Introduction To Machine Learning
CSL-603

Report :- Lab 1 (Decision Trees and Forests)

Piyush Jain
2015CSB1023

Introduction

In this assignment we are taking movie data from stanford database and constructing decision tree with or without noise and study the accuracy on test and training dataset and after that pruning the tree to solve the problem of overfitting and later on we have implemented decision forest based on feature bagging and study the effect of number of trees in the forest on the prediction accuracy of the test data set.

Constructing Dataset

After downloading the dataset from link, we had taken our training dataset as 1000 random movie reviews from “labeledBow.feat” file consisting of 500 positive(>= 7) and 500 negative reviews. In same way , we also had taken test dataset and validation dataset (for pruning purpose) from test folder. For attributes we had top 5000 words as attributes.For this we had created “input.cpp” file that take random reviews and store in 6 files .

Constructing Decision Tree

Now using the training data we had successfully implemented decision tree by ID3 algorithm . First of all , for reducing run time we had implemented vector containing list , and list contains pairs of word index and word count and each index of vector represent a movie review.

In ID3 we had passed positive list , negative list and attribute and height for early stopping purpose, for every node we had passed positive list and

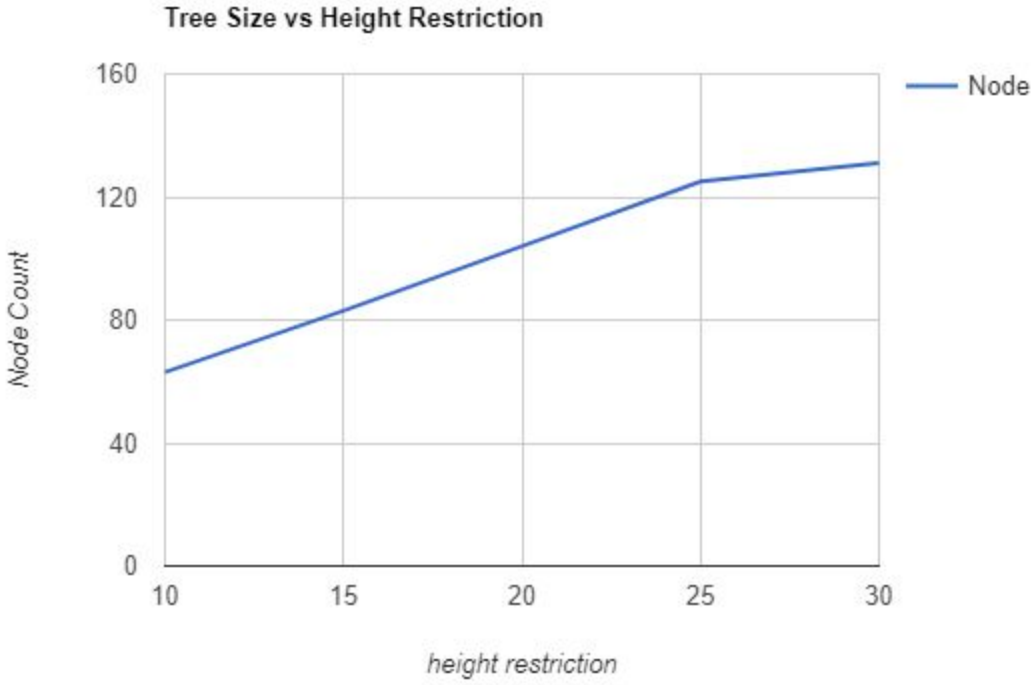
negative list so we had to create four list (positivePresent list , negativePresent list , positiveAbsent list , negativeAbsent list) present list will pass to left child and absent list will pass to right child. For splitting purpose we had set threshold as 1 .

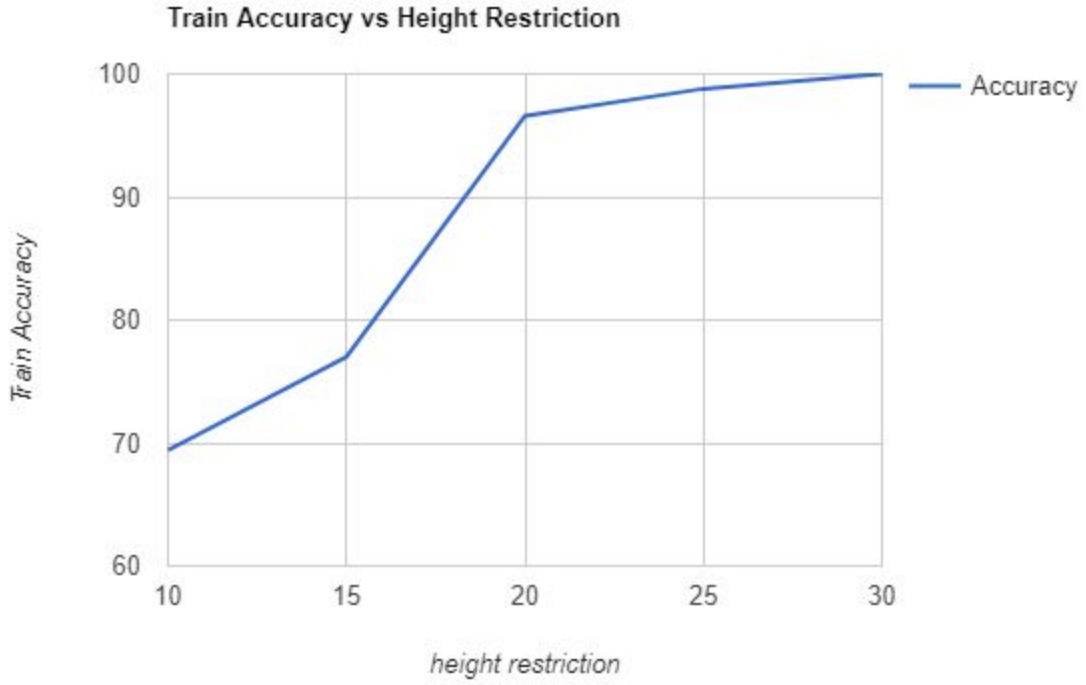
To reduce further complexity we had used a frequency word matrix of size 89527 X 2 so that while calculating information gain for each attribute we get how many positive and negative reviews it is present in O(1) time.So for each node in tree we have to create this frequency word matrix and get which attribute has maximum information gain and then delete that attribute and call recursively until one of the base cases had been occurred.

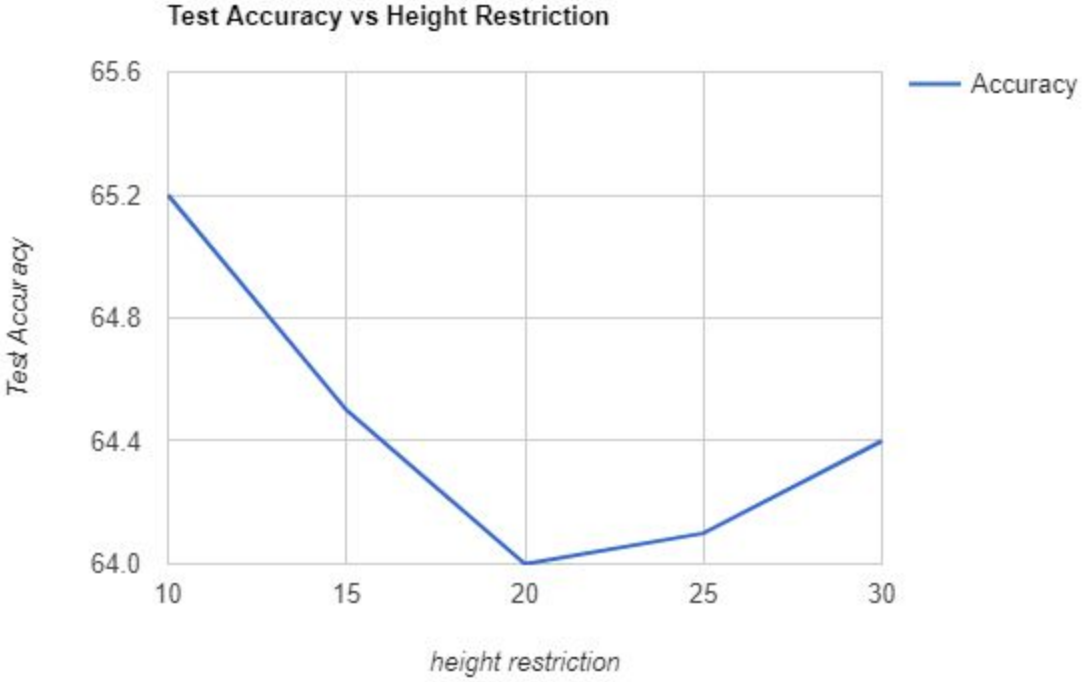
Effects of early stopping on decision tree

We had considered height as early stopping criteria, depending upon various height such as 10, 15, 20, 25, 30 effect on tree nodes and accuracy on training and test data are :-

S.No.	Height	Tree Nodes	Train accuracy	Test accuracy
1	10	63	69.4%	65.2%
2	15	83	77%	64.5%
3	20	104	96.6%	64%
4	25	125	98.8%	64.1%
5	30	131	100%	64.4%
6	No restrict	131	100%	64.4%


Tree Size vs Height Restriction
160
120
80
40
0
Node Count
10
15
20
25
30
height restriction
Node


Train Accuracy vs Height Restriction
100
90
80
70
60
Train Accuracy
10
15
20
25
30
height restriction
Accuracy


Test Accuracy vs Height Restriction
65.6
65.2
64.8
64.4
64.0
Test Accuracy
10
15
20
25
30
height restriction
Accuracy

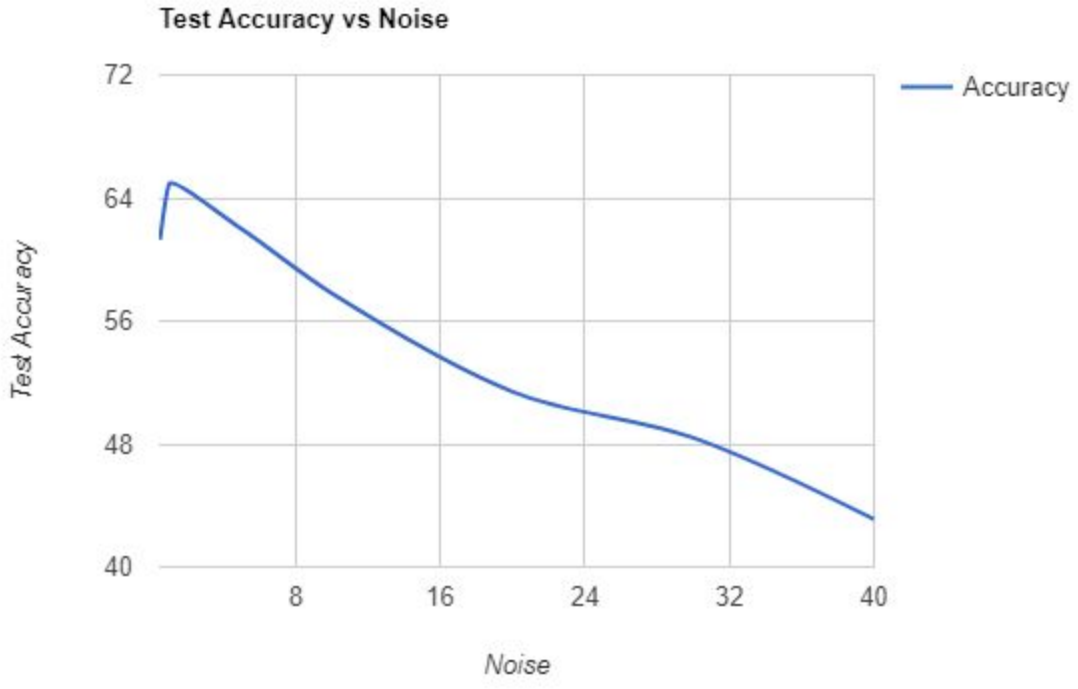
On restricting height obviously tree nodes will get reduced as compared to when there is no restriction, since accuracy on training data will get lowered on further decreasing the height of tree as tree is not well learned yet on train data.

Adding Noise

Noise had been added by swapping first ith data from positive and negative list where ith index is decided by how much percent noise you want to add and then call the ID3 function to construct the tree and measure the accuracy accordingly.

Given below is table telling the accuracy on test data after adding noise : -

S.No.	Noise (in %age)	Test accuracy(in %age)
1	0.5	61.3
2	1	64.9
3	5	62
4	10	57.8
5	20	51.4
6	30	48.4
7	40	43.1



Since on adding noise and after that creating decision tree can lead to decrease in test accuracy ideally as previously we have correct data on adding noise we are changing its label so the tree will contain now more attributes .

Pruning the Tree

Here we had implemented post pruning method which is first search the node whose removal with its subtree gets maximum accuracy on validation set and update the tree and then apply pruning on the updated tree.

S.No.	Characters	Original Tree	Prune Tree
1	Root Index	77	77

2	Height	26	17
3.	Node Count	127	27
4.	Test Accuracy	64.2%	71.1%

Since on pruning height and node count will get reduced and test accuracy will increase obviously (as we are avoiding overfitting) , hence we can infer that above observations are correct.

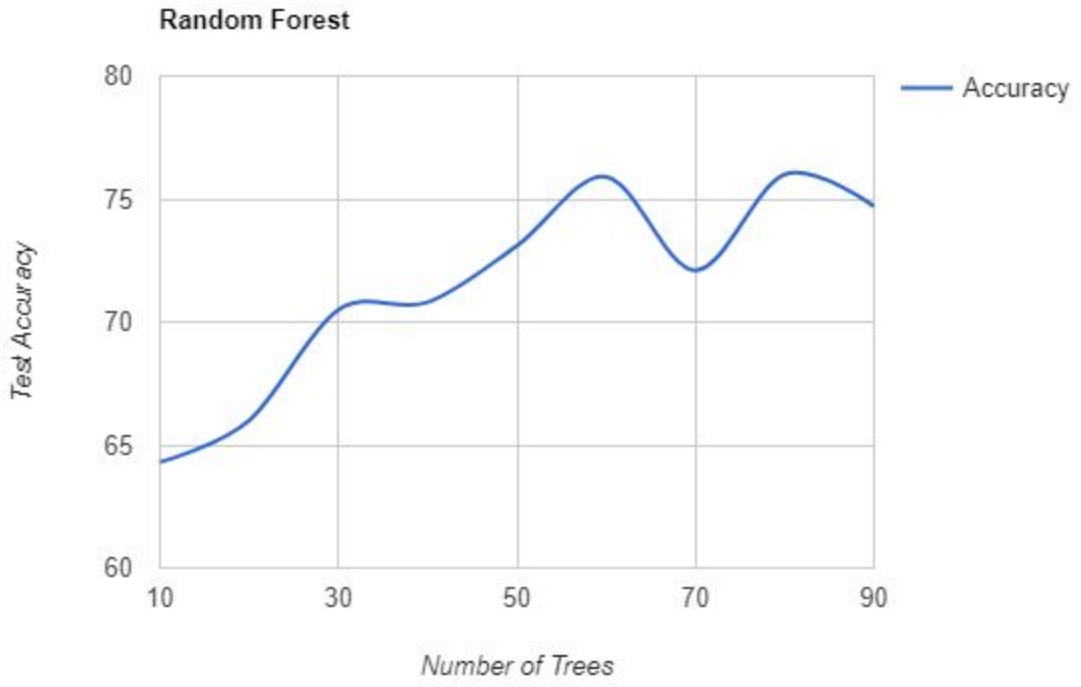
Random Forest

In this forest , we had implemented by taking √D element where D are total number of attributes, so we had implemented random forest by taking 300 attributes with different number of trees and while finding accuracy on test data we get labels from these different trees and find majority among them and check it is correct or not .

Table below represents the relation between number of trees and accuracy on test data.

S.No.	No. of Trees	Train Accuracy(%age)	Test Accuracy(%age)
1	10	100	64.3
2	20	100	66
3	30	100	70.5
4	40	100	70.8
5	50	100	73.1

6	60	100	75.9
7	70	100	72.1
8	80	100	76
9	90	100	74.7



In general, the more trees you use the better get the results. However, the improvement decreases as the number of trees increases, i.e. at a certain point the benefit in prediction performance from learning more trees will be lower than the cost in computation time for learning these additional trees.