Analyzing Customer Buying Habits Using Transaction Data

Hlaing Nwe Tun ,Thandar Win
University of Computer Studies, Yangon
hlaingnwetun08@gmail.com

Abstract

Data mining is the process of digging trough large files and databases to discover useful, non-obvious and often unexpected trends and relationships. Association rules are popular representations in data mining. It finds interesting association or correlation relationships among large set of data items. Most of the recent years, a very influential association rule mining algorithm, Apriori, has been used. It is to find frequent patterns, which produces candidate generation and multiple scans of database. Therefore, it is time consuming. Frequent pattern mining (FP-growth), is another milestone in development of association rules mining, which breaks the main bottlenecks of the Apriori. The frequent itemsets are generated with only two passes over the database and without any candidate generation process. This paper presents user buying habits using the sales transaction of stationery and FP-growth algorithm in association rule mining which is efficient and without candidate generation.

1. Introduction

Association rule mining is one of the most important and well researched techniques of data mining. Its aim to extract interesting correlations, frequent patterns, associations or casual structures among set of items in the transaction databases or other data repositories. The extracted knowledge is valuable for organization or company, this knowledge provide them precise and up-to-date information required for decision making. This proposed system, one year of customer buying items collects in database, and users or managers can easy to extract (or) see the sales transaction by date, category, brand and finally this data can be used to analyze of customer buying habits by using Fp-growth algorithm to find a correlation between purchase of stationery items (which items are how to associate to buy with other item) , association of category , association of brand. The results from this proposed system, can help users to make better decision., and to figure out which item or combination of item, category and brand should be put together on sales and other cases can be applied as user desire. As a result of this finding, the stationery shop is alleged to have items next to associate other items, resulting in increase sales of both. Finally, sales managers or stationer can evaluation of retail promotion and plan their shelf space based on user buying habits information. Therefore, improving customer satisfaction- once they've found one of the items they want, the customer does not have to look all over the store for something they want to buy then items that sell together should be found together.

2.Related Work

Association rules are widely used in various area such as telecommunication network, market and risk management, inventory control, etc. The work on frequent patterns mining started with development of AIS algorithm. Since then, there have been several algorithms to find frequent patterns. Among them, Apriori is the most widely used algorithm for finding frequent patterns [3]. But, there are two bottleneck of Apriori algorithm, one is complex candidate generation process that uses most of time, space and memory. Second is the multiple scans of database.[4] Fp-growth algorithm breaks the main bottlenecks of the Apriori i.e. only two passes over database and without candidate generation process and generates frequent patterns. Therefore, most researchers using FP-growth to find association of different items (products) and association of words.[4]. Moreover, researchers compare performance of Apriori and FP-growth algorithm using 10000 with transaction records.[5].This proposed system presents to find association rules using sales transaction of stationery and Fp-growth algorithm.

3. Background Theory

3.1 Association Rule Mining

Association rules have been widely used to determine customer buying patterns from market basket data. This process analyzes customer buying habits by finding associations between the different items that customers in their shopping baskets. Association rule mining is to find out association rules that satisfy the predefined minimum support and confidence from a given database.

Support : The support is the number of transaction that include all items in the antecedent and consequent parts of the rule.

Confidence : The confidence is the ratio of the number of transactions that include all items in the consequent as well as the antecedent (the support) to the number of transactions that include all item in the antecedent.

confidence (A → B) = P(B|A)

$$= \frac{sup\_count(A \cup B)}{sup\_count(A)} \qquad (1)$$

Association rule mining is a two step process:

(i) Find all frequent itemsets : By definition, each of these itemsets will occur at least as frequently as a pre-determined minimum support count.

(ii) Generate strong association rules from the frequent itemsets : By definition, these rules must satisfy minimum support and minimum confidence.[3]

3.2. Frequent Pattern Growth

One of the most cite algorithms proposed after Apriori is the FP-Growth. It is efficient and scable for mining both long and short patterns without candidate generation and only two passes over the database.It is based on a prefix tree representation of given database of transactions called **FP-tree** which can save considerable amount of memory for storing the transactions. A FP-tree is a compact representation of all relevant frequency information in a database. The frequent patterns generation process has two steps:

(i) Construct FP-Tree

(ii) Generate frequent patterns from FP-tree.

3.2.1 Construction of FP-Tree

FP-tree consists of one "null" root, a set of item-prefix subtrees as the children of root and frequent-item header table. Each node in the item-prefix subtrees consists of item-name, count and node-link fields. Each entry in the frequent-item header table consists of item name and head of node-link fields. Every branch of FP-tree represents a frequent itemset, and the node along the branches are stored in decreasing order of frequency of the corresponding items, with leaves representing the least frequent items. In step 1, database is scanned one and then frequent 1-itemsets found. Next, frequent items are ordered in frequency descending criteria. Finally, scan the database again and construct the FP-tree.[5]

FP-tree is highly compact and a much smaller than its original database, and thus saves the costly database scans in the subsequent mining process.

3.2.2 FP-Growth

Based on FP-tree structure, an efficient frequent pattern mining algorithm, FP- growth method is proposed, which is a divide-and-conquer methodology: decompose mining task into smaller ones, and only need sub-database test. FP-growth performed as follows:

1. For each node in the FP-tree construct its conditional pattern base, which is a "sub-database" constructed with the prefix subpath set co-occurring with the suffix pattern in the FP-tree. FP-tree traverses nodes in the FP-tree from the least frequent item;
2. Construct conditional FP-tree from each conditional pattern base;
3. Execute the frequent pattern mining recursively upon the condition FP-tree. If the conditional FP-tree contains the single path, simply enumerate all the patterns.[3]

3.2.3. FP-Growth Algorithm

Input : A transactional database and minimum support threshold s.

Output : Its frequent pattern tree, fp-tree.

Method : The fp-tree is constructed in the following steps:

1. Scan transactional database DB once. Collect the set of frequent items F and their support. Sort descending order as L, the list of frequent item.
2. Create the root of an fp-tree T, and label it as "root". For each transaction *Trans* in DB do the following.
    a. Select and sort frequent items in Trans according to the order of L. Let the sorted frequent item list in *Trans* be | *p* | *P* |, where p is the first element and P is the remaining list, Call insert-tree (| *p* | *P* |,*T*)
    b. The function *insert-tree* (| *p* | *P*|,*T)* is perform as follows. If *T* has a child N such that *N.item-name = p.item-name*, then increment *N's* count by 1; else create a new node *N*, and let its count be 1, its parent link be linked to *T*, and it node-link be linked to the nodes with the same item-name via the node-link structure. If *P* is nonempty, call *insert-tree(P,N)* recursively.

The fp-growth algorithm for mining frequent patterns with fp-tree by pattern fragment growth is:

Input : a FP-tree constructed with the above mentioned algorithm;

D – transaction database;
s – minimum support threshold.
Output : The complete set of frequent patterns.
Method :
call Fp-growth *(FP-tree,null)*
Procedure FP-growth *(Tree,A)*
{
if *Tree* contains a single path *P*
then for each combination (denoted as *β*) of the nodes in the path *P* **do**
generate pattern *β U A* with *support =minimum support of nodes* in *β*
else for each *ai* in the header of the *Tree* **do**
{
generate pattern *β = ai U A* with *support = ai.support* ;
construct *β's* conditional pattern base and *β's* conditional FP-tree *Treeβ*
if *Treeβ ≠ Ø* **then**
call Fp-growth *(Treeβ, β)*
}
}

3.3. Advantages of FP-Growth

- only two passes over data-set
- compress data-set
- no candidate generation
- much faster than Apriori
- efficient and scalable for mining both long and short frequent patterns [2]

3.4. Disadvantages of FP-Growth

- FP-tree may not fit in memory
- FP-tree is expensive to built [2]

4. Proposed System Architecture

This proposed system presents user buying habits (or) association of purchase of stationery items (which items are how to associate to buy with other item) using one year sales transactions from stationery database and Fp-growth algorithm. Firstly, user desires date to get sales transaction from database and then from these transactions to find frequent 1-itemset and their support count. Then, sort descending order of these itemsets according to their support count. Second, scans sales transaction from database to built FP-tree. And then built Fp-tree using these transaction. After built FP-tree, we get conditional pattern base of each item from these FP-tree. Conditional pattern base consists of set of prefix paths .

And then construct conditional fp-tree from each conditional pattern base and recursively mining frequent patterns upon the conditional fp-tree. From these conditional fp-tree, we get frequent patterns. Among these frequent patterns we remove some frequent patterns because these frequent patterns are less than minimum support. And then, calculate the confidence of remaining frequent patterns by using the formula of association rule, see in (1) and from these frequent patterns remove some of frequent patterns which are not suitable of user desire minimum confidence. See in Figure 1.

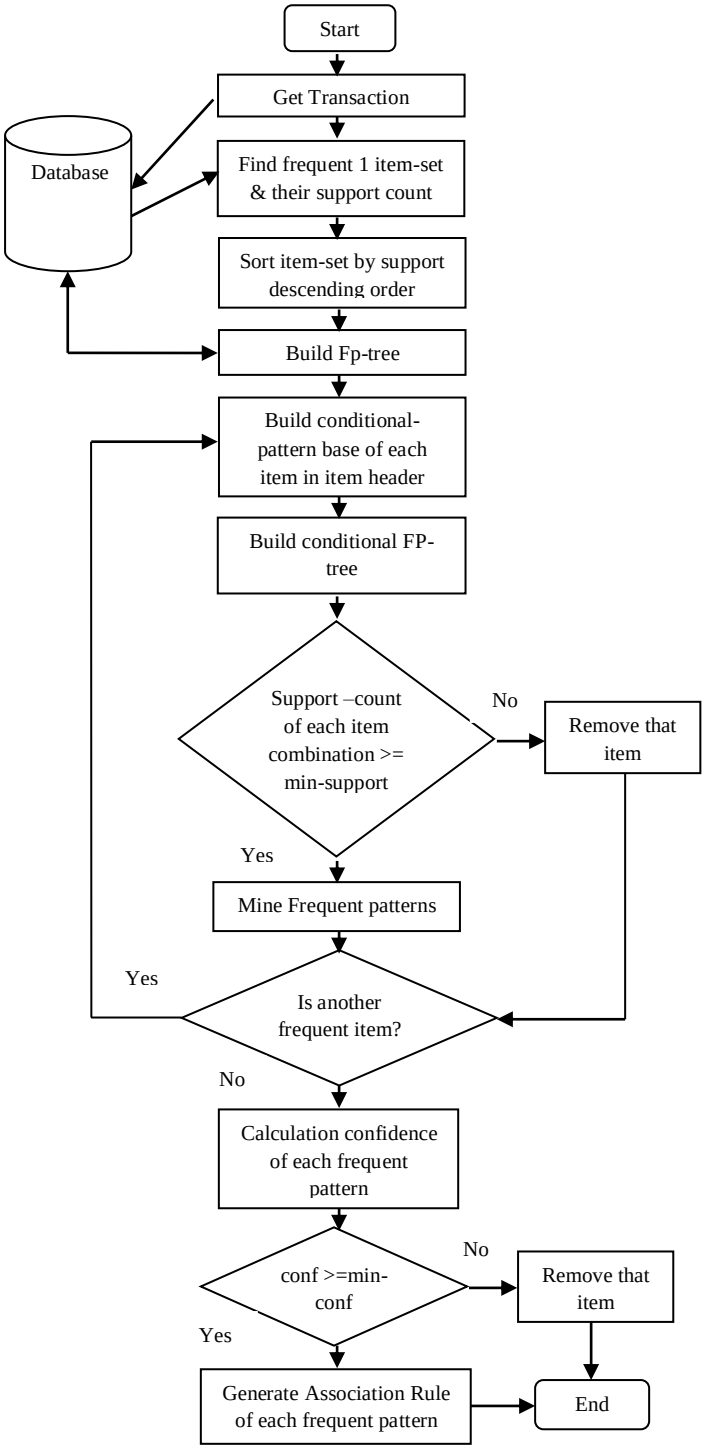

Figure1. Overview of the proposed System

Finally, generates association rules with user desire. Business manager (or) stationer by using this results can make decisions sales promotion, catalog design and grouping of items in different aisles in order to increase the sale and profit.

4.1 Database Design

This proposed system uses the following tables in database.

(1) **Brand table** : Brand table stores all brand description used in the system.

(2) **Category tables** : Category table stores all category description used in the system.

(3) **Item table** : Item table stores all the items information used in the system.

(4) **Stock table** : Stock table stores the amount of all items' quantity used in the system.

(5) **Voucher table** : Voucher table stores all sales transaction record of the system.

(6) **Sales table** : Sales table stores all the detail description of sales transaction records in the system.

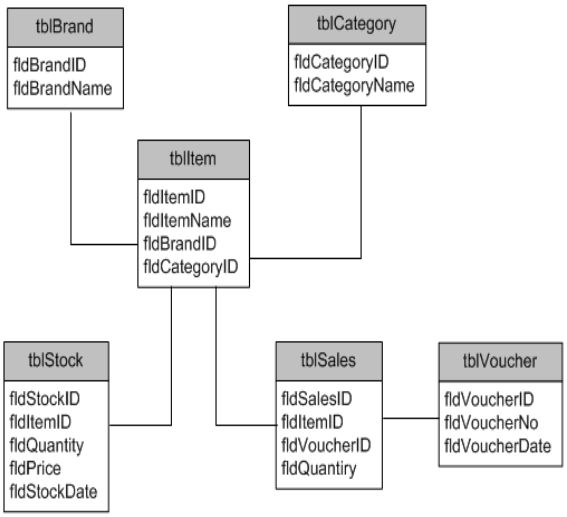

Figure.2. Data Model of the System

5. Implementation of the System

This proposed system, using Fp-growth algorithm to get the frequent patterns. It is process in the following steps:

Table 1. Transaction Database and support count=2

TID	Transaction
1	book, pen, pencil
2	ruler, pen, eraser
3	book, ruler, pen, eraser
4	ruler, eraser

Firstly, scans transaction from transaction database (Table 1). Find frequent 1-itemsets and their support and remove frequent item when support count less than minimum support, shown in Table 2.

Table 2. Frequent 1-itemsets and their support count

Item	Sup-Count
pen	3
ruler	3
eraser	3
book	2

Sort each transaction descending order according to their support count as shown in Table 3.

Table 3. Transactions with descending order

TID	Descending Order
1	pen, book
2	ruler, pen, eraser
3	ruler ,pen, eraser, book
4	ruler, eraser

And then, built FP-tree shown in Figure 3.

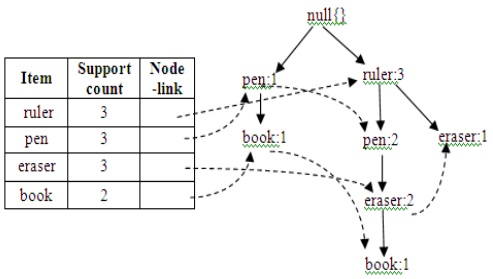

Figure 3. FP-tree

From this fp-tree, we can generate frequent pattern in the following way:

Table 4. Mining of frequent pattern based on fp-tree in Figure 3.

Item	Conditional Pattern Bases	Conditional FP-tree	Frequent Pattern
book	{(pen:1), (ruler:1,pen:1, eraser:1)}	(pen:2)	pen book:2
eraser	{(ruler:2,pen:2),(ruler:1)}	(ruler:3,pen:2)	ruler eraser:3,pen eraser:2,ruler pen eraser:2
pen	{(ruler:2)}	(ruler:2)	ruler pen:2
ruler	Φ	Φ	Φ

From this Table 4. we get largest itemset and then calculate confidence for each frequent itemset and generate association rules.

confidence({ruler, pen}→ {eraser}) = 2/3 * 100
= 66.67%

confidence({ruler, eraser}→ {pen}) = 2/3 * 100
= 66.67%

confidence({eraser, pen}→ {ruler}) = 2/2 * 100
= 100%

confidence({ruler}→ {pen, eraser}) = 2/3 * 100
= 66.67%

confidence({pen}→ { ruler, eraser}) = 2/3 * 100
= 66.67%

confidence({eraser}→ { pen, ruler}) = 2/3 * 100
= 66.67%

Finally, Stationer (or) managers can be use these rule if desire in many cases such as retail sales promotion and identifying market opportunies and product positioning based on user buying habits information. Therefore, the stationery shop is increase sales and profits.

6. Comparative Study

The two frequent pattern mining algorithm were implemented in Java , tested on same data set and the same kinds of computer and SQL server are used. Any transaction may contain more than one frequent itemset. The number of items in a transaction may vary. Also the number of items in a itemset is variable. The generated data sets depend on the number if items in a transaction, number of items in a frequent itemset, etc.

The data set is generated for a number of items N=100 and a maximum number of frequent itemsets |L| = 3000. |T| was chosen to be 10,where N= Number of item, |L|= Number of maximal potentially large itemsets and |T|= Average size of the transactions. Some of the results of the comparison between the Apriori and Fp-growth and support factor is 5%, are shown in Table 5.

Table 5. The result for support factor of 5%

Transactions(K)	Execution Time (sec)	
	Apriori	FP-growth
10	13.94	3.76
30	48.37	14.63
50	107.65	34.30
110	1471.40	95.50
190	5320.60	273.60
400	17259.20	849.70

Table 5 shows that the execution time of the algorithms. The best performance (or) can find frequent items in short time algorithm is FP-growth algorithm. Because it breaks the main bottleneck of the Apriori i.e. the frequent itemsets are generated with only two passes over the database and without any other candidate generation process. Therefore, Fp-growth algorithm is best performance rather than the Apriori algorithm.

7. Conclusion

In this system, the association rule mining is very useful for finding of items association, forecasting the sales and arranging the shelf space. By using this system, user or managers can get the desire results such as association of items, association of brand, association of category and executive summaries of sales, in short time. The results can help users to make better decision and can apply as user desire. This system reduce the time rather than Apriori algorithm. Therefore, it save time and then manager can know how to arrange and how to change the shelf space in future for customer, who can get more convenience in choosing item to increase the sale and profit.

References

[1] Ashoka Savasere, Edward Omiencinski, and Shamkant B. Navathe. “An Efficient Algorithm for Mining Association Rules in Large Databases.” In Proceedings of the 21st International Conference on Very Large Databases, pag. 432 - 444, 1995.

[2] Florian Verhein fverhein@it. usyd. edu.au School of Information Technologies, The

University of Sydney, Australia."Frequent Pattern Growth Algorithm"

[3] J.Han, M. kamber, "Data Mining Concept and Techniques", Morgan Kaufmann Publishers, San Francisco, USA, 2001,ISBN-1558604898.

[4] Sotiris Kotsiantis, Dimitris Kanellopoulos. "Association Rule Mining:A Recent Overview"

[5] Shang Xuequn, Sattler Kai-Uwe, and Geist Ingolf. "SQL Based Frequent Pattern Mining with FP-growth"