Oregon State University

AI_534_001_F2021 MACHINE LEARNING

Implementation Assignment 4

Professor: Xiaoli Fern
Student: Chengxu Xu

Part1

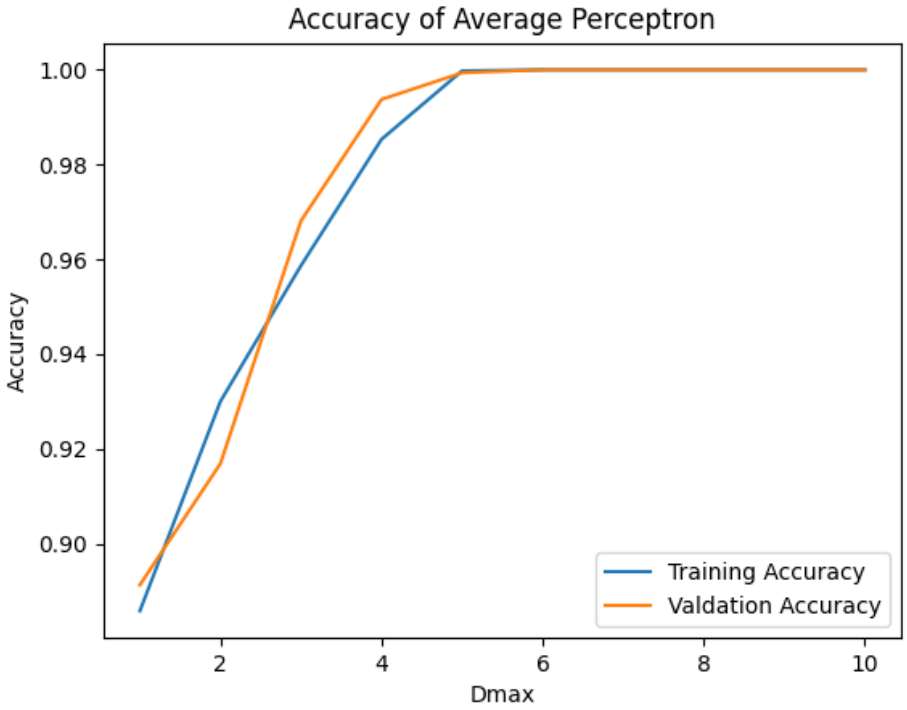
(a) What are the first three splits selected by your algorithm? This is for the root, and the two splits immediately beneath the root. What are their respective information gains?

The first three splits selected by my algorithm and their respective information gains was show in the image:

```
Root of the tree:                       gill-spacing=c
Respective information gains of it:     0.6780237353936194
Second split of the tree:               odor=l
Respective information gains of it:     0.2792410341573887
Third split of the tree:                cap-surface=f
Respective information gains of it:     0.2604641992961125
```

(b) Evaluate and plot the training and validation accuracies of your trees as a function of dmax ranging from 1 to 10. At which depth does the train accuracy reaches to 100%? If your tree could not get to 100% before the depth of 10, keep on extending the tree in depth until it reaches 100% for the train accuracy. Do you observe any overfitting?

The generated accuracies vs dmax images are as follows. The training accuracy reaches 100% at depth of 6. In most cases the accuracy were underfitting and the overfitting case only occurs once at depth of 2.

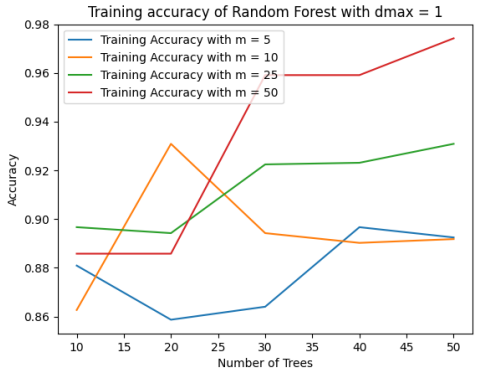


Part2

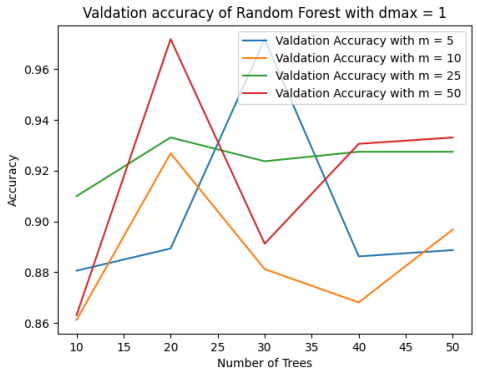
(a) For each dmax value, create two figures, one for training accuracy and one for validation accuracy. The training accuracy figure should contain four curves, each showing the train accuracy (y-axis) of your random forest with a particular m value as a function of T (x-axis). Be sure to use different colors/lines to indicate which curve corresponds to which m value, and include a clear legend 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?

Dmax = 1:

Training:

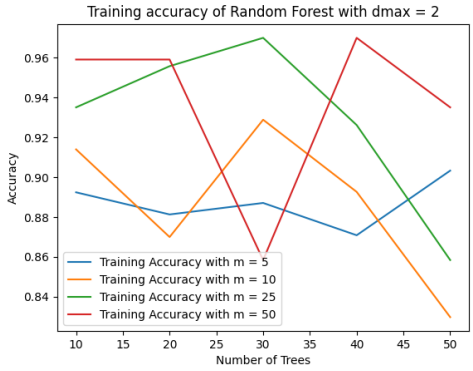


Valdation:

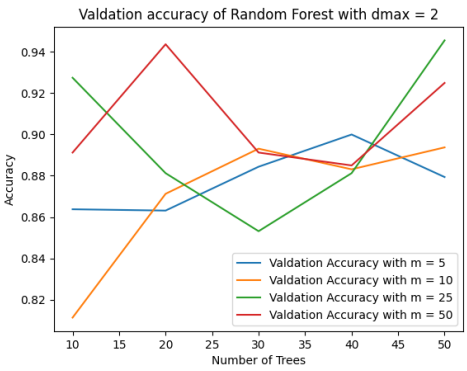


Dmax = 2:

Training:

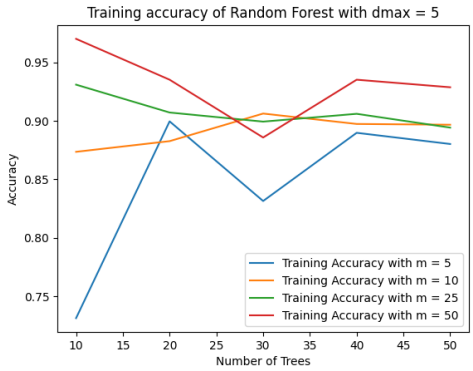


Valdation:

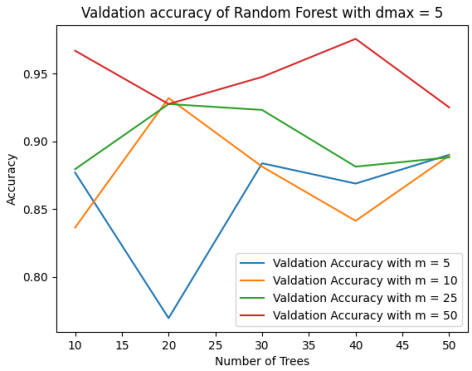


Dmax = 5:

Training:



Valdation:



After observing the generated images, I found that the model is more likely to be underfitting when the dmax is small (1 and 2) and the number of features is high, and more likely to be overfitting when the dmax is large (5) and the features are small.

I think one possible reason for the overfitting phenomenon is that when the number of features is small and the depth is large, more noise data are generated in the process of tree generation, thus affecting the accuracy of verification.

(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?

When dmax = 1 or 2, I think the dominating factor in the performance loss based on the concept of bias-variance decomposition is the number of trees, so I think a more appropriate forest configuration is to use a suitable value of T, say T = 20, because the accuracy from the image is most affected by T;

When dmax=5, I think the dominating factor in the performance loss based on the concept of bias-variance decomposition is number of features, so I think a more appropriate forest I think a more appropriate configuration of the forest is to use a relatively large value of m, such as m = 50, because a higher accuracy is obtained when m is larger in the image;