

## Summary:

### Introduction :

Let us first introduce topic of our paper, “*Efficient Search for Association Rules*”. We have something that is familiar that is **association rules** and something to explore that is **efficient search** for it using algorithm other than Apriori algorithm.

Paper argues that for some applications direct search for association rule is more efficient than two stage Apriori algorithm which first finds the large item set which are then used to identify the associations.

Overall paper trying to prove that Apriori is having a large computational overhead when a number of frequent item set is very large.

Basic algorithm presented in the paper is the OPUS (Optimum Pruning for Unsorted Search spaces) algorithm that basically used for efficient search for the association rule. Algorithm presented is said to be the computationally efficient for association rule analysis.

### Background :

Early approaches to identifying the interestingness of the rule are purely dominated by attempts to form small sets of rule for accurate classification of further previously unseen data.

#### *Two approaches:*

First approach is to eliminate the objective of using the rules for classification, which in turn eliminates the requirement that a small number of the rules be identified. Here they tried to get all the rules that satisfy some criterion of interestingness.

Second approach is “*Association Rule Discovery*” which has different intent from most of the other paradigms.

### **What is the difference in two approaches?**

Answer is while other paradigms have concentrated on finding the rules that are predictive of a single, pre selected, class variable, association rule discovery has been motivated by finding rules that actually predict increased values, without limitation on the values that may appear in the consequent of a rule.

### **Aims of the “Association Rule Discovery”:**

- discovering all the rules that satisfy the given set of the constraints.
- An emphasis on processing large training set
- Allowing any available condition to appear as either an antecedent or consequent.

### **What is OPUS ?**

Optimum Search for Unsorted Search spaces is the approach to bring these divergent braches of the rule discovery research back together. Presented variants of the OPUS by Bayarado and Agrawal, developed in the context of the classification rules research, to discover key rules of the type sought by association rule discovery.

OPUS is distinguished by its ability to efficiently find the pre specified number of rules that maximize an arbitrary function measuring rule quality.

### **Key Point :**

The ability to restrict search to a predefined number of target rules can allow new algorithm to efficiently process search spaces as major concern of the algorithm is to reduces the passes thru the database to increase the efficiency of computation. Which might be the major drawback if not all the data maintained in the memory otherwise need not to be a serious handicap.

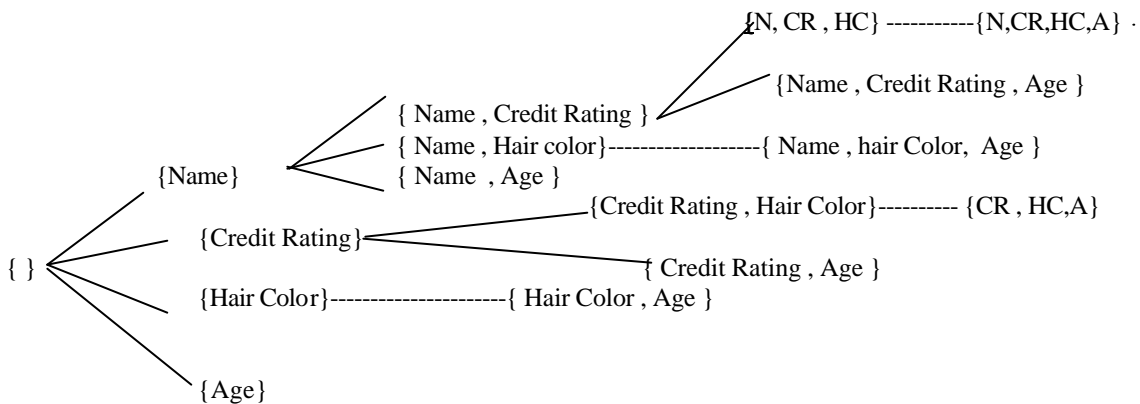
### Drawbacks of Apriori Algorithm ?

- For large item set the overhead of the maintenance and manipulation of the item set can severely impact upon the computational feasibility of the approach.
- It gives computational burden and reduces the efficiency.

For e.g. Applying the Apriori algorithm to the dataset with only 120 items, with many items occurring very frequently, results in 14,567,892 large item set and this example is enough to describe the computational burden it may have on the machine.

### Why OPUS ? An Example :

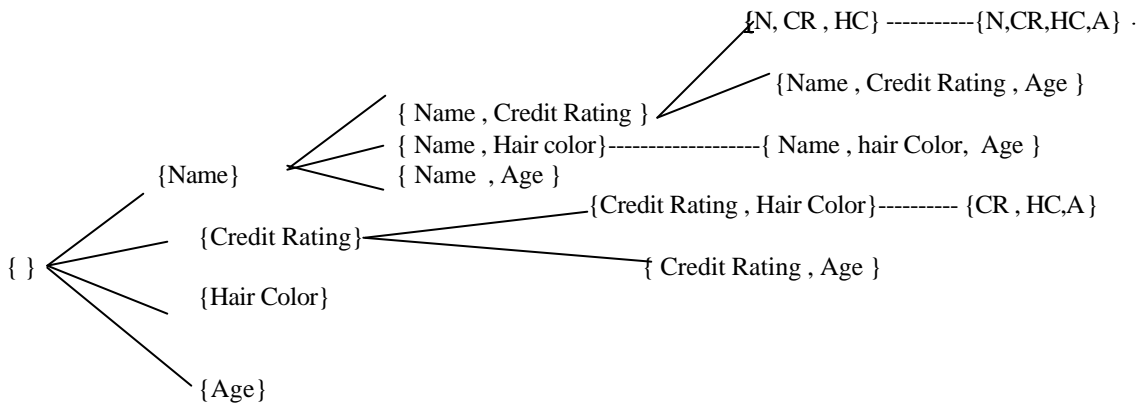
Consider three figures below :



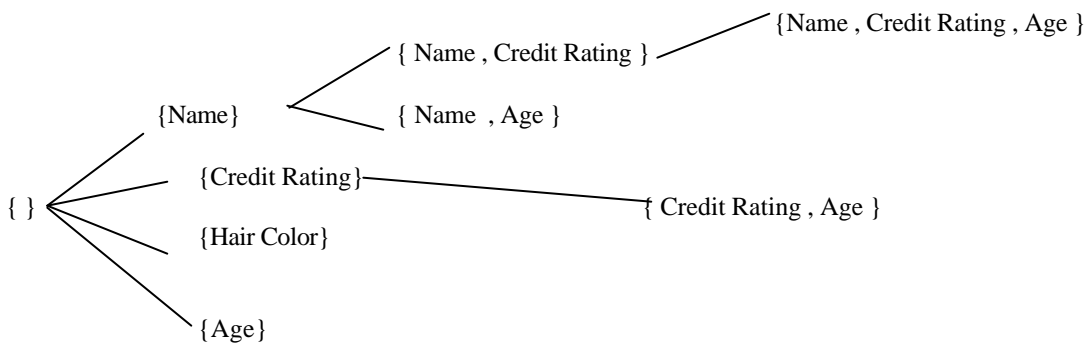
**Fig. 1 : A Fixed Structure Search Space**

Consider figure 1 & 2:

Here the search space is exponential in size. If there are 10,000 conditions, a figure commonly exceeded in market basket analysis, the search space size is  $2^{10000}$ . Even if it is pruned it is only able to remove *one node* from the search space as in fixed structure search algorithm typically seek the branches that does not contain solution.



**Fig. :2 Pruning a branch from Fixed Structure Search Space**



**Fig.:3 Pruning All Nodes Containing a single operator from Fixed Structure Search Space**

**Consider figure 3:**

Consider the “rule discovery” approach in which many pruning rules are consider for a given node N whether any search node in the space below N that contains a given condition C can be a solution. If we identify for example that no node containing the “Hair Color” may contain a solution then outcome would be the removal of all the nodes containing “Hair Color” that approximately halves the search space and that is the key idea behind the algorithm called OPUS.

### **What is the discussion all about?**

Above example and discussion demonstrates that using OPUS search and pruning the search space on the basis of inter-relationships between item set, it can be feasible to perform the efficient association rule analysis on data set for which the apriori is infeasible.

Whether or not this is useful depends upon the whether there are inter-item set constraint that should be applied for a given association rule application.

It seems plausible, however that for many applications an upper limit on the number of the association rules to be generated will be appropriate, and this can be all that is required to enable the efficient search.

### **What is the conclusion of the discussion?**

Paper presents the algorithm for association rule discovery analysis based on the efficient OPUS algorithm. This approach is distinguished from the widely utilized Apriori algorithm by its ability to use inter relation between the item sets to constrain the number of the item set that are considered.

It is also distinguished from the number of other algorithms, that have presented as the alternatives of the Apriori algorithm, by exploring associations containing all available conditions as consequents.

### **Main Disadvantage:**

This approach has a potential disadvantage that it requires many more passes through the data than Apriori. Where the data can be maintained in the main memory this need not be the serious handicap.

### **Key Advantage :**

Some facts that demonstrate the advantage of using the OPUS :

For Cover Type Data set having 581,012 cases described by 55 attributes. The ten continuous valued attribute were discretized in to three sub ranges and remaining were all binary. Overall 120 attribute-values.

**Apriori:**

Requires generation and analysis of 14,567,892 itemset. Even when the item set size is restricted to five.

Total time required : 96 CPU hours.

**OPUS:**

Constraint: Find the 1000 association rule with the highest values of lift required evaluation of only 384,312 rules and 84,649 distinct antecedents.

Total time required: 50 CPU minutes.

Above facts represent the comparison of the Apriori with the OPUS algorithm.

All of the above is the summary of the paper selected by the Manoj Gadhiya and Prakash Patel for the presentation on date 11<sup>th</sup> April .