Nandini Das[1], Avishek Ghosh[2] Prasun Das[1]

1)SQC & OR Division
Indian Statistical Institute
203, B. T. Road
Kolkata 700 108, West Bengal, India
email: nandini@isical.ac.in
dasprasun@rediffmail.com,

2) M. Stat Student, Indian Statistical Institute
avishekghosh@gmail.com

MINING ASSOCIATION RULES TO EVALUATE CONSUMER PERCEPTION: A NEW FP-TREE APPROACH

Abstract: *Association rule mining finds interesting relationships among large set of data items. While finding the important (or, frequent) relations from the set of consumer survey data, a modified algorithm based on frequent pattern growth is developed in this work. The sensitivity of support and confidence used for rule mining on the data is tested. The interaction between the order of the attributes and the confidence used is observed in terms of the number of rules mined. The impact of the product features on the level of consumer perception is thoroughly studied.*

Keywords: *Data mining, Association rule mining, Itemset, Frequent pattern tree, Support, Confidence, Order, Consumer satisfaction*

1. INTRODUCTION

1.1 Knowledge Discovery using Association Rule Mining

Data mining, popularly known as knowledge discovery and databases, refers to mining patterns, trends and correlations representing knowledge implicitly stored in large database (**Han and Kamber, 2006**). These information can help industry management in many business decision making process. Association Rule Mining (ARM) is one of such data mining techniques which tries to finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories. In past years and recently too, literatures on association rule mining focuses on developing efficient algorithms for finding frequent patterns and association rules **(Agrawal et al., 1993; Agrawal and Srikant, 1994; Han et al., 2000; Liu et al., 2003; Gouda and Zaki, 2005; Hahsler and Hornik, 2007)**. Applications of ARM in the area of basket data analysis, cross-marketing, catalog design, telecommunication networks, market and risk management, medical research, inventory control, website navigation, clustering and classification are noteworthy. As the name suggests, given the relational or transactional database of information, ARM is used to find all rules that correlate the presence of one set of items with that of another set of items. For example, 88% of people who purchase tires and auto accessories also get automotive services done. It is a two step procedure: (1) *find all frequent itemsets*; and (2) *generate strong association rules from frequent itemsets*. Two important algorithms are followed for finding the frequent itemsets: (a) Apriori Algorithm: Mining frequent itemsets through candidate generation, and (b) Mining frequent itemsets without candidate generation. *Section 2* gives a brief

overview about the methodology of these two algorithms.

1.2 Retail Business, Consumer Perception and Customer Satisfaction

In India, after liberalization and globalization took place a few years back, the companies began to face a tough challenge to retain their market share due to increase in number of competitors venturing into the vast Indian market. Further, the concept of manufacturing has began to be slowly transformed into a service operation for retail businesses. This situation has stimulated consumer expectations to change rapidly over time. Process improvements, advent of new technology, changes in consumer's priorities, improved quality of service provided by competitors is just a few. It has, therefore, become important to periodically update the knowledge of consumer expectations and transform the organization to a customer-focused organization. With better understanding of consumers' perception about product quality and service, a company can identify its relative strengths and weakness and chart a path for future progress and improvement.

Customer satisfaction is consumers' perception about a product or service that a supplier has met or exceeded their expectations. With the increase in volume and complexity of retail business, the need for further improvement has become steeper and faster too. In matters of product quality offered by several companies, there exists a strong link among consumer perception, their satisfaction level and retention for the company. **Kleinsorge, Schary and Ray (1992)** used data envelopment analysis modeling to incorporate quantitative measures of customer satisfaction. **Aysar and William (1997)** showed that the orientation towards both quality and competitiveness for a company improves its performance in the areas of business growth and customer satisfaction. Many retail operations are trying to establish their quality through integrated retail chains and reviewing the performance from the perspective of end point customers by adopting on-line and/or off-line surveys. **Wang et al. (1999)** used fuzzy logic operations for developing multicustomer due-date bargaining tool to allocate the resource and determine the order completion times, following the priority sequence of orders. **Shimazu (2001)** analyzes conversation models of human sales clerks to effectively match a customer's buying points and product selling points. Fuzzy if-then rules are used to map the human expectation as an input and human need and budget for the particular commodity as an output on the basis of combination of the two navigation models. **Iglesias (2004)**, through a retail banking sector study, established a strong effect of preconceptions about the service category on the perceptions of quality during service. **Petrick (2004)** assessed cruise passengers' behavioral intentions based on the satisfaction model, perceived value model and quality model. In another work, **Kuofie and Qaddour (2005)** proposed a total customer satisfaction methodology, both for internal and external customers. **Homburg (2005)** showed that repurchase intentions of customer are influenced by the magnitude of the price increased and the perceived fairness of motive for the increase of price. Implementation of a customer relationship management (CRM) model based on a customer satisfaction survey has been immensely successful for profitable customer segmentation **(Jang and Sang, 2005; Roh et al., 2005)**. **Tong and Lu (2009)** find that the band of ARM and CRM can help enterprises to find meaningful rules or patterns, which provide the basis for enterprises to make decision to be in a better competitive

position. **Das and Gauri (2006)** demonstrated the importance for a footwear company to know its consumers' perceptions about the product features and service it provides. **Das and Mukherjee (2008)** aimed at finding the relative brand position of an organization along with the nature of preference as perceived by the customers using cluster and principal component analysis. In another work **Das (2009)**, an attempt has been made to model the customer choice in FMCG product design during purchase in retail outlets, using fuzzy approximate reasoning and rule generation technique, based on customer survey. Very recently, **Chen and Chen (2010)** examine the visitor experience of heritage tourism and investigate the survey based relationships between the quality of those experiences, perceived value, satisfaction, and behavioral intentions, using structural equation modeling technique.

1.3 Study Objective

The concerned tobacco company has its retail chain business spread out in almost every parts of India. The management of the company has taken a policy to strengthen its retail business through identification of critical tobacco features and service related to business at retail outlets and improve upon them. Since the best way to understand the consumer's need is to know it directly from them, it was decided to institute a consumer feedback survey system. An attempt has, thereafter, been taken in this work to mine the customers' perceptions using association rule mining technique. Rules for preferring tobacco attributes by the customers, prevailing primarily at retail outlets, are evaluated with the help of generation of frequent item-sets.

The study is organized as follows. *Section 2* talks about the preliminaries of association rule mining. The practical problem considered for ARM is described in *Section 3*. *Section 4* and *Section 5* elaborate about the proposed FP-tree and its corresponding ARM algorithm, including implementation scheme. *Section 6* displays all the findings of the study and the relevant discussions. The proper choice of *order is* explained in *Section 7*. The study is concluded with *Section 8*.

2. PRELIMINARIES OF ASSOCIATION RULE MINING

2.1 Formalization

Let $J = \{i_1, i_2, \ldots, i_m\}$ be a set of items. D is the task relevant data, which is also a set of database transactions. A transaction T is a set of items such that $T \subseteq J$. A transaction T is said to contain A, a set of items, *iff* $A \subseteq T$.

An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset J$, $Y \subset J$, and $X \cap Y = \Phi$.

There are two important basic measures for association rules, *support* and *confidence*. Since, from a large database, users concern about only those frequently purchased items, usually thresholds of support and confidence are predefined by users to prune rules that are not so useful. The two thresholds are called minimal support (*min_sup*) and minimal confidence (*min_conf*) respectively. *Support* of an association rule is defined as the percentage of data points that contain $X \cup Y$ to the total number of records in the database. Suppose, the *support* of an item is 0.2%, it means only 0.2 percent of the transaction contain purchasing of this item. *Confidence* of an association rule is defined as the percentage of the number of transactions that contain $X \cup Y$ to the total number of data points that contain X. *Confidence* is a measure of strength of the association rules. Suppose the *confidence* of the association rule $X \Rightarrow Y$ is 80%, it means that 80% of the transactions that

contain X also contain Y together. In probability terms, assuming *support* and *confidence* values occur between 0 and 1,

$$\text{support } (X \Rightarrow Y) = P(X \cup Y)$$
$$\text{confidence } (X \Rightarrow Y) = P(Y|X)$$

The rules which satisfy *min_sup* and also *min_conf* are called **strong**.

An *itemset* is a set of items. The *frequency* of an itemset is the number of transactions containing the itemset. This is also called the *support count* of the itemset. If an itemset satisfies *min_sup*, it is called *frequent*. A subset of a frequent itemset must also be a frequent itemset. Frequent itemsets are iteratively found with cardinality from 1 to k (k-itemset). Then, these itemsets are used to generate association rules.

2.2 Apriori Algorithm: frequent itemsets through candidate generation

Apriori is the most well-known algorithm for mining Boolean association rules (**Heglan, 2003**). As the name suggests, it uses prior knowledge of frequent itemset properties. Apriori is an iterative algorithm in which k-itemsets are used to find (k+1)-itemsets. To improve the search procedure we use the **Apriori** property, which says that 'all nonempty subset of a frequent itemset have to be frequent '. Apriori uses pruning techniques to avoid measuring certain itemsets, while guaranteeing completeness. The algorithm works as follows:

1. In the first iteration, each item is a member of the set of candidate 1-itemset C_1. Transactions are scanned to find out the *support count* of each of the items.
2. The set of *frequent* 1-itemset L_1 is found out by listing all the members of C_1 satisfying *min_sup*.
3. The algorithm uses $C_1 \triangleright\triangleleft C_1$ to generate a candidate set of 2-frequent itemset C_2. The support counts of each of the itemsets in C_2 are calculated.
4. The set of frequent 2-itemsets is constructed by those 2-itemsets of C_2 satisfying *min_sup*.
5. At k^{th} stage C_k is generated by $L_{k-1} \triangleright\triangleleft L_{k-1}$. The support count of each of the itemsets is found out. Then L_k is constructed using the k-itemsets satisfying *min_sup*.
6. The algorithm terminates at l^{th} step where $L_l = \Phi$.

After finding the frequent itemsets, the following algorithm is used to generate association rules from them.

1. For each frequent itemset A, generate all nonempty subsets of A.
2. For every nonempty subset $B \subset A$, generate the rule $A \Rightarrow (A-B)$ if $\{support\_count(A) / support\_count(B)\} \geq min\_conf$.

Many new algorithms of ARM have been designed, based on the Apriori algorithm, with some modifications.

2.3 Mining frequent itemsets without candidate generation

The Apriori algorithm, however, suffers from the following drawbacks:

- It may lead to generation of huge number of candidate sets using lot of time and memory (*complex candidate generation*).
- It may need to scan the database a number of times (*multiple scanning*) and check a large set of candidates.

Frequent pattern growth, or simply **FP-growth**, is another algorithm which compresses the database representing frequent items in a **frequent pattern tree** (FP-tree**)** but retains the itemset association information **(Han and Pei, 2000)**. This solves the major constraints of the Apriori algorithm where the frequent itemsets are completely generated with only two passes over the database and without any candidate generation process.

The frequent patterns generation process includes two sub processes: constructing the FP-tree, and generating frequent patterns from the FP-tree. The mining result is the same as Apriori algorithm. Let us understand the algorithm through an example.

Let D_T be the transaction database (*ref.* **Table 1**).

Table 1. *Transactions in* D_T

Transaction-ID	Item-ID's
T1	A,B,E
T2	B,D
T3	B,C
T4	A,B,D
T5	A,C
T6	B,C
T7	A,C
T8	A,B,C,E
T9	A,B,C

The first scan of the database is the same as Apriori, which derives the frequent 1-itemset and their support count. The set is then sorted in the descending order of support count and the following resulting list is obtained.

L = [B:7, A:6, C:6, D:2, E:2]

An FP-tree is then constructed as follows. First, create a root of the tree labeled with {*null*}. Scan the database again. The items in each transaction are processed in *L* order and a branch is created for each transaction. We begin with {T1: A,B,E} which contains the items {B,A,E} in *L* order. So we construct the first branch of the tree:

{null}→ {B:1}→ {A:1}→ {E:1}

The second transaction is {T2: B,D} containing {B,D} in *L* order. So a {B:1} → {D:1} link is created and {B:1} is modified to {B:2}. So, whenever a new transaction is inserted in the tree, the counts in all the nodes along the path of the transaction are incremented by 1. In this way the FP-tree is constructed (*ref.* **Figure 1**).

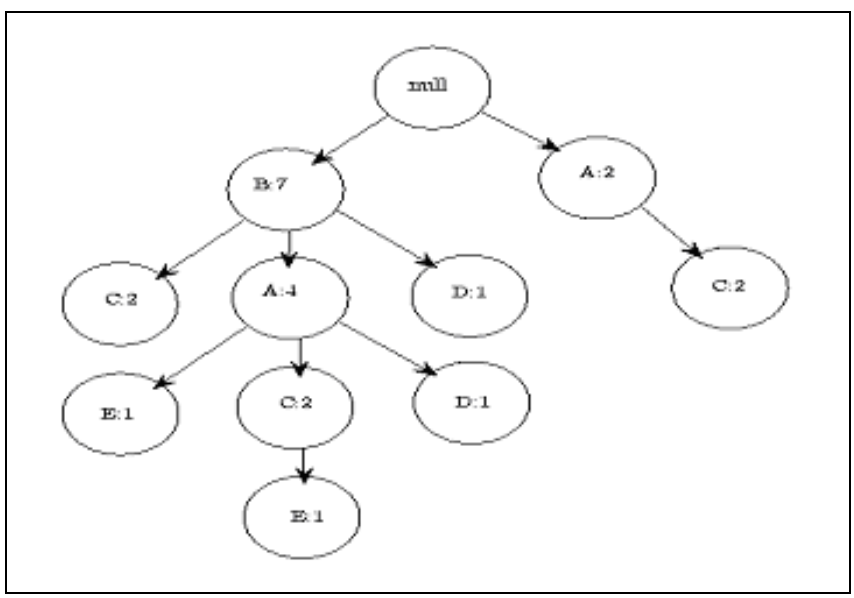

Fig. 1: An example of FP-tree

The mining of the FP-tree proceeds as follows. We start from each frequent length-1 pattern (an initial *suffix pattern*), construct the *conditional pattern base* (a

sub database consisting of a set of prefix paths co-occurring with the suffix pattern) and its (*conditional*) FP-tree, and finally perform mining recursively on such trees using some pre-specified level of minimum support.

In a nutshell, the efficiency of FP-tree algorithm can be summarized in three steps. First, it is a compressed representation of the original database because only those frequent items are used to construct the tree. Secondly, this algorithm only scans the database twice. Thirdly, FP-tree uses a divide and conquer method that considerably reduces the size of the subsequent conditional FP-tree. Mining on the FP-tree for this particular example (*ref.* **Table 1**) is given in **Table 2**.

Table 2. Mining on the FP-tree in example transaction database D_T

Item	Conditional pattern base	Conditional FP-tree	Frequent patterns generated
E	{(BA:1),(BAC:1)}	⟨ B:2, A:2 ⟩	BE:2, AE:2, BAE:2
D	{(BA:1),(B:1)}	⟨ B:2 ⟩	BD:2
C	{(BA:2),(B:2),(A:2)}	⟨ B:4, A:2 ⟩ , ⟨ A:2 ⟩	BC:4, AC:2, BAC:2
A	{(B:4)}	⟨ B:4 ⟩	BA:4

3. AN INDUSTRIAL CASE: RETAIL BUSINESS ENVIRONMENT

The socio-economic conditions and culture of the people in different parts of India are different and accordingly the needs of the consumers differ. One of the important issues under the consumer feedback survey is to know the choice of product (in this case, cigarettes) features like *brand*, *price*, and *quality* along with *service level* in the retail outlets. Accordingly, one of the key objectives of this survey has been decided to determine the optimal rules for classifying the consumers' choice based on the following topics of interest.

1. perception about *Brand* (Excellent, Moderate, Traditional)
2. perception about *Price* (Economic, Value for money, Expensive)
3. perception about *Quality* (Satisfactory, Average, Unfit for use)
4. perception about *Service level* (Strong, Moderate, Weak)
5. Overall consumer's rating about the brand (Excellent, Good, Average, Poor)

Since it was decided to collect the information from consumers through personal interview using predefined questionnaire, the data are collected from those persons who have used the product of a particular tobacco company under study. At the time of interview, the consumers have been informed about the purpose of the study. During the period of survey, responses from 554 consumers could be obtained out of which 53 filled-up questionnaires could not be analyzed, after proper scrutiny, due to lack of information.

4. MODIFIED FP-TREE AND ITS IMPLEMENTATION

For this particular database, a modified version of the FP-tree algorithm is developed and implemented to mine some frequent itemsets. The steps apparently do not support association rule mining as because there is no itemset. From each of the respondent (consumer),

we have observations on four attributes, namely *Brand* (*BT*), *Price* (*PR*), *Quality* (*QT*), *Service* (*SE*). Each of these features can have three possible categories and hence, coded as {1,2,3}. So, in total, there are 81 ($=3^4$) possible 4-tuples, which can occur in the database. Moreover, the *CSI (Customer Satisfaction Index)* value for each of those 4-tuples is available. From this database, we want to choose some frequent 4-tuples which appear to be significant. The modified FP-tree conceived, using the data, is now described below.

Consider an order of the attributes; say {*BT, PR, QT, SE*}. For each value of *BT,* there are 3 possible values of *PR*. For each combination of {*BT,PR*} there are 3 possible values of *QT*. And again for each combination of {*BT,PR,QT*} there are 3 possible values of *SE*. Let us consider a tree having {*null*} in its topmost node. In the first stage, the topmost node has 3 children corresponding to 3 levels of *BT*. Each of them has 3 children corresponding to 3 levels of *PR* in the second stage. Then, there is a third stage of *QT* levels and a fourth stage of *SE* levels constructed similarly. So, there are now 81 leaf nodes at the fourth or *SE* stage corresponding to 81 possible 4-tuples. This 4-stage architecture looks like Figure 2. At each node, there is a counter set to 0 initially. Let us now feed our data in this 4-stage architecture.

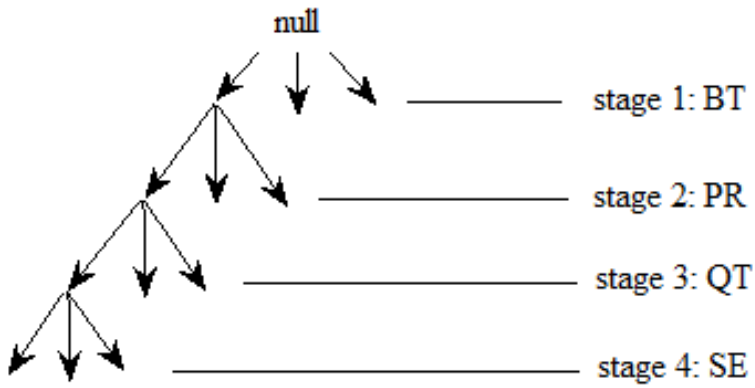

Fig. 2: Modified FP-tree

For a given observation of 4 attributes, we first order them as needed and then traverse the tree along its path and increase the counters of each node in the path by 1. Once we do it for every data point, the tree is then experienced with the modified database.

Let us interpret the counter at each node of each stage. The counter at the i^{th} node of the first stage keeps track of the number of data points having *BT* value *i*, *i* = 1,2,3. In the second stage, the j^{th} node under the i^{th} parent keeps track of the number of data points having *BT* = *i and PR* = *j, j* = 1,2,3. Similarly, for the third and the fourth stage. So the counters at the 81 leaf nodes give the number of occurrences of the 81 different 4-tuples.

Now, let us define *support* and *confidence* for this case. To say a rule to be valid, it must be frequent. We define the *support* of a rule (or, an observed 4-tuple of {*BT, PR, QT, SE*}) as the proportion of data points supporting the rule. In other words, it is the ratio of counts at the leaf nodes of the 4^{th} stage to the total number

of data points, i.e.,

$$support([i,j,k,l]) = P(BT = i \cap PR = j \cap QT = k \cap SE = l)$$

Now, we define *confidence* as the ratio of the count at a node to the count of its immediate parent node. So, in terms of probability, it is the conditional probability defined by:

$$confidence([i,j]) = P(PR = j|BT = i)$$

or,

$$confidence([i,j,k]) = P(QT = k|BT = i, PR = j)$$

or,

$$confidence([i,j,k,l]) = P(SE = l|BT = i, PR = j, QT = k)$$

Then, rules, satisfying for some values of *minimum support* and *minimum confidence,* can be generated and selected.

5. PROPOSED ARM ALGORITHM

Based on the modified FP-tree structure designed for this problem, the algorithm for ARM is proposed as follows.

a) Consider an order of {*BT, PR, QT, SE*}.
b) Construct the FP-tree and feed the database into the tree.
c) At each stage, from 2nd stage onwards, check for the *minimum confidence*. If a node fails to satisfy *minimum confidence*, delete the entire subtree below it including that node itself.
d) From the remaining nodes at the last stage, take those for which the counts are more then *minimum support*.
e) Backtrack their paths to get the corresponding rules.

An integer array of length 120 is used in MATLAB code to construct the tree where, the indices are related by this simple rule:

If i is the index of a node then the indices of its children are given by (3*i*+1), (3*i*+2), (3*i*+3).

The indices 40-120 are identified for 81 leaf nodes. While retracing a path from a leaf node (needed for getting the associated rules), the following relation is used:

The index of the immediate parent of a node indexed by j is given by $\left[\frac{j-1}{3}\right]$, where [.] is the well-known box function.

The input arguments of the function are $0 \le min\_sup \le 1$, $0 \le min\_conf \le 1$. The *order* is a vector of length 4 consisting of some permutations of [1,2,3,4], which specifies the order of the four attributes to be considered while constructing the FP-tree where *BT*=1, *PR*=2, *QT*=3 and *SE*=4. The function returns the mined rules along with their corresponding *CSI* values. The output rules are always given in the natural order i.e. [*BT PR QT SE*].

6. RESULTS AND DISCUSSIONS

The results are presented for varying choices of *min_sup*, *min_conf* and *order*. The effect of choice of *order*, while mining rules, is further explained in the next section. However, some judicious values of *min_sup* and *min_conf* are selected as follows. Since, the database consists of preferences of 501 consumers and 81 possible rules, so, under ideal conditions, each rule would occur approximately 6 times. An important rule has to occur more frequently than that. So, 2% i.e. 0.02 can be a good choice of minimum support. For each parent node, there are 3 possible values of children. So a logical minimum confidence may be 20%-30%. Although one may take more or less than that according to the strictness of requirements and heterogeneity of the data. Few combinations of (*min_sup*, *min_conf*)

and the rules mined thereof are presented in **Table 3**, **4** & **5** respectively (see **ANNEXURE**).

- *(min_sup, min_conf)* = (0.02, 0.2): *ref.* **Table 3**
- *(min_sup, min_conf)* = (0.02, 0.3): *ref.* **Table 4**
- *(min_sup, min_conf)* = (0.03, 0.2): *ref.* **Table 5**

It is observed from the results of rule mining that for a fixed *order* of the attributes, the number of rules mined becomes less as *minimum support* or *minimum confidence* increases. But the set of rules mined with higher support or confidence is a subset of the set of rules mined using lower support or confidence. The reason is quite easy to understand and the fact is evident from the results too.

Now, considering the effect of choice of *minimum support*, it is observed that when 2% minimum support is used, 8-10 rules are mined on an average. The average number of rules comes down to 5-7 when 3% minimum support is used. So, the database considered for this work is very much sensitive towards the *minimum support*. Also, there are only a few rules, which are really predominant. These rules have relatively higher support count as compared to the other rules.

The choice of *minimum confidence* also plays an important role in deciding the frequent rules according to our proposed algorithm. At 2% *minimum support* level, when 20% *minimum confidence* is used, we get 8-10 rules on an average. But, if *minimum confidence* increases to 30%, the average number of mined rules comes down to 5-8. Another important observation is that, when higher *minimum confidence* is used the choice of *order* becomes important. In particular, when *order* is changed at a higher *minimum confidence*, much fluctuation in the number of mined rules is observed.

Next, looking at the values of the *CSI* for the mined rules, it is found that in most of the cases, rules with *CSI* values 2 or 3 are mined. Rules having *CSI* value 1 are mined rarely and those having *CSI* values 4 are never mined.

There is some effect of the *order* in mining the rules. Though, some very frequent rules always got mined irrespective of the order (e.g. [1 1 1 1], [2 2 2 2] etc.), but change in *order* induces change in other mined rules. Sometimes, few rules fail to get mined and on the other hand some new rules get mined due to the change in *order*. The results obtained show the changes due to *order*. One interesting observation is that the orders [*BT PR QT SE*] and [*BT QT PR SE*] result in same set of mined rules in two cases (*ref.* **Table 3** and **5**) and only one replacement in the other case (see **Table 4**). A possible explanation is provided heuristically for such occurrence. In general, it can be said that the exchange of *PR* and *QT* during the construction of the FP-tree does not affect the mining much. In other words, *price* and *quality* have *equal precedence* in the construction of the FP-tree.

If higher *minimum support* level or *minimum confidence* level is used, the number of mined rules comes down drastically to 2-4. Moreover, almost the same set of rules gets mined for different choices of *order*. Two of them are [1 1 1 1] and [2 2 2 2] which cannot provide knowledge to understand the true nature of the data. So, these results are not presented here.

7. CHOICE OF *ORDER*

In the previous section, it is observed that the *order* of the attributes, while constructing the tree, is very important for mining frequent rules. The question is, what should be a logical choice of *order* for a problem given. A technique is proposed here, which can explain some phenomena observed in the results. But there may be some other choices too.

Recall that in *Section 2.3*, while constructing the FP-tree, the attributes are

ordered by descending number of support counts. From another point of view, we may say that the attributes are ordered in descending amount of interest as attributes having more count is of more interest and has more effect in the rule. Similar things can be done here too. The count cannot be the distinguishing factor here. Since, *CSI* is a function of *BT, PR, QT, SE*, we can order them in their descending amount of effect on *CSI*. This can be done using logistic regression analysis and calculating odds ratio of each attribute for finding their importance as effect, including regression coefficients of each attrubute. A descending order these effects based on the importance can be considered while constructing the FP-tree.

In the data considered for this work, the attribute *BT* is found to be the most important. Then, there are attributes of importance like *PR*, *QT*, and *SE* is the least important of all. Also, it has been found that there is not much difference between *PR* and *QT*. So a good choice of the *order* is [1 2 3 4] or [1 3 2 4]. The mined rules for these two choices of the *order* are almost the same whatever be the *min_sup* or *min_conf* as mentioned in *Section 6*.

8. CONCLUSION

The association rule mining technique applied here reveals the structure of the data and gives the idea of consumers' choice based on the product (cigarettes) features like *brand*, *price* and *quality* along with *service level*. A modified version of the existing FP-tree based ARM algorithm is developed for the survey database containing 501 sets of consumers' opinion (preference) about the product, comprising of 81 possible rules. Since, the database is heterogeneous and skewed towards very few rules, at most 10 or 11 of them are mined using 2%-3% *minimum support* and 20%-30% *minimum confidence*. The data is observed to be very much sensitive towards the *minimum support* and *confidence level*s. The choice of ordering the attributes becomes important in terms of number of rules mined when higher minimum confidence is used. However, exchanging the two attributes, namely, *price* and *quality*, during construction of the FP-tree seems to be indifferent for rule mining. The impact of the product features on the level of consumer perception is thoroughly studied. Two rules, namely, [1 1 1 1] and [2 2 2 2] are found to be most frequent for different choices of ordering. So, there may be a tendency of *'average marking'* against the data. The category 3 is extremely rare for any of the product features. So, consumers are not interested to mark 'Traditional' for *Brand*, 'Expensive' for *Price*, 'Unfit for use' for *Quality* or 'Weak' for *Service level*. In most of the cases, the rules for the level of consumer perception mined are 2 ('Good') or 3 ('Average'). Rules having perception level 1 ('Excellent') are mined rarely, and, those having 4 ('Poor') are never mined. In future, the algorithm proposed here can be fine tuned by giving some more logical choice for the *order*. It will also be very interesting to compare the efficiency and effectiveness of the proposed algorithm with standard rule developing algorithms.

REFERENCES:

[1] Han, J. and Kamber, M. (2006), Data mining: concepts and techniques, 2nd ed. Morgan Kaufmann Publishers.

[2] Agrawal, R., Imielinski, T., and Swami, A. N. (1993), Mining association rules between sets of items in large databases, In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, 207-216.

[3] Agrawal, R. and Srikant, R. (1994), Fast algorithms for mining association rules, Proc. of the International Conf. on Very Large Databases (VLDB'94), 487-499, Santiago, Chile.
[4] Han, J. Pei, J. and Yin, Y (2000), "Mining frequent patterns without candidate generation" In Proc. 2000 ACMSIGMOD Int. Conf. Management of Data (SIGMOD'00), pages 1-12, Dallas, TX.
[5] Liu, G., Lu, H., Xu, Y. and Yu, J. X. (2003), Ascending frequency ordered prefix-tree: efficient mining of frequent patterns, Proc. 2003 Int. Conf. on Database Systems for Advanced Applications (DASFAA03), Kyoto, Japan.
[6] Gouda, K. and Zaki, M. (2005), GenMax: an efficient algorithm for mining maximal frequent itemsets, Data Mining and Knowledge Discovery: An International Journal, 11(3), 223-242.
[7] Hahsler, M. and Hornik, K. (2007), New probabilistic interest measures for association rules, Intelligent Data Analysis: an Internnationl Journal, 11(5), 437-455.
[8] Kleinsorge, Ilene K., Schary, P. B. and Ray, D. T. (1992), Data envelopment analysis for monitoring customer-supplier relationships, Journal of Accounting and Public Policy, 11(4), 357-372.
[9] Aysar, P. S., William, C. J. (1997), The impact of market/quality orientation on business performance. Computers & Industrial Engineering, 33(1–2), 161–165.
[10] Wang, D., Fang, S. and Nuttle, L. W. (1999), Soft comuting for multi-customer due-date bargaining, IEEE Transactions on Systems, Man, and Cybernetics, Part C, pp. 566-575.
[11] Shimazu, H. (2001), Expert clerk: Navigating shoppers' buying process with the combination of asking and proposing, in Nebel,B. (Ed.) Proceedings of the 17 International Joint Conference on Artificial Intelligence (IJCAI-01), pp. 1443-1448, Morgan Kaufmann, Seattle, Washington, USA.
[12] Iglesias, V. (2004), Preconceptions about service: How much do they influence quality evaluations?, Journal of Service Research, 7(1), 90–103.
[13] Petrick, J. F. (2004), The roles of quality, value and satisfaction in predicting cruise passengers' behavioural intentions, Journal of Travel Research, 42(4): 397-407.
[14] Kuofie, M. and Qaddour, J. (2005), Total customer satisfaction: using an information system to assist in total quality management, Journal of Quality, http://www.thecqi.org/journalofquality/.
[15] Homburg, C. (2005), Customers' reactions to price increases: do customer satisfaction and perceived motive fairness matter?, Journal of the Academy of Marketing Science, 33(1): 36-49.
[16] Jang, H. L. and Sang, C. P. (2005), Intelligent profitable customers' segmentation system based on business intelligence tools, Expert Systems with Applications, 29(1), 145–152.
[17] Roh, T. H., Ahn, C. K. and Han, I. (2005), The priority factor model for customer relationship management system success, Expert Systems with Applications, 28(4):641–654.
[18] Tong, H. and Lu, X. (2009), Consumption psychoanalysis and customer relationship management based on association rules mining, World Congress on Computer Science and Information Engineering, 384-388.
[19] Das, P. and Gauri, S. K. (2006), Enhancing business growth through increase in consumer focus: A case study. Journal of Quality, http: // www.iqa.org /journalofquality/latest.asp.
[20] Das, P. and Mukherjee, S. (2008), Modeling of customer preferences on product features and comparing the competitors' performances, Quality Engineering, 20(1), 53-62.
[21] Das, P. (2009), Adaptation of fuzzy reasoning and rule generation for customers' choice in retail FMCG business, Journal of Management Research, 9(1), 15-26.

[22] Chen, Ching-Fu and Chen, Fu-Shian. (2010), Experience quality, perceived value, satisfaction and behavioral intentions for heritage tourists, Tourism Management, 31(1), 29-35.
[23] Hegland, M. (2003), Algorithms for Association Rules, Lecture Notes in Computer Science, 2600, 226 – 234
[24] Han, J. and Pei, J. (2000), Mining frequent patterns by pattern-growth: methodology and Implications, ACM SIGKDD Explorations Newsletter, 2(2), 14-20.

Annexure

Table 3. Mined rules for different *orders* and (*min_sup* , *min_conf*) = (0.02,0.2)

Order	BT	PR	QT	SE	CSI	Order	BT	PR	QT	SE	CSI
BT PR QT SE	1	1	1	1	2	PR BT QT SE	1	1	1	1	2
	1	1	2	1	3		1	1	2	1	3
	1	1	2	2	3		1	1	2	2	3
	1	2	1	1	1		1	2	1	1	1
	1	2	2	2	3		1	2	2	2	3
	2	1	1	1	3		1	3	1	1	2
	2	1	2	2	3		2	1	1	1	3
	2	2	1	1	2		2	1	2	2	3
	2	2	2	1	3		2	2	1	1	2
	2	2	2	2	3		2	2	2	1	3
							2	2	2	2	3
BT PR SE QT	1	1	1	1	2	PR BT SE QT	1	1	1	1	2
	1	1	2	2	3		1	1	2	2	3
	1	2	1	1	1		1	2	1	1	1
	1	2	1	2	3		1	2	1	2	3
	1	2	2	2	3		1	2	2	2	3
	2	1	1	1	3		1	3	1	1	2
	2	1	2	2	3		2	1	1	1	3
	2	2	1	1	2		2	1	2	2	3
	2	2	2	1	3		2	2	1	1	2
	2	2	2	2	3		2	2	2	1	3
							2	2	2	2	3
BT QT PR SE	1	1	1	1	2	PR QT BT SE	1	1	1	1	2
	1	1	2	1	3		1	1	2	1	3
	1	1	2	2	3		1	1	2	2	3
	1	2	1	1	1		1	2	1	1	1
	1	2	2	2	3		1	2	2	2	3
	2	1	1	1	3		1	3	1	1	2
	2	1	2	2	3		2	1	1	1	3
	2	2	1	1	2		2	1	2	2	3
	2	2	2	1	3		2	2	1	1	2
	2	2	2	2	3		2	2	2	1	3
							2	2	2	2	3
BT QT SE PR	1	1	1	1	2	PR QT SE BT	1	1	1	1	2
	1	1	2	1	3		1	1	2	1	3
	1	1	2	2	3		1	1	2	2	3
	1	2	1	1	1		1	2	1	1	1
	1	2	2	2	3		1	2	2	2	3
	2	1	1	1	3		1	3	1	1	2
	2	1	2	2	3		2	1	2	2	3
	2	2	1	1	2		2	2	1	1	2
	2	2	2	1	3		2	2	2	1	3
	2	2	2	2	3		2	2	2	2	3
BT SE PR QT	1	1	1	1	2	PR SE BT QT	1	1	1	1	2
	1	1	2	2	3		1	1	2	2	3
	1	2	1	1	1		1	2	1	1	2
	1	2	1	2	3		1	2	1	2	3
	1	2	2	2	3		1	2	2	2	3
	2	1	1	1	3		1	3	1	1	2
	2	1	2	2	3		2	1	1	1	3
	2	2	1	1	2		2	1	2	2	3
	2	2	2	1	3		2	2	1	1	2
	2	2	2	2	3		2	2	2	1	3
							2	2	2	2	3
BT SE QT PR	1	1	1	1	2	PR SE QT BT	1	1	1	1	2
	1	1	2	2	3		1	1	2	1	3
	1	2	1	1	1		1	1	2	2	3
	1	2	1	2	3		1	2	1	1	1
	1	2	2	2	3		1	2	2	2	3
	2	1	1	1	3		1	3	1	1	2
	2	1	2	2	3		2	1	2	2	3
	2	2	1	1	2		2	2	1	1	2
	2	2	2	1	3		2	2	2	2	3
	2	2	2	2	3						

Table 3. contd….

Order	BT	PR	QT	SE	CSI	Order	BT	PR	QT	SE	CSI
QT BT PR SE	1	1	1	1	2	SE BT PR QT	1	1	1	1	2
	1	1	2	1	3		1	1	2	2	3
	1	1	2	2	3		1	2	1	1	3
	1	2	1	1	1		1	2	1	2	1
	1	2	2	2	3		1	2	2	2	3
	2	1	1	1	3		2	1	1	1	3
	2	1	2	2	3		2	1	2	2	3
	2	2	1	1	2		2	2	1	1	2
	2	2	2	1	3		2	2	2	1	3
	2	2	2	2	3		2	2	2	2	3
QT BT SE PR	1	1	1	1	2	SE BT QT PR	1	1	1	1	2
	1	1	2	1	3		1	1	2	2	3
	1	1	2	2	3		1	2	1	1	1
	1	2	1	1	1		1	2	1	2	3
	1	2	2	2	3		1	2	2	2	3
	2	1	1	1	3		2	1	1	1	3
	2	1	2	2	3		2	1	2	2	3
	2	2	1	1	2		2	2	1	1	2
	2	2	2	1	3		2	2	2	1	3
	2	2	2	2	3		2	2	2	2	3
QT PR BT SE	1	1	1	1	2	SE PR BT QT	1	1	1	1	2
	1	1	2	1	3		1	1	2	2	3
	1	1	2	2	3		1	2	1	1	1
	1	2	1	1	1		1	2	1	2	3
	1	2	2	2	3		1	2	2	2	3
	2	1	1	1	3		2	1	1	1	3
	2	1	2	2	3		2	1	2	2	3
	2	2	1	1	2		2	2	1	1	2
	2	2	2	1	3		2	2	2	1	3
	2	2	2	2	3		2	2	2	2	3
QT PR SE BT	1	1	1	1	2	SE PR QT BT	1	1	1	1	2
	1	1	2	1	3		1	1	2	1	3
	1	1	2	2	3		1	1	2	2	3
	1	2	1	1	1		1	2	1	1	1
	1	2	2	2	3		1	2	2	2	3
	2	1	2	2	3		2	1	2	2	3
	2	2	1	1	2		2	2	1	1	2
	2	2	2	1	3		2	2	2	2	3
	2	2	2	2	3						
QT SE BT PR	1	1	1	1	2	SE QT BT PR	1	1	1	1	2
	1	1	2	1	3		1	1	2	2	3
	1	1	2	2	3		1	2	1	1	1
	1	2	1	1	1		1	2	1	2	3
	1	2	2	2	3		1	2	2	2	3
	2	1	1	1	3		2	1	1	1	3
	2	1	2	2	3		2	1	2	2	3
	2	2	1	1	2		2	2	1	1	2
	2	2	2	1	3		2	2	2	2	3
	2	2	2	2	3						
QT SE PR BT	1	1	1	1	2	SE QT PR BT	1	1	1	1	2
	1	1	2	1	3		1	1	2	2	3
	1	1	2	2	3		1	2	1	1	1
	1	2	1	1	1		1	2	1	2	3
	1	2	2	2	3		1	2	2	2	3
	2	1	2	2	3		2	1	2	2	3
	2	2	1	1	2		2	2	1	1	2
	2	2	2	1	3		2	2	2	2	3
	1	2	2	2	3						

Table 4. Mined rules for different *orders* and (*min_sup , min_conf*) = (0.02,0.3)

Order	BT	PR	QT	SE	CSI	Order	BT	PR	QT	SE	CSI
BT PR QT SE	1 1 2 2 2	1 2 1 1 2	1 1 1 2 2	1 1 1 2 2	2 1 3 3 3	PR BT QT SE	1 1 1 2 2 2	1 2 3 1 1 2	1 1 1 1 2 2	1 1 1 1 2 2	2 1 2 3 3 3
BT PR SE QT	1 1 2 2 2 2 2	1 2 1 1 2 2 2	1 1 1 2 1 2 2	1 1 1 2 1 1 2	2 1 3 3 2 3 3	PR BT SE QT	1 1 1 2 2 2 2 2	1 2 3 1 1 2 2 2	1 1 1 1 2 1 2 2	1 1 1 2 1 1 2 2	2 1 2 3 3 2 3 3
BT QT PR SE	1 1 2 2 2	1 2 1 2 2	1 1 1 1 2	1 1 1 1 2	2 1 3 2 3	PR QT BT SE	1 1 1 1 1 2 2	1 1 1 2 3 1 2	1 2 2 1 1 2 2	1 1 2 1 1 2 2	2 3 3 1 2 3 3
BT QT SE PR	1 1 2 2	1 2 2 2	1 1 1 2	1 1 1 2	2 1 2 3	PR QT SE BT	1 1 1 1 1 2 2	1 1 1 2 3 1 2	1 2 2 1 1 2 2	1 1 2 1 1 2 2	2 3 3 1 2 3 3
BT SE PR QT	1 1 2 2 2 2	1 2 1 2 2 2	1 1 2 1 2 2	1 1 2 1 1 2	2 1 3 2 3 3	PR SE BT QT	1 1 1 1 1 1 2 2 2 2	1 1 2 2 2 3 1 2 2 2	1 2 1 1 2 1 2 1 2 2	1 2 1 2 2 1 2 1 1 2	2 3 1 3 3 2 3 2 3 3
BT SE QT PR	1 1 2 2 2	1 2 2 2 2	1 1 1 2 2	1 1 1 1 2	2 1 2 3 3	PR SE QT BT	1 1 1 1 2 2	1 1 2 3 1 2	1 2 1 1 2 2	1 2 1 1 2 2	2 3 1 2 3 3

Table 4. contd….

Order	BT	PR	QT	SE	CSI	Order	BT	PR	QT	SE	CSI
QT BT PR SE	1 1 2	1 2 2	1 1 2	1 1 2	2 1 3	SE BT PR QT	1 1 1 1 1 2 2	1 1 2 2 2 1 2	1 2 1 1 2 2 2	1 2 1 2 2 2 2	2 3 1 3 3 3 3
QT BT SE PR	1 1 2	1 2 2	1 1 2	1 1 2	2 1 3	SE BT QT PR	1 1 1 1 1 2	1 1 2 2 2 2	1 2 1 1 2 2	1 2 1 2 2 2	2 3 1 3 3 3
QT PR BT SE	1 1 1 1 2 2	1 1 1 2 1 2	1 2 2 1 2 2	1 1 2 1 2 2	2 3 3 1 3 3	SE PR BT QT	1 1 1 1 1 2 2 2	1 1 2 2 2 1 2 2	1 2 1 1 2 2 1 2	1 2 1 2 2 2 1 2	2 3 1 3 3 3 2 3
QT PR SE BT	1 1 1 1 2 2	1 1 1 2 1 2	1 2 2 1 2 2	1 1 2 1 2 2	2 3 3 1 3 3	SE PR QT BT	1 1 1 2 2	1 1 2 1 2	1 2 1 2 2	1 2 1 2 2	2 3 1 3 3
QT SE BT PR	1 1 2	1 2 2	1 1 2	1 1 2	2 1 3	SE QT BT PR	1 1 2	1 2 2	1 1 2	1 1 2	2 1 3
QT SE PR BT	1 1 2	1 2 2	1 1 2	1 1 2	2 1 3	SE QT PR BT	1 1 2	1 2 2	1 1 2	1 1 2	2 1 3

Table 5. Mined rules for different *orders* and (*min_sup , min_conf*) = (0.03,0.2)

Order	BT	PR	QT	SE	CSI	Order	BT	PR	QT	SE	CSI
	1	1	1	1	2		1	1	1	1	2
	1	2	1	1	1		1	2	1	1	1
	1	2	2	2	3		1	2	2	2	3
BT PR QT SE	2	1	2	2	3	PR BT QT SE	2	1	2	2	3
	2	2	1	1	2		2	2	1	1	2
	2	2	2	1	3		2	2	2	1	3
	2	2	2	2	3		2	2	2	2	3
	1	1	1	1	2		1	1	1	1	2
	1	2	1	1	1		1	2	1	1	1
	1	2	2	2	3		1	2	2	2	3
BT PR SE QT	2	1	2	2	3	PR BT SE QT	2	1	2	2	3
	2	2	1	1	2		2	2	1	1	2
	2	2	2	1	3		2	2	2	1	3
	2	2	2	2	3		2	2	2	2	3
	1	1	1	1	2		1	1	1	1	2
	1	2	1	1	1		1	2	1	1	1
	1	2	2	2	3		1	2	2	2	3
BT QT PR SE	2	1	2	2	3	PR QT BT SE	2	1	2	2	3
	2	2	1	1	2		2	2	1	1	2
	2	2	2	1	3		2	2	2	1	3
	2	2	2	2	3		2	2	2	2	3
	1	1	1	1	2		1	1	1	1	2
	1	2	1	1	1		1	2	1	1	1
	1	2	2	2	3		1	2	2	2	3
BT QT SE PR	2	1	2	2	3	PR QT SE BT	2	1	2	2	3
	2	2	1	1	2		2	2	1	1	2
	2	2	2	1	3		2	2	2	1	3
	2	2	2	2	3		2	2	2	2	3
	1	1	1	1	2		1	1	1	1	2
	1	2	1	1	1		1	2	1	1	1
	1	2	2	2	3		1	2	2	2	3
BT SE PR QT	2	1	2	2	3	PR SE BT QT	2	1	2	2	3
	2	2	1	1	2		2	2	1	1	2
	2	2	2	1	3		2	2	2	1	3
	2	2	2	2	3		2	2	2	2	3
	1	1	1	1	2		1	1	1	1	2
	1	2	1	1	1		1	2	1	1	1
	1	2	2	2	3		1	2	2	2	3
BT SE QT PR	2	1	2	2	3	PR SE QT BT	2	1	2	2	3
	2	2	1	1	2		2	2	1	1	2
	2	2	2	1	3		2	2	2	2	3
	2	2	2	2	3						

Table 5. contd….

Order	BT	PR	QT	SE	CSI	Order	BT	PR	QT	SE	CSI
QT BT PR SE	1 1 1 2 2 2 2	1 2 2 1 2 2 2	1 1 2 2 1 2 2	1 1 2 2 1 1 2	2 1 3 3 2 3 3	SE BT PR QT	1 1 1 2 2 2 2	1 2 2 1 2 2 2	1 1 2 2 1 2 2	1 1 2 2 1 1 2	2 1 3 3 2 3 3
QT BT SE PR	1 1 1 2 2 2 2	1 2 2 1 2 2 2	1 1 2 2 1 2 2	1 1 2 2 1 1 2	2 1 3 3 2 3 3	SE BT QT PR	1 1 1 2 2 2 2	1 2 2 1 2 2 2	1 1 2 2 1 2 2	1 1 2 2 1 1 2	2 1 3 3 2 3 3
QT PR BT SE	1 1 1 2 2 2 2	1 2 2 1 2 2 2	1 1 2 2 1 2 2	1 1 2 2 1 1 2	2 1 3 3 2 3 3	SE PR BT QT	1 1 1 2 2 2 2	1 2 2 1 2 2 2	1 1 2 2 1 2 2	1 1 2 2 1 1 2	2 1 3 3 2 3 3
QT PR SE BT	1 1 1 2 2 2 2	1 2 2 1 2 2 2	1 1 2 2 1 2 2	1 1 2 2 1 1 2	2 1 3 3 2 3 3	SE PR QT BT	1 1 1 2 2 2	1 2 2 1 2 2	1 1 2 2 1 2	1 1 2 2 1 2	2 1 3 3 2 3
QT SE BT PR	1 1 1 2 2 2 2	1 2 2 1 2 2 2	1 1 2 2 1 2 2	1 1 2 2 1 1 2	2 1 3 3 2 3 3	SE QT BT PR	1 1 1 2 2 2 2	1 2 2 1 2 2 2	1 1 2 2 1 2 2	1 1 2 2 1 1 2	2 1 3 3 2 3 3
QT SE PR BT	1 1 1 2 2 2 2	1 2 2 1 2 2 2	1 1 2 2 1 2 2	1 1 2 2 1 1 2	2 1 3 3 2 3 3	SE QT PR BT	1 1 1 2 2 2 2	1 2 2 1 2 2 2	1 1 2 2 1 2 2	1 1 2 2 1 1 2	2 1 3 3 2 3 3