C3AA10: Random Forest Deep Clustering

Jaya Meena (IIT2018029), Atul Kumar (IIT2018030), Raushan Raj (IIT2018031)

IV Semester B. tech, Information Technology,

Indian Institute of Information Technology, Allahabad, India

Abstract: In the following paper, we deduce an algorithm to implement clustering with the implementation of random forest algorithm.
We have discussed the space complexity and time complexity of the algorithm by both Apriori and Aposteriori analysis. The application of Random Forest Algorithm is also discussed.

Contribution: we discussed this question among the group members. After that each of us proposed a solution of the problem. We later discussed and analysed all solution and came up with this one.
Jaya Meena: analysed the algorithm and worked on the pseudo code and Apriori and Aposteriori analysis. and on graph modelling

Atul Kumar: worked on the space complexity and graph modelling of analysis.

Raushan Raj: worked on Application of random forest algorithm and model time complexity.

Keywords:

1. Random Forest Algorithm
2. Clustering
3. Decision Tree
4. Time Complexity
5. Space Complexity

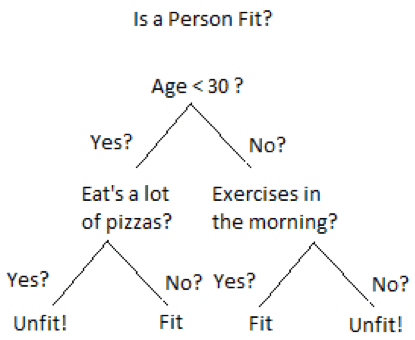
I. INTRODUCTION

Clustering is unsupervised learning and it is dividing a data set into a group of data that belongs to a group or have similar data within each group and dissimilar to the data in another group. It forms the cluster of data based on their similarity and dissimilarity among them.

In this paper we will do clustering by using random forest algorithms.

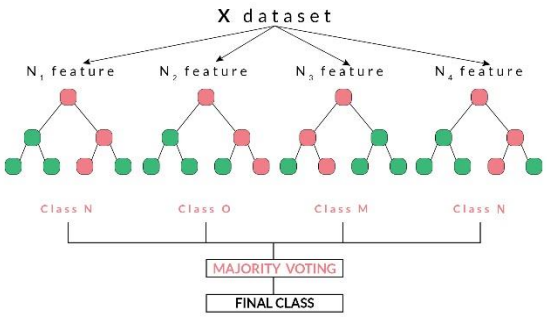
Random forest is method that construct a "forest" of multiple decision trees. The final output is made on the based on the decision of majority of the decision trees.

A decision tree is a tree like structure which is used for making decisions or outcomes. It is a one-way algorithm and contains conditional control statement at each level. The nodes represent the condition with an attribute and the branch represents the outcome of the condition. The terminal nodes hold a class label.



For example: to check whether a person is fit we deduce the condition that if a person is young (<30) then he is healthy is he don't eat a lot of pizzas and if the person is older (>30) then he is healthy is he exercise daily. The root node states the condition based on the attribute age and the following branches direct are affirmative or negative result of the condition. If the person is <30 then the next nodes check whether the person eats a lot of pizza, if yes then the branch leads to the "unfit" class label and if yes then the branch leads to the "fit" class label. If the person is >30 then the next nodes check whether he exercise daily. If yes then the branch leads to the "unfit" class label and if yes then the branch leads to the "fit" class label.

Random forest takes many decorrelated decision trees and based on all of their outcomes selects an outcome supported by majority number of decision trees.



The dataset has many independent attributes. The dataset is divided into subsets and each subset are passed through a decision tree with some features on them. The class label selected by majority of the decision trees is the final class of the data.

II. PROPOSED MODEL

Clustering is the task to partition the data into subsets so that the data is a subset are similar and different from the data in the other sets.

In this model we will use Random Forest Algorithm to perform clustering on the set of data set. Random Forest Algorithm is basically used for regression and classification. Here, we will give the number of clusters we want to create like we do in K-Means. Since decision tree is used to improve the purity and split the dataset. While, clustering is used to split the data to have purer groups.

So here we used Random Forest in the approach used for classification to perform clustering where number of clusters are given.

A dataset is given with a list of features, we will train our data and divide the dataset is some groups based on some features.

Randomly select k features out of m features of the data sets where (k<m).
Among the k features, calculate the node ‘d’ using the best split point.
Split the ‘d’ node into daughter nodes using best split.
Repeat the above three processes until leaf nodes are present.
Build the forest by repeating the above processes for creating n trees.

III. ALGORITHM DESCRIPTION

Suppose we have dataset of fruits (mango, grapes, pineapple, banana) with its features (colour, diameter, shape, season). This algorithm divides the data into 4 clusters, and shows the name of the fruits based on the feature list.

We are given a dataset where 0 is mango, 1 is grapes, 2 is pineapple, 3 is banana and the list of features have colours (yellow-0, green-1, orange-2), diameter, shape (circle-0, banana's shape-1, pineapple's shape-2), season (summer-0, winter-1, allseason-2).

We are given a list of features and we have to find in which cluster the data fits based on its features.

Step 1: First, we will randomly select ‘k’ features from total 4 features (k<m).

Step 2: After selecting those k features, we will build decision trees based on these k features.

Step 3: Repeat Step 1 and 2 till the number of the decision trees you want in your forest

After getting the decision of the all the decision tree in the forest the class label of the majority number of decision tree is selected and it is the decision of the whole the Random forest as a whole.

We need to understand how decision tree works:

Decision tree uses a tree representation we put Boolean function on the discrete attributes at the node of the tree. First, we consider the whole training set as the root. Records are distributed recursively on the basis of attribute values.

Step 1: Put the best Boolean function associated with the best attribute as the root node of the tree.

Step 2: This will split the dataset into subsets.

Step 3: Repeat Step 1 and 2 until you reach the leaf nodes. The class label of the leaf nodes is the output.

We will first train the model with the training features and variables list and then a list of features is given by the user as input to and the list is fitted into the trained model to put the features in one of the cluster in which the data set is divided.

IV. ALGORITHM AND ANALYSIS

Algorithm: Given a training set find that in which cluster to the list of features given by user resides.

Input: list of features. (List)
Output result: the value of the cluster.

Method:

***Procedure*:**

Precondition:
Training set = (x1, y1), (x2, y2), . . ., (xn, yn);
No. of trees = T;
Features = F;

```
function RanForestBuild (S, F)
        H ← ∅
        for i in range [1, B] do
                S (i) ← A sample from S
                hi ← RandomDecisionTree (S (i), F)
                H ← H ∪ {hi}
        end for
        return H
end function
function RandomDecisionTree (S, F)
        At each node:
                f ← very small subset of F
                Split on best feature in f
        return the learned tree
end function
```

Algorithm Ends

V. APRIORI ANALYSIS

The time complexity of the above algorithm depends is:

The time complexity to make a complete decision tree is O(n log(n)), where n is number of variables or record in the list, and log(n) is the depth of the tree.

For v number of features the time complexity is O (v * n log(n)).
Now in Random Forest Algorithm we don't use all the number of features in a tree so let us assume we use k number of features then the time complexity is O (k * n log(n)). (K<4)
In the algorithm we make 100 decision trees so the time complexity is O (100 * k * n log(n)).

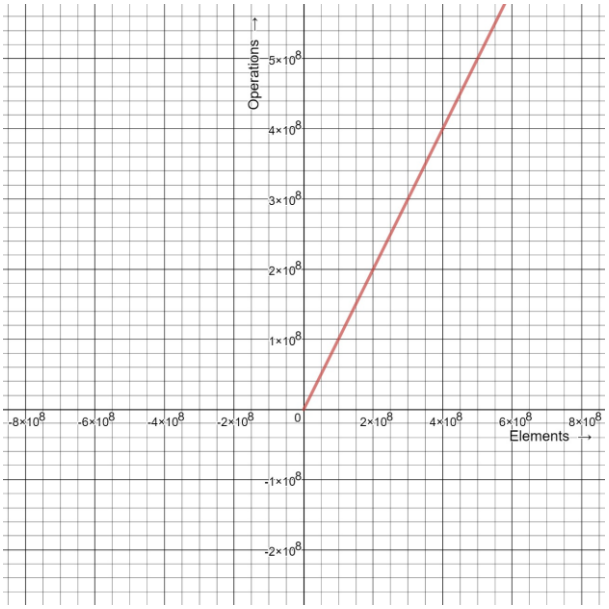
The space complexity of the above algorithm depends is:
The space complexity to make a complete decision tree is:
O(n) where n is number of variables or record in the list.
The space complexity to of a random decision tree is: O(n*100).

VI. APOSTERIORI ANALYSIS.

For experimental analysis we will measure the execution time for our algorithm for different values of N and then plot the graph for the time taken vs the values of N.
For checking the space complexity, we will plot the graph between the value of N and the space consumed by the program with value N.

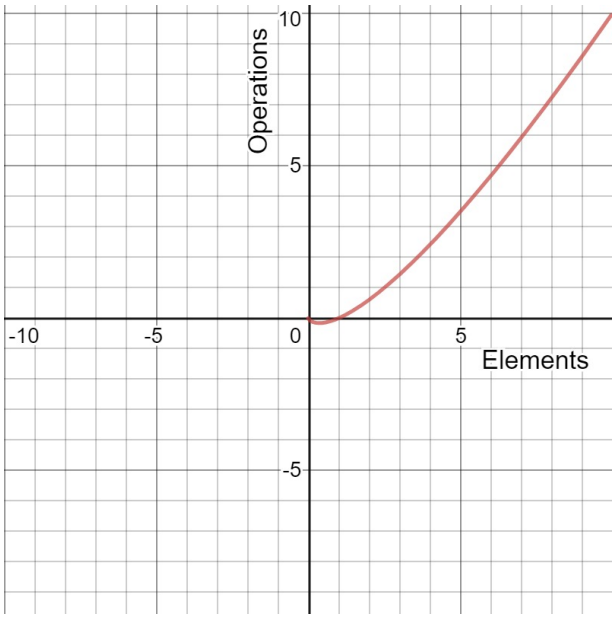
Time complexity:

Number of N	Time(us)
$2*10^8$	$2*10^8$
$3*10^8$	$3*10^8$
$4*10^8$	$4*10^8$
$5*10^8$	$5*10^8$
$6*10^8$	$6*10^8$
$7*10^8$	$7*10^8$
$8*10^8$	$8*10^8$



In the following graph:
The value of n is denoted the X-Axis,
And time is denoted by the Y-Axis

The graph is similar to a linear graph but for very small value of n we can see some differnces. The following graph shows the variaition with very small value of N.

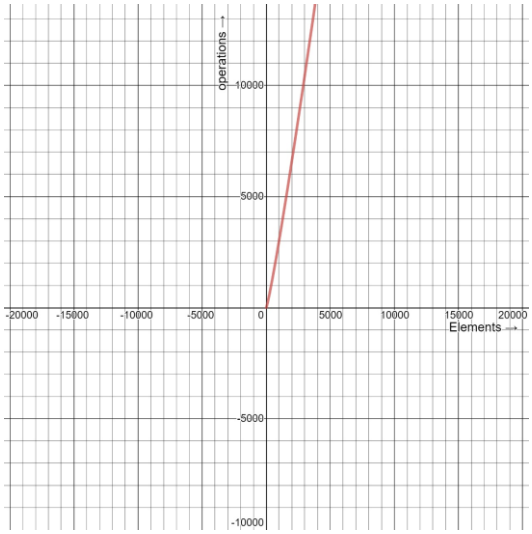


In the following graph:
The value of n is denoted the X-Axis,
And time is denoted by the Y-Axis

The above graph shows that aposteriori analysis is consistent with apriori analysis.

Space complexity:

Number of N	Memory
$3*10^3$	$1*10^4$
$6*10^3$	$2*10^4$
$9*10^3$	$4*10^4$
$12*10^3$	$4*10^4$
$15*10^3$	$5*10^4$
$18*10^3$	$6*10^4$
$21*10^3$	$7*10^4$
$24*10^3$	$8*10^4$



In the following graph:
The value of n is denoted ny X-Axis,
And the memory is denoted by Y-Axis.

The above graph shows that aposteriori analysis is consistent with apriori analysis.

VII. APPLICATION

Random forest algorithm learning is used for both classifications and regression task. Random Forest is not used for clustering, but could be used to create distance metrics that feed into traditional clustering method. It provides greater accuracy. Random forest classifier will maintain the accuracy of a large proportion of data even if some values are missing in the feature list.

Advantages:

- The predictive performance of random forest algorithm is one of the best supervised learning algorithms.

- It provides a more reliable feature importance prediction.
- It is very stable. Even after introduction of new data in the dataset the overall algorithm is not much affected.
- Random forest algorithm handles non-linear parameter efficiently.

Disadvantages:

- Random forest algorithm creates a lot of decision trees, so the complexity of this algorithm is very high.
- It takes longer to train the data since it creates a lot of decision trees and each generates a result out of which the decision with majority votes is selected.
- It causes over-fitting

VIII. CONCLUSION

From this paper we can conclude that the time complexity of clustering using Random Forest Algorithm:
O(100 * k * n log(n)), where k is the number of features selected on an average and n is the number of records.
The space complexity to clustering using Random Forest Algorithm:
O(n*100).

IX. REFERENCES

[1]https://datascience.stackexchange.com/questions/17539/space-complexity-of-classification-algorithms

[2]https://www.educative.io/edpresso/what-is-a-binary-search-tree

[3]https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm

[4]https://bradleyboehmke.github.io/HOML/random-forest.html

[5]https://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/

X. ANNEXURE-I.

```
from sklearn.ensemble import RandomForestClassifier
import pandas as pd
import numpy as np
np.random.seed(0)

#colour,diameter,shape,season
features=[[0,12,0,0],[0,11,0,0],[1,13,0,0],[0,12,0,0],[0,13,0,0],
        [2,12,0,0],[0,11,0,0],[2,10,0,0],[0,13,0,0],[0,12,0,0],
        [1,1,0,1],[1,2,0,1],[1,1,0,1],[1,2,0,1],[1,3,0,1],
        [1,2,0,1],[1,2,0,1],[1,2,0,1],[1,4,0,1],[1,2,0,1],
        [2,22,2,2],[2,19,2,2],[2,23,2,2],[2,22,2,2],[2,20,2,2],
        [2,25,2,2],[2,19,2,2],[2,20,2,2],[2,23,2,2],[2,21,2,2],
        [0,6,1,2],[0,7,1,2],[0,8,1,2],[0,7,1,2],[0,6,1,2],
        [0,6,1,2],[0,7,1,2],[0,8,1,2],[0,7,1,2],[0,6,1,2]]
# (colour) yellow-0, green-1, orange-2
# (shape)  circle-0, banana's shape-1, pineapple's shape-2
# (season) summer-0, winter-1, allseason-2

values=[0,0,0,0,0,0,0,0,0,0,
       1,1,1,1,1,1,1,1,1,1,
       2,2,2,2,2,2,2,2,2,2,
       3,3,3,3,3,3,3,3,3,3,]
#0-mango, 1-grapes, 2-pineapple, 3-banana

clf = RandomForestClassifier(n_estimators=100)
# n_estimator is the number of decision trees

clf.fit(features,values)
#trains the data (Fit the model)

print(clf.predict([[0,9,0,0]]))
#print the result
```

Output: [0]

So, the list [[0,9,0,0]]
Where it means an entity with yellow colour,
9 cm diameter, circular shape, and found in summer season is mango.

How to run the code:

- ***Download python 3.6 or any other version***
- ***Download the following libraries:***
    1. ***NumPy***
    2. ***Pandas***
    3. ***Sklearn***