorithm encodes the data set using compact data structure called an FP-tree and then extracts frequent itemsets directly from this structure. It is based on a prefix tree representation of the given database of transactions, which can save considerable amounts of memory for storing the transactions. The basic idea of the FP-growth algorithm can be described as to generate the FP-tree for all the transactions.[7] Every path of FP-tree represents a frequent itemset and the nodes in the path are stored in decreasing order of the frequency of the corresponding items.[6]

Calculate Fp-tree:[5]

Step 1: Input the transactional database D and minimum support.

Step 2: Now generate the FP-tree. Scan the transaction database D once. Collect F, the set of frequent Items, by applying their support count. Sort F in support count descending order as L, the list of frequent items.

Step 3: Create the root of an FP-tree, and label it as “NULL”. For each transaction TID in D do the following:

A. Select and sort the frequent items in TID according to the order of L. Let the sorted frequent item list in TID be [q|R], where q is the first element and R is the remaining list.

B. Call insert_tree ([q|R], Tr), which is performed as follows. If Tr has child ND such that ND.item-name=q.item-name, then increment ND’s count by 1; else create a new node ND, and let its count be 1, its parent link be linked to Tr, and its node-link to the nodes with the same item-name via the node-link structure. If R is non-empty, call insert_tree(R,ND) recursively. Below shown the sample of ten transactions in figure1.1.

1	{m,p,r}	6	{m,n,p}
2	{m,o,p,q}	7	{n,p,q}
3	{n,p}	8	{n,o,q,s}
4	{n,o,p}	9	{o,p,r}
5	{n,o}	10	{m,n,p}

Figure 1.1 : It represents the transactions

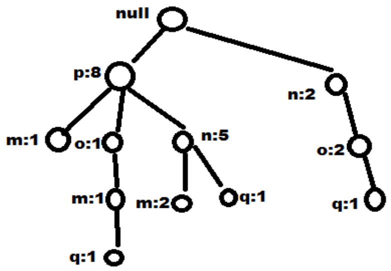

Figure 1.2: generation of FP-tree for ten transactions

Step 4: After creating FP-tree, now call the FP-growth (FP_tree, NULL) for mining, which is implemented as:

A. Algorithm for FP-growth:

FP-growth (Tree,α)[5]

```
1. If Tree contains a single path P, then
2. For each combination (denoted as β) of the nodes in the path P.
3. Generate pattern βUα with support_count= minimum support count of nodes in β:
4. Else for each $a_i$ in the header of Tree{
5. Generate pattern β=$a_i$Uα with support_count = $a_i$.support_count;
6. Construct β’s conditional pattern base and then β’s conditional FP-tree $Tree_β$;
7. If $Tree_β$≠ϕ then
8. Call FP_growth($Tree_β$,β);}
```

V. EXPERIMENT RESULTS

A synthetic dataset has been used with 50 items each for analysis. A set of association rules are obtained by applying Apriori algorithm and FP-growth. By analyzing the data, and giving different support and confidence values, we can obtain different number of rules. During analysis it found that FP-growth is much faster for large number of transactions as compare to apriori. It takes less time to generate frequent itemsets. We work on synthetic data which contains 75000 transactions. All the results are collected from Pentium Dual core processor with 1.73GHz speed and 1-GB RAM.

Support	Confidence	Apriori	Fp-growth	Rules
0.001	0.2	53s	38s	1115
0.009	0.2	49s	34s	352
0.01	0.2	44s	31s	352
0.05	0.2	41s	29s	4

VI. CONCLUSION AND FUTURE WORK:

In this paper we analyze the apriori and FP-growth algorithm. It found that apriori algorithm takes more time to compute association rules, even both contain same number of transactions. FP-growth is much faster than apriori because there is no candidate generation, it uses compact data structure, it eliminates repeated transaction scan. In future Apriori algorithm needs to improve so that it takes less time to find association rules on large datasets.

VII. REFERENCE

[1] Agrawal, R., T. Imielinski and A. N. Swami, Mining Association Rules between Sets of Items in Large Databases", Proceedings of the 1993 ACM SIGMOD In- ternational Conference on Management of Data, pp. 207{216, Washington, D.C.,May 1993.

[2] Aggrawal, R. and R. Srikant,

"Fast Algorithms for Mining Association Rules in Large Databases", Proceedings of the 20th International Conference on Very Large Data Bases (VLDB), Santiago, Chile, September 1994.

[3] "Market Basket Analysis "

http://web.fhnw.ch/personenseiten /taoufik.nouri /Data% 20Mining / Course/Case%20Study/PA-Tutorial/mba.html

[4] "Information drivers"

http://www.information-drivers.com/market_basket_analysis.php

[5] Jiwan Han and Micheline Kamber, Data mining, concepts and techniques. Morgan Kaufman: 2009

[6] Bo Wu, Defu Zhang, qihua lan, Jiemin heng, "An efficient frequent patterns mining algorithm based on Apriori Algorithm and the FP-tree structure", Third international conference on convergence and hybrid information technology,2008

[7] Christian Borgelt,"An Implementation of the FP-growth Algorithm" ,Department of knowledge Processing and language Engineering, Germany

[8] Srinivasan G, "Services Sector and its Contribution to the Indian Economy", Yojana, Volume 55, pp. 5-7, September 2011