COL774 : Assignment 3

Aditi Singla
2014CS50277

9th April 2017

1. Decision Trees (& Random Forests)

   (a) The decision tree has been implemented by choosing the best attribute at every node, and splitting discrete attributes into 0 & 1 and continuous attributes about the median of training data examples present at that node, the accuracies observed were as follows:

   - Over Training Data : 99.977051 % (348527/348607)
   - Over Validation Data : 89.994148 % (104575/116202)
   - Over Test Data : 90.303177 % (104935/116203)

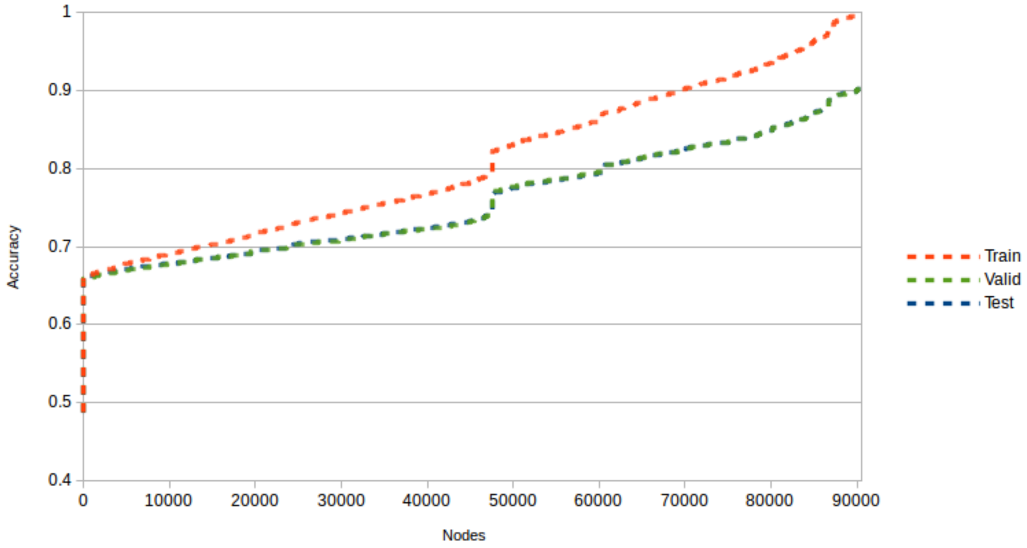


Figure 1: Accuracies on train, validation & test data, while growing the tree

For training, validation and test sets, we observe that there are jumps in the graph. This can be understood by the fact that during the DFS search, we recursively run DFS search on one of the subtrees and then on the other tree. So, when we expand the second child on a node which is near the root, the accuracy increase is way higher than the one observed on expanding a node near the leaves (generally far from the root).

On training data, we observe an almost 100% accuracy, but not exactly 100%, since there is an approximation used, that when no attribute gives better Mutual Information than others, we choose the label of maximum examples at that node.

Similar patterns are observed for all the three, where train gives the best accuracies, while test and validation reach 90% accuracy.

(b) The decision tree is then pruned on the validation data, and the accuracies are noted after every iteration of pruning the best node. The accuracies observed are:

- Over Training Data : 97.697121 % (340579/348607)
- Over Validation Data : 93.612846 % (108780/116202)
- Over Test Data : 90.674934 % (105367/116203)

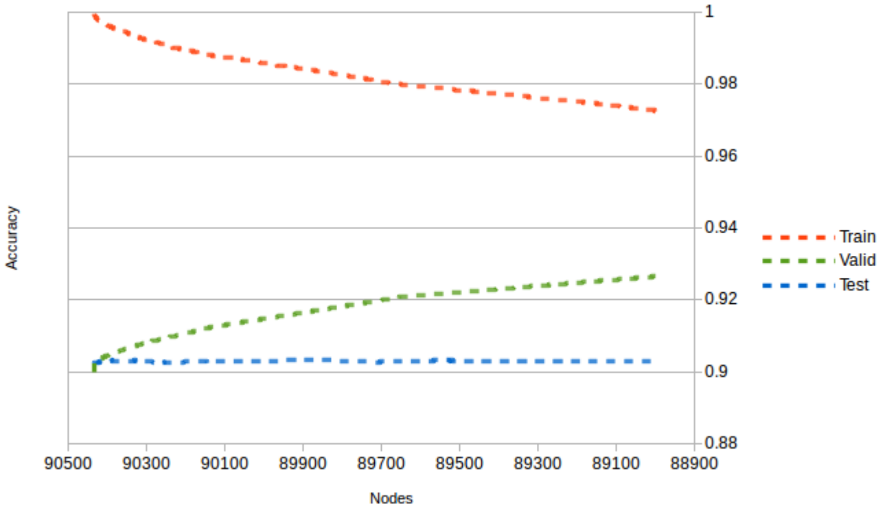


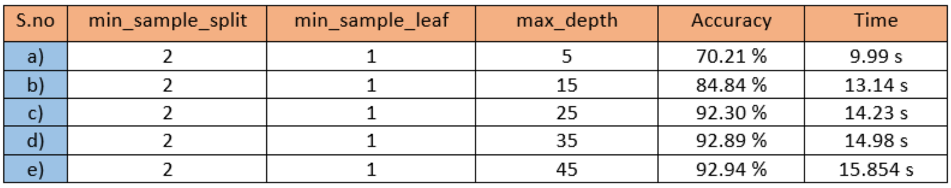
Figure 2: Accuracies on train, validation & test data, while post pruning the tree

While post pruning the nodes in the tress, a decay in the training accuracy is observed since the overfitting nodes are being eliminated. Since the post pruning is being done on the validation set, validation accuracy increases as expected. A very slight increase in test accuracy is observed towards the end, as post pruning intends to make the tree fit the underlying distribution.

(c) Scikit-Learn Library- Decision Trees:

For this part, one of the parameters is varied, keeping the other two fixed. The decision tree was trained on the training data and accuracy was tested on the validation data. A few observations made were:

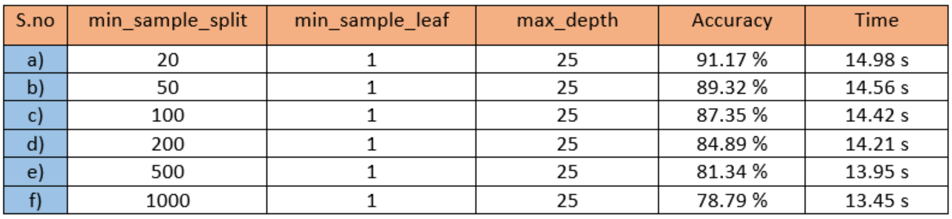
- On varying the max_depth from 5 to 45 at intervals of 10, an increase in accuracy is observed from 70% to 93%, ending up in saturation. So, an optimal value of max_depth is 25, since beyond that the time taken increases unnecessarily, without giving much benefit in terms of benefit in terms of accuracy. The depth at this point is large enough to not underfit the data and small enough to not overfit the data.



S.no	min_sample_split	min_sample_leaf	max_depth	Accuracy	Time
a)	2	1	5	70.21 %	9.99 s
b)	2	1	15	84.84 %	13.14 s
c)	2	1	25	92.30 %	14.23 s
d)	2	1	35	92.89 %	14.98 s
e)	2	1	45	92.94 %	15.854 s

Figure 3: SciKit-Learn Decision Trees : Varying max_depth

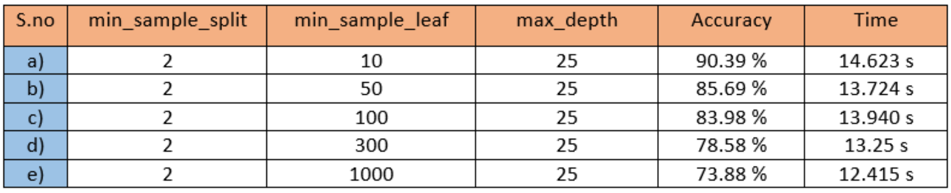
- On varying the min_samples_split from 20 to 1000, a decrease in accuracy is observed, along with a decrease in the computation time. This seems logical since amount of splitting of nodes reduces as we increase the parameter. This might form a tree which underfits the data and the tree may not capture the underlying distribution.



S.no	min_sample_split	min_sample_leaf	max_depth	Accuracy	Time
a)	20	1	25	91.17 %	14.98 s
b)	50	1	25	89.32 %	14.56 s
c)	100	1	25	87.35 %	14.42 s
d)	200	1	25	84.89 %	14.21 s
e)	500	1	25	81.34 %	13.95 s
f)	1000	1	25	78.79 %	13.45 s

Figure 4: SciKit-Learn Decision Trees : Varying min_samples_split

- On varying the min_samples_leaf from 10 to 1000, again a decrease in accuracy is observed, along with a decrease in the computation time. So, few nodes maynot be expanded due to few supporting datapoints.



S.no	min_sample_split	min_sample_leaf	max_depth	Accuracy	Time
a)	2	10	25	90.39 %	14.623 s
b)	2	50	25	85.69 %	13.724 s
c)	2	100	25	83.98 %	13.940 s
d)	2	300	25	78.58 %	13.25 s
e)	2	1000	25	73.88 %	12.415 s

Figure 5: SciKit-Learn Decision Trees : Varying min_samples_leaf

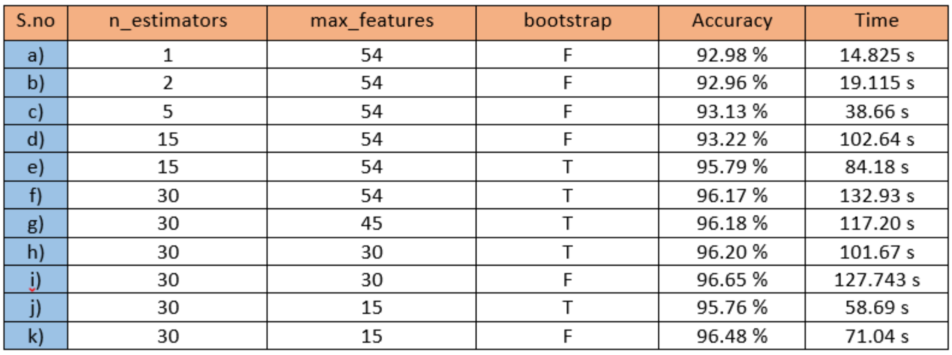
Out of these, the best values observed are at min_depth = 25, min_samples_split = 20 & min_samples_leaf = 10. The accuracies observed are:

- Over Training Data : 100 % (348607/348607)
- Over Validation Data : 92.93 % (107981/116202)
- Over Test Data : 92.97 % (108061/116203)

We observe better results for training and test data, but less accuracy for validation data. While training the data, due to a few approximations, we didn't achieve 100% earlier. For validation, pruning yielded better results. And for test data, pruning maynot be good enough to give results on unknown data.

(d) Scikit-Learn Library- Forests:

For this part, one of the parameters is changed at every step. The first one is on a single tree, considering all the attributes and no bootstrapping. A few observations made were:



S.no	n_estimators	max_features	bootstrap	Accuracy	Time
a)	1	54	F	92.98 %	14.825 s
b)	2	54	F	92.96 %	19.115 s
c)	5	54	F	93.13 %	38.66 s
d)	15	54	F	93.22 %	102.64 s
e)	15	54	T	95.79 %	84.18 s
f)	30	54	T	96.17 %	132.93 s
g)	30	45	T	96.18 %	117.20 s
h)	30	30	T	96.20 %	101.67 s
i)	30	30	F	96.65 %	127.743 s
j)	30	15	T	95.76 %	58.69 s
k)	30	15	F	96.48 %	71.04 s

Figure 6: Scikit-Learn: Random Forests

- On increasing the n_estimators, which is the number of trees, time taken increases, since the trees being trained increase in number. Also, the accuracy increases with the increasing number of trees.
- On varying the max_features, which is the maximum number of attributes allowed, we observe the maximum validation accuracy at around 30, since at this point, the forest neither overfits nor underfits the data.
- Bootstrapping means to pick random samples from the data. On bootstrapping, tree builds faster but with slightly less accuracy, while max_features is in the middle of ts range.

Out of these, the best values observed are at n_estimators = 30, max_features = 30 & bootstrap = True. The accuracies observed are:

- Over Training Data : 100 % (348607/348607)
- Over Validation Data : 96.61 % (112266/116202)
- Over Test Data : 96.66 % (112326/116203)

We observe far better results for training, validation and test data. This is because random forests allow better understanding of the underlying pattern and hence better accuracies on all three.

2. Neural Networks

(a) The stopping criteria for this is number of iterations (=40), since convergence was very slow. The learning rate is 0.003 and the number of hidden units is set to 100. The algorithm take about 7 minutes to train the neural network. The accuracies observed were:

- Over Training Data : 82.25 %
- Over Test Data : 71.45 %

(b) The learning rate is dynamically varied w.r.t the number of iterations (inverse square root). The convergence becomes slower with the decreasing learning rate. The accuracies observed were:

- Over Training Data : 82.16 %
- Over Test Data : 71.67 %

(c) The number of perceptrons in the hidden layer are varied from 50 to 500. The results were as follows:

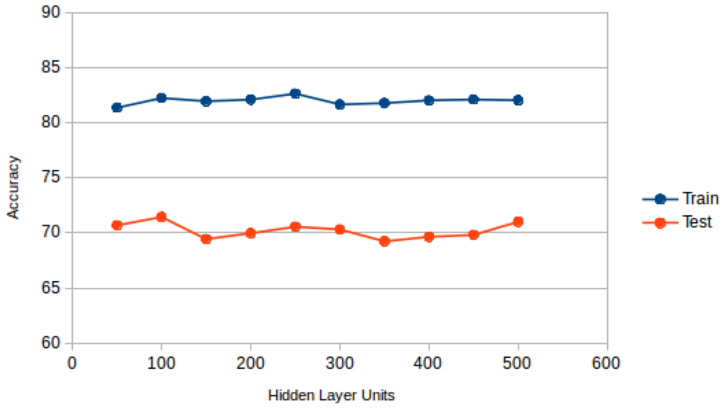


Figure 7: Neural Network

Due to the random initialisation of weights here, no definite pattern is observed in the test accuracy. But we can conclude that the increase in hidden layer units doesnot give any benefit to the test set accuracy.

(d) With the softplus activation function, 100 hidden layer unita and 0.003 learning rate, the accuracies observed are:

- Over Training Data : 82.11 %
- Over Test Data : 69.7898 %

This activation performs worse than the sigmoid function.