CS7.301 Machine and Data Learning
Assignment 2 Report

February 21, 2023

1 Introduction

In statistics and ML , the bias-variance tradeoff is the property of a model that the varinace of the parameter estimated across the sample can be reduced by increasing the bias in the estimated parameters. Let's see this image. The total

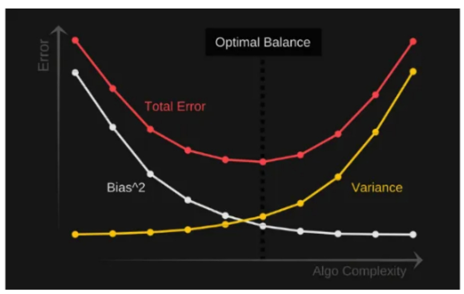

Figure 1: Bias Variance Tradeoff

error is the sum of squared bias and variance and irreducible error.

$$TotalError = Bias^2 + Variance + IrreducibleError$$

It is not that a good model can reduce irreducible error, it is the error that is created by noise in the data.

Underfitting and Overfitting Underfitting is when the model is too simple and does not capture relevant features of the data. Overfitting is when the model is too complex and captures the noise in the data. This could be understood better by the image below.

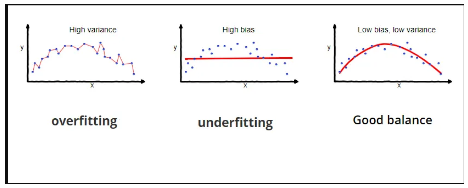

Figure 2: Underfitting and Overfitting

Bias Bias is the difference between the expected value of the parameter and the true/ target value of the predictor model. It is given by the formula. High Bias can cause underfitting of the model.

$$Bias = (E[\hat{f}(x)] - f(x))^2$$

Variance Variance is the variability of the parameter estimated across the sample. Model with high variance pays a lot of attention and does not generalize on the data which it hasn't seen before. This is what we call overfitting. Variance is given by the formula below.

$$Variance = E[(\hat{f}(x) - E[\hat{f}(x)])^2]$$

2 Task 1: Linear Regression

Write a brief about what function the method `LinearRegression().fit()` performs

The method `LinearRegression().fit()` is a function provided by the scikit-learn library in Python that fits a linear regression model to a given dataset.

More specifically, the `fit()` method takes as input a set of input features and corresponding target values, and then trains a linear regression model to map the inputs to the targets. This is done by calculating the coefficients of the linear regression equation that best fits the data, such that the difference between the predicted values and the actual target values is minimized.

Once the model has been trained, it can then be used to make predictions on new input data. This makes the `fit()` method a crucial step in the process of building a machine learning model that can make predictions on new, unseen data.

3 Task 2: Gradient Descent

Explain how gradient descent works to find the coefficients. For simplicity, take the case where there is one independent variable and one dependent variable

Gradient descent is an optimization procedure that is used to discover the best values of a model's coefficients that minimise the difference between the predicted and actual values of the dependent variable. Gradient descent works by iteratively modifying the coefficient values to minimise the cost function until it achieves a minimum.

When only one independent variable and one dependent variable are present, the model may be written as follows:

$$y = b_0 + b_1 * x$$

where y is the dependent variable, x is the independent variable, b_0 is the y-intercept, and b_1 is the slope of the line.

The goal of gradient descent is to identify b_0 and b_1 values that minimise the cost function $J(b_0, b_1)$, which is defined as the sum of squared errors between predicted and actual y values:

$$J(b_0, b_1) = (1/2m) * \Sigma(y_{pred} - y_{actual})^2$$

If m is the number of data points, y_{pred} represents the projected value of y, and y_{actual} represents the actual value of y.

To update the values of b_0 and b_1, gradient descent uses the partial derivatives of the cost function with respect to each of the coefficients:

$$\delta J/\delta b_0 = (1/m) * \Sigma(y_{pred} - y_{actual})$$
$$\delta J/\delta b_1 = (1/m) * \Sigma(y_{pred} - y_{actual}) * x$$

The update rules for the coefficients are:

$$b_0 = b_0 - \alpha * \delta J/\delta b_0$$
$$b_1 = b_1 - \alpha * \delta J/\delta b_1$$

where α is the learning rate, which controls the step size in each iteration. The learning rate is usually set to a small value to ensure that the algorithm converges to the minimum. The algorithm starts with some initial values of b_0 and b_1, and then iteratively updates the coefficients using the above update rules until the cost function reaches a minimum.
The steps involved in the gradient descent algorithm are:

1. Initialize the values of b_0 and b_1 to some random values.
2. Calculate the predicted values of y using the current values of b_0 and b_1.
3. Calculate the cost function $J(b_0, b_1)$using the predicted values and actual values of y.
4. Calculate the partial derivatives of the cost function with respect to b_0 and b_1.
5. Update the values of b_0 and b_1 using the update rules.

6. Repeat steps 25 until the cost function reaches a minimum or a predefined number of iterations is reached.

By following the above steps, the gradient descent algorithm is able to find the optimal values of b_0 and b_1 that minimize the cost function and provide the best fit for the data.

4 Task 3: Calculating Bias and Variance

Degree	BIAS^2	VARIANCE	MSE(TEST ERROR)	IRR
1	0.114267	0.005578	0.119845	-2.775558e-17
2	0.012214	0.001386	0.013600	1.734723e-18
3	0.004839	0.000856	0.005695	-1.734723e-18
4	0.004435	0.000979	0.005414	-1.734723e-18
5	0.004381	0.001100	0.005481	-8.673617e-18
6	0.004386	0.001236	0.005623	-1.734723e-18
7	0.004364	0.001722	0.006086	-8.673617e-19
8	0.004462	0.003008	0.007470	-2.602085e-18
9	0.004530	0.009285	0.013814	-6.938894e-18
10	0.010998	0.120438	0.131436	2.775558e-17
11	0.010168	0.181224	0.191391	-2.775558e-17
12	0.134423	2.581771	2.716194	0.000000e+00
13	0.465318	8.765777	9.231096	-1.776357e-15
14	2.620887	46.346644	48.967531	-7.105427e-15
15	17.009511	357.670171	374.679682	0.000000e+00

Figure 3: Bias and variance

As one can see from the image above, the bias and variance of a model are inversely related. Also at degree 14 and 15 the error is high because of overfitting. Overfitting is also accompanied by high variance. Let us see the following plots to see how degree of the polynomial affects the bias and variance and eventually our prediction. The above above shoe how the test data is not fitting the model for lower degrees 1 and 2 As the degree of the polynomial function increases, the model becomes more complex and is better able to fit the training data. However, this can also lead to overfitting, where the model becomes too

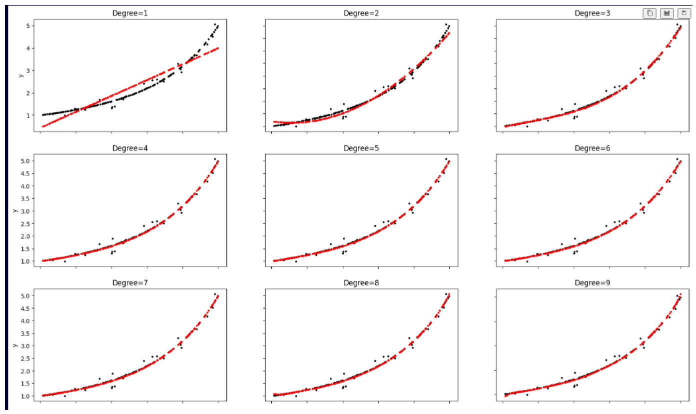

Figure 4: Bias and variance plot

specialized to the training data and fails to Underfitting is also accompanied by high bias. The models with 1 degree and 2 are underfitting and the error is high. generalize well to new data. This can cause the irreducible error to increase as

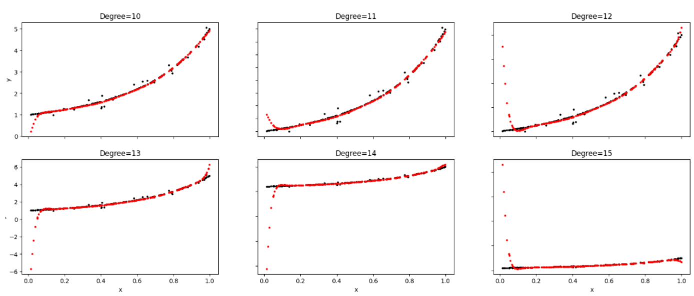

Figure 5: Bias and variance plot

the degree of the polynomial function increases. But eventually model starts overfitting and the error increases. The Bias Variance Tradeoff is a fundamental concept in machine learning. It is the idea that a model can only be optimized for either low bias or low variance, but not both. The Graph below is plotted for the same data and different degrees od polynomial function.

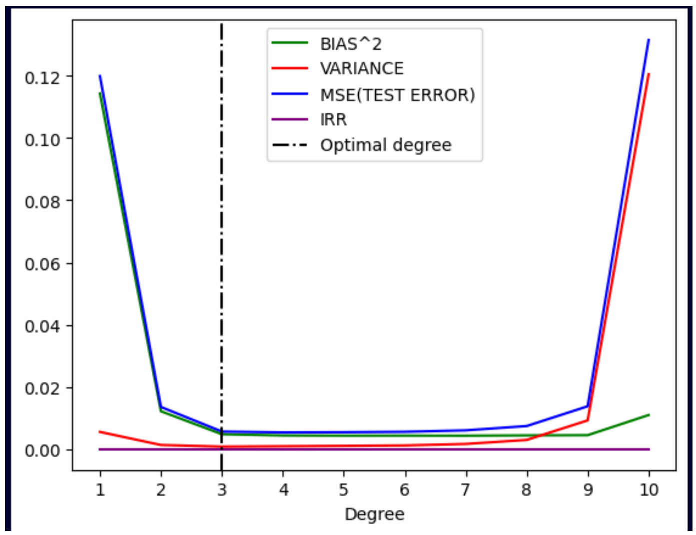

Figure 6: Bias and variance plot

This graph shows that as the degree of the polynomial function increases, the model becomes more complex and is better able to fit the training data. However, this can also lead to overfitting, where the model becomes too specialized to the training data and fails to generalize well to new data. This can cause the irreducible error to increase as the degree of the polynomial function increases. The optimal model is of degree 3 according to the graph.

5 Task 4: Bonus

As we see that the equation is not a linear equation, it has an exponent term in it

$$Q = CV_0 e^{-t/RC}$$

We can use the log function to convert it into a linear equation

$$log(Q) = log(CV_0) - t/RC$$

Now we can use the linear regression model to fit the data. Now the intercept

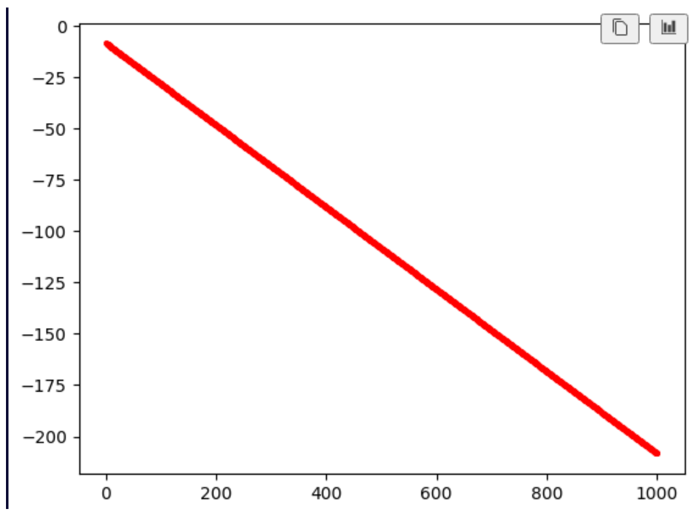

Figure 7: Bias and variance plot

value is $log(CV_0)$ and the slope is $-1/RC$ therefore the equation becomes value of C can be found out and is $4.999999999999631 * 10^(- 5)$ The value of R can be found out and is $1.0000000000000744 * 10^5$