(CS534) Machine Learning

Implementation Assignment 4: Decision Tree and Random Forest

A-young Kang (ID:933-610-350)

General Introduction

In this assignment, the decision tree algorithm is implemented in Part 1. We use information gain to pick the best feature to split and observe how the training and validation accuracies vary as the maximum depth of the tree increases from 2 to 50. In part 2, the Random Forest algorithm is implemented. We tune parameters—the number of trees (T), the number of features to sub-sample (m) and the maximum depth of the trees (dmax) and see how these parameters impact the performance of our random forest.

Part 1. Decision Tree

(a) What are the first three splits selected by your algorithm? This is for the root, and the two splits immediately beneath the root. What are their respective information gains?

- Root: Previously_Insured (Information gain: 0.3076)
- Left child: Vehicle_Damage (Information gain: 0.0349)
- Right child: Vehicle_Damage (Information gain: 0.0043)

(b) Evaluate and plot the training and validation accuracies of your trees as a function of dmax. When do you see your tree start overfitting?

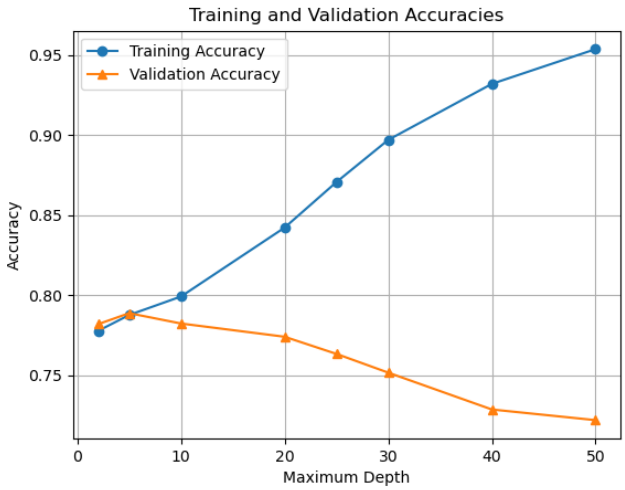


Figure 1. Training and Validation Accuracies of the Decision Trees

Figures 1 shows the training and validation accuracies of the decision trees. As the maximum depth increases from 2 to 5, both the training and validation accuracy increase. However, when the maximum depth is 10, the training accuracy increases but the validation accuracy decreases. Thus, we can conclude that the overfitting starts when the maximum depth is somewhere between 5 to 10. In order to figure out the exact maximum depth, we learn from the training data with *dmax* = 2, 5, 6, 7, 8, 9, 10 (Figure 2) and found out that when the maximum depth is 6, the tree starts overfitting.

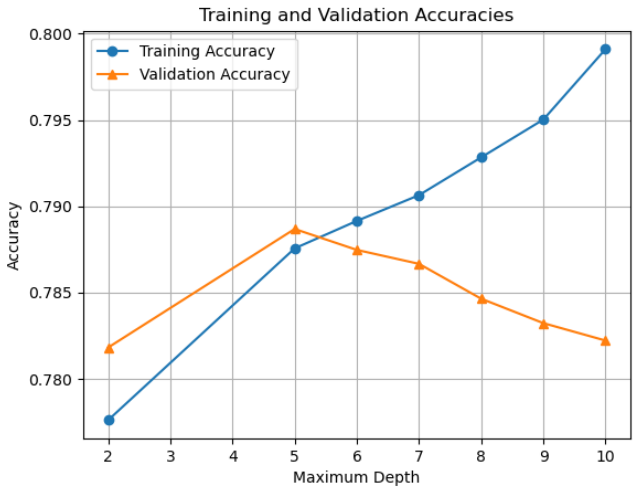


Figure 2. Training and Validation Accuracies of the Decision Trees with dmax ∈ [2,10]

Part 2. Random Forest

(a) For each dmax value, create two figures, one for training accuracy and one for validation accuracy. For the training accuracy figure, it will contain four curves, each showing the train accuracy of your random forest with a particular m value as we increase the ensemble size T = 10; 20, …, 100. That is, plot the training accuracy (y axis) as a function of the ensemble size T (x-axis), for each m value. Be sure to use different colors/lines to indicate which curve corresponds to which m value, and include a clear legend for your figure to help the readability. Repeat the same process for validation accuracy. Compare your training curves with the validation curves, do you think your model is overfitting or underfitting for particular parameter combinations? And why?

Figure 3 shows the training and validation accuracies of the random forest. When (*dmax* = 2 and *m* = 5, 25) and (*dmax* = 10 and *m* = 5), both the training and validation accuracies are low and fluctuate. In this case, the model is underfitting. Increasing the *m* value generally improves the performance of the model because we have a more number of options to be considered. If *m* is substantially smaller than the original number of features, the model has higher chance of splitting on irrelevant features. When *dmax* = 25 and *m* = 25, 50 ,100, my model is overfitting but not by much. This is because deep trees can incur unnecessary variance and thus, they are prone to overfitting.

(b) For each dmax value, discuss what you believe is the dominating factor in the performance loss based on the concept of bias-variance decomposition. Can you suggest some alternative configurations of random forest that might lead to better performance for this data? Why do you believe so? Are there any issues inherent with the data you can find that make the performance increase difficult?

When *dmax* = 2, 10, the dominating factor is the number of features to sub-sample (*m*) because increasing the number of features usually improves the performance of the model as I mentioned above. However, this is not necessarily true as this decreases the diversity of individual tree. When *dmax* = 25, the validation accuracies are similar regardless of the *m* value. As the model gets complex, bias decreases but variance increases. As can be seen in Figure 3(f), the validation accuracies are a little higher when the Ensemble size is greater than or equal to 20 compared to *T* = 10. Thus, we can see that bagging reduces the variance of the model. When sub-sampling the features, if we find the feature importance and allow important features to be selected more instead of randomly sub-sampling, it would lead to better performance because splitting on inappropriate features creates unnecessary fragmentation of the training examples for learning. Also, by pre-processing of the data and removing irrelevant features, we would be able to improve the accuracy. Expected Loss is decomposed into bias, variance and noise. We cannot deal with noise in our data, which makes the performance increase difficult.

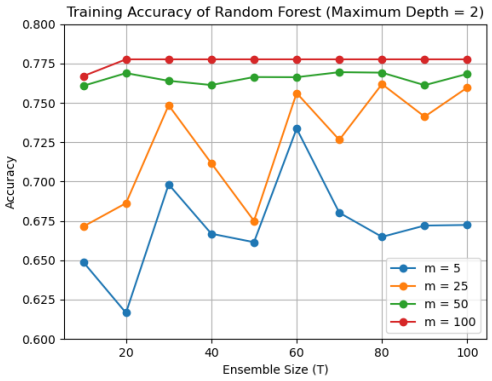
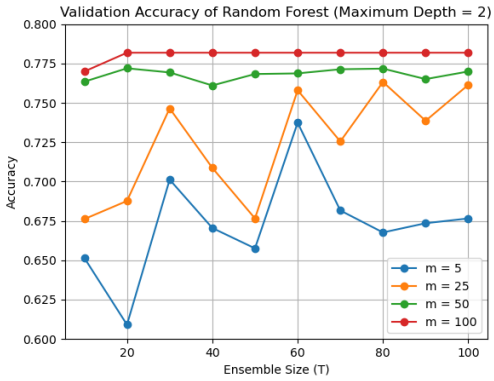
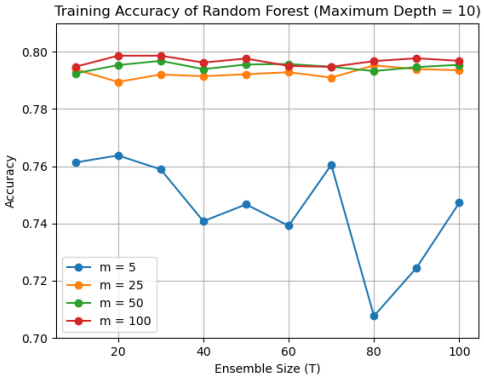
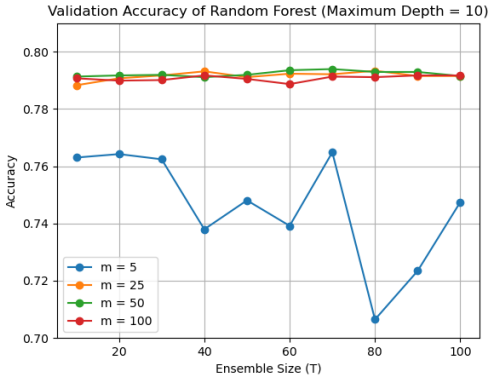
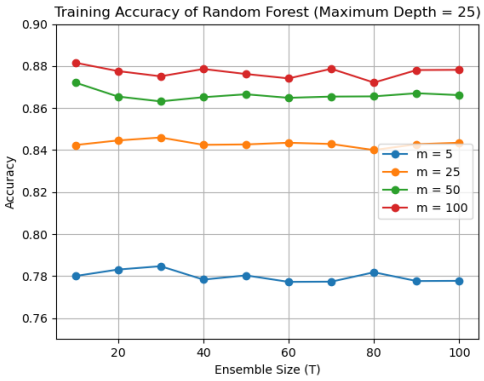
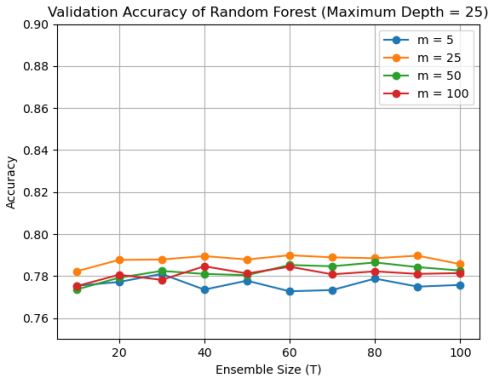
Training Accuracies	Validation Accuracies
	
(a) *dmax* = 2	(b) *dmax* = 2
	
(c) *dmax* = 10	(d) *dmax* = 10
	
(e) *dmax* = 25	(f) *dmax* = 25

Figure 3. Training and Validation Accuracies of Random Forest with different dmax values