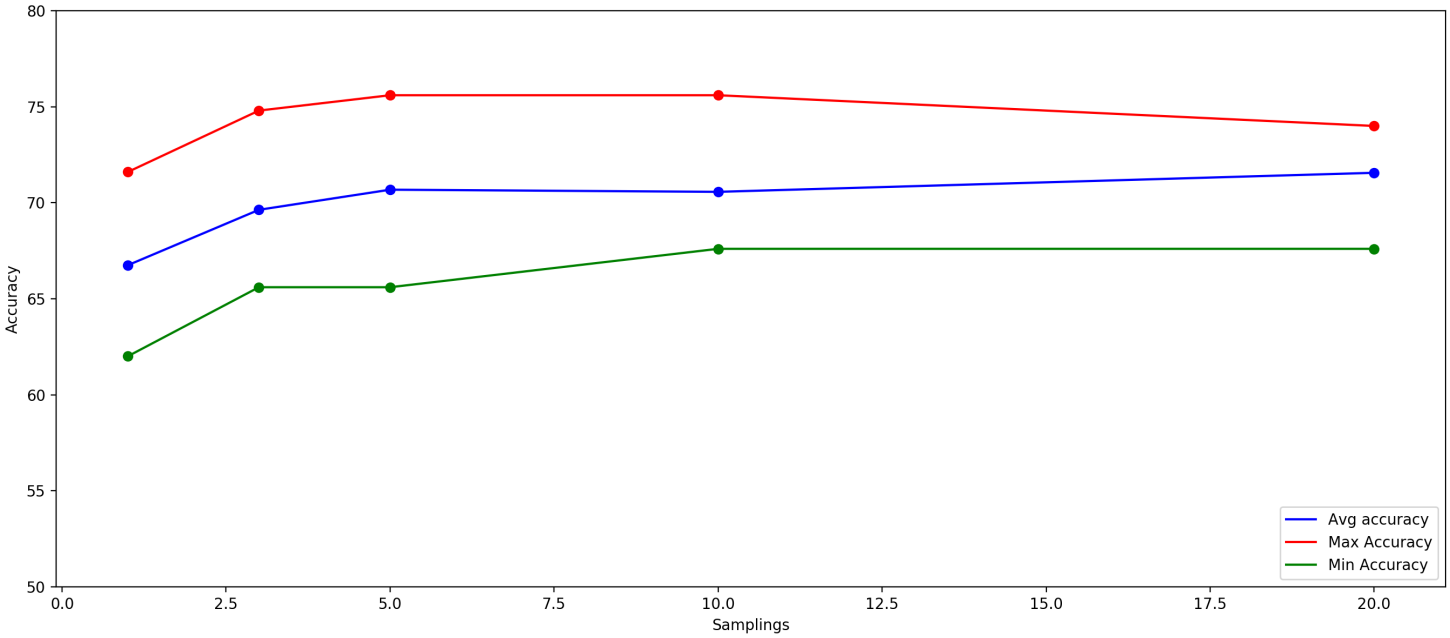
CSEP546 - HW4 Answers

Name: Abhijith Jagannath
Student ID: 1776317

Autumn 2017
12 / 03 / 2017

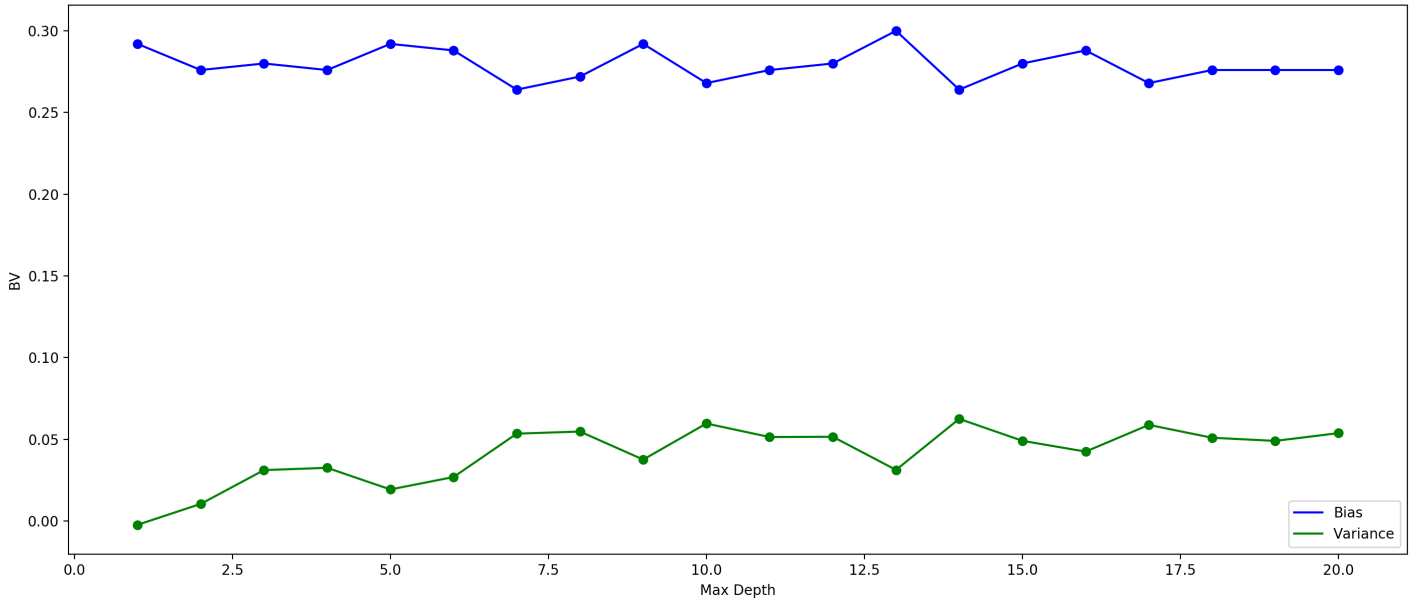
1.0: Test Accuracies of 1, 3, 5, 10, 20 Samplings

Accuracies varied for each run, hence I took an average of accuracies over 50 runs. Plot shows max and min values along with average.



1.1: speculate whether its bias increase or decrease with max_depth?

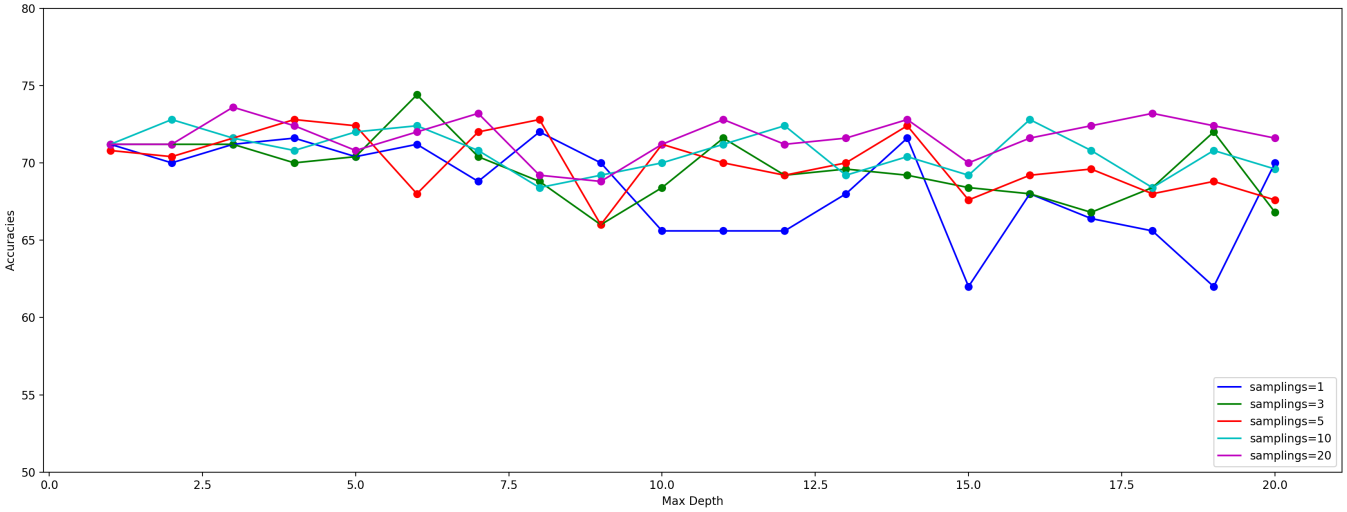
We can say that with increase in max_depth, we will have to traverse more paths. Which means stricter rules. And hence predictions will tend to fall within the strict bounds decreasing the bias and increasing the variance. So with increase in max_depth, I speculate the decrease in bias and increase in variance.
From our example, it looks like the speculation is somewhat correct with some marginal noise.



1.2. :

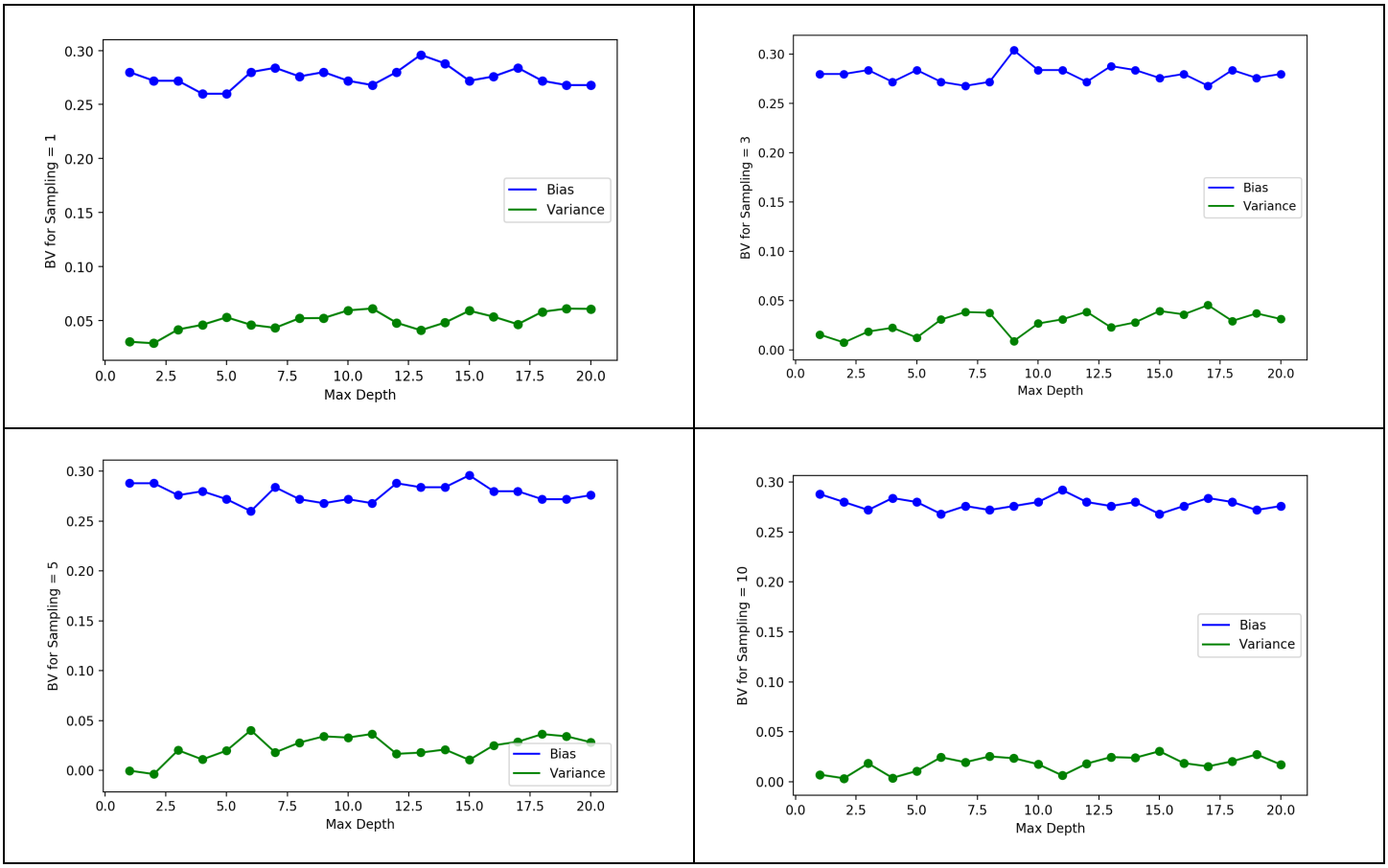
Accuracy initially increases as the max_depth increases and then tends to decrease as the classifier starts to overfit the data.

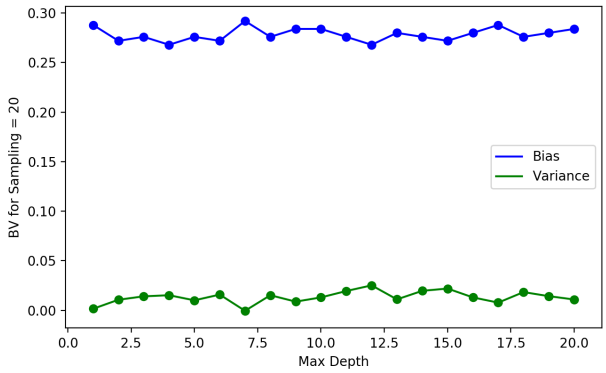
However, with increase in samplings, ensemble fights overfitting, and accuracy stays more or less constant. (*samplings = 20* line)



Bias and variance for each case:

It looks like with increase in samplings, ensemble is able to fight the increase in variance. Hence, with increase in samplings, variance has gone far below 0.05 in the graphs. Otherwise in general, with increase in max_depth, bias has decreased and variance has increased.





2: PAC learning:

When the number of hypotheses is finite, number of samples (m) required for error(ε) and confidence ($1 - \delta$) is given by

$$m \geq \frac{1}{\varepsilon}(ln|H| + ln(1/\delta))$$

Given ε = 0.03 and δ = 0.01

Number of hypothesis | H | for given data is 10 * 9 * 2^4 = 1440. However this is a loose bound with redundant trees. To calculate number of unique hypotheses consider

Part 1: When ALL data is either + or - **⇒ 2**

Part 2: When ALL data can be divided by only one attribute (+ on one side and - on other) **⇒ 10 * 2**

Part 3: When we have to use 2 attributes, in here we can say that some cases are already covered by part 1 and part 2.

At the leaves All '+' and All '-' are covered in Part 1

All '+' on one side and all '-' on one side is covered in Part 2

Hence ⇒ **(10 Choose 2) * (2^4 - 6) = 450**

By adding all the parts we get: **2 + 20 + 450 = 472 unique Hypotheses**

That gives us: m >= (1/ 0.03) * (6.156 + 4.605) = 358.7

m >= 359 samples.

3: VC Dimension:

Let's start with a hypothesis where

K = 1 , With 3 points it breaks down and hence can only shatter 2 points, so **VC Dimension = 2**

-----[+]------ - --------- + ---

K = 2, It can be seen that this breaks down with 5 points and hence **VC Dimension = 4**

-----[+]---- - ------[+]------ - ------- +

K = 3, It can be seen that this breaks down with 5 points and hence **VC Dimension = 6**

-----[+]---- - ------[+]------ - -------[+]------- - ------- +

Similarly by bounding only + or - points, we can at most shatter 2k points with k intervals.

Hence VC Dimension on a real line with class k is **2k**

4.2 / 4.3:

We get highest accuracy with linear kernel and as complexity increases, it is tending to overfit the data and accuracy is decreasing.

It can be said that Neural network is also a complex classification algorithm and given such less data, it is overfitting and hence less accuracy than any simple mechanisms like ID3 or linear kernel SVMs.

Kernels	**Linear**	**RBF**	**Polynomial**	**Sigmoid**	***Neural Network***
Accuracies	74.0	66.8	36.4	66.8	*67.5*

4.4:

With bagged Decision trees, Maximum accuracies went over 75 and it consistently accuracies at 72 with 20 samplings.

So we can say that, bagged decision trees perform well for the given data set with respect to accuracy.

Bias and Variance:

SVM:

Kernels	**Linear**	**RBF**	**Polynomial**	**Sigmoid**
Bias	0.256	0.332	0.604	0.332
variance	0.0088	0.0	-0.0605	0.0

Bagging (Un pruned trees):

Samplings	**1**	**3**	**5**	**10**	**20**
Bias	0.284	0.268	0.272	0.276	0.272
Variance	0.043	0.0411	0.0314	0.0232	0.0213

While the accuracy is better with bagging. It looks like SVM with linear Kernel is able to fight bias and variance better than bagging ensemble of decision trees.

Difference can be explained with multiple reasoning:

1. With the bagging ensemble, the samplings are correlated. Hence there is a chance that when one tree makes a mistake most of the others are also making the same mistake.
2. Very few data points, which means that the complex algorithms like NN or Polynomial Kernels will easily overfit the data and hence less accuracy.
3. However, I am not able to explain why RBF or Sigmoid Kernels has zero variance and also less accuracy. That seems to suggest that the model is underfitting the data and is not a right one to use in this case.

Notes:

1. pip3 install sklearn : used for all the external library
2. pip3 install decision-tree-id3 : used for decision trees