Bias Variance Tradeoff in Hypothesis Classes

Alankrit Singh, Rohan Jha
Yash Ahirrao

April 2024

1 The Problem

In the context of Machine Learning, one particular consistent observation is the Bias-Variance trade-off. Different models have different extents of bias and variances, which affect the fundamentals of our machine learning model. It affects our train-test split, tells us if we are under-fitting or over-fitting, how many features to select and even what hypothesis class to use. A hypothesis class that is too simple might not capture the underlying pattern in the data (high bias), whereas a class that is too complex might fit the noise instead of the ground truth function (high variance). Thus, balancing bias and variance is crucial for optimal model performance. The main goal of this project is to see how varying complexity affects performance in terms of the bias-variance trade-off. We aim to systematically explore how different complexities within a range of hypothesis classes affect the predictive accuracy of machine learning models. where MSE: Mean-Squared Error

2 Methodology

We picked a polynomial of degree 6 as our ground truth function and added noise to it. Then, we took a sample from it and split it into training and testing data. Firstly, we plotted the best-fit polynomials to the training data obtained from degrees 1 through 11, along with our training and testing data. This allowed us to obtain the errors for polynomials of various complexities on our degree 6 polynomial. We also plotted the curve of MSE(test) vs Complexity and MSE(train) vs Complexity where MSE(test) is MSE over testing data and MSE(train) is MSE over training data. Similar method was followed for Neural Networks and Decision Trees.

3 Experimental Results

3.1 Dataset

The ground truth function used to generate the dataset is: $x^6 - x^5 + 2x^4 + 1x^3 + 4x^2 + 2x - 2 + e$
where e is noise. We generated the dataset in a Python notebook using NumPy. Scikit-Learn and Matplotlib.

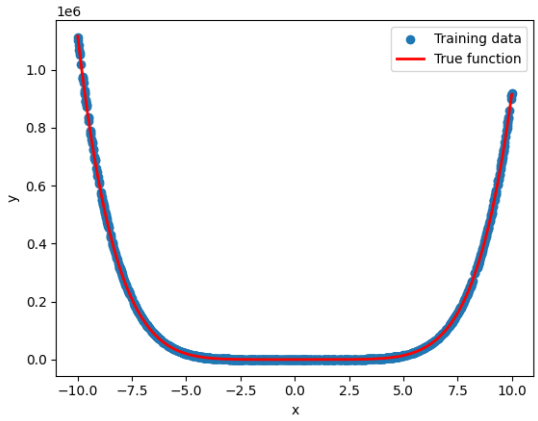


Figure 1: Dataset

We will first test our data on linear and higher-degree polynomial models.

3.2 Testing on Polynomial Models

3.2.1 The Curves of Best Fit Polynomial Models

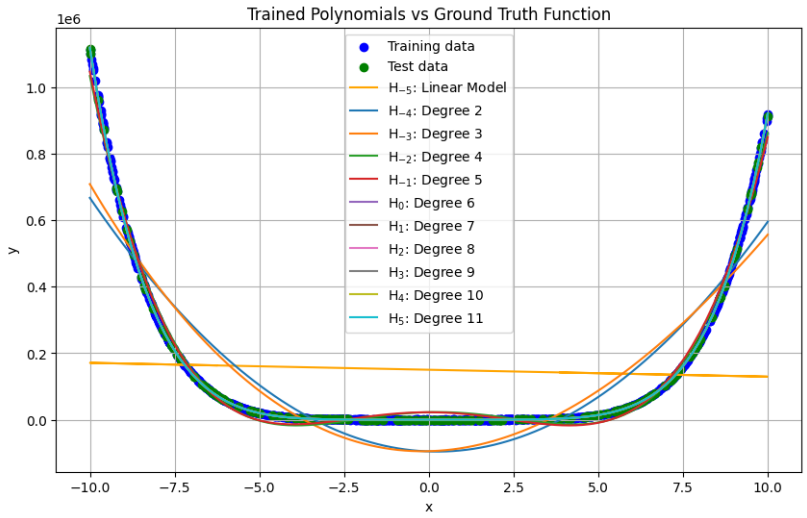


Figure 2: Fitting Trained Polynomials

As can be seen, H_{-5} doesn't do a good job at all, as it cannot cope with the complexity of our degree 6 polynomial. H_{-4} and H_{-3} being quadratic and cubic, are starting to take the shape of our ground function but still not good predictors at all. H_{-2} is where our model from this zoomed-out looks "somewhat good", but a closer look tells us that is not true. We will look further into this in the next subsection.

3.2.2 Error vs Complexity

As can be seen in the table, the training MSE decreases monotonically, while the test MSE decreases till H_0 (degree = 6) and then increases and varies, but it's never better than H_0.

Complexity (Polynomial Degree)	Training MSE	Test MSE
1	5.91e+10	5.99e+10
2	1.13e+10	1.26e+10
3	1.10e+10	1.25e+10
4	3.81e+08	4.06e+08
5	3.67e+08	3.83e+08
6	9.20e+03	1.62e+02
7	9.19e+03	1.62e+02
8	9.19e+03	1.68e+02
9	9.19e+03	1.78e+02
10	9.17e+03	1.66e+02
11	9.17e+03	1.83e+02

We can say H_{-5} through H_{-1} heavily under-fit the data while H_1 through H_5 slightly over-fit the data while H_0 is the best model which we knew already because it is of the same complexity as our ground-truth function.

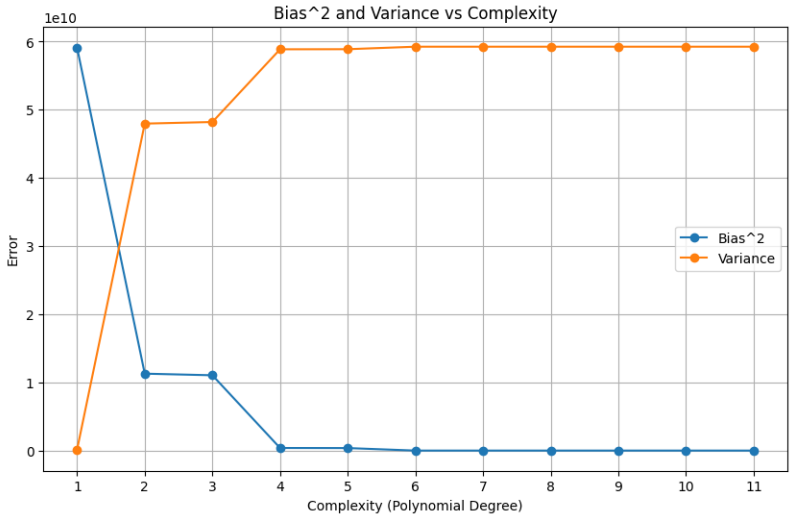


Figure 3: Bias-Variance Curve vs Complexity

3.3 Testing on Decision Trees

For Decision Trees, one way to vary complexity is to change the maximum depth of the tree. As you increase the depth of the tree, it becomes more complex and can represent more intricate decision boundaries in the feature space. This increased complexity can help reduce bias, allowing the decision tree to capture more detailed patterns in the data. However, deeper trees are more prone to overfitting. As the tree grows deeper, it becomes increasingly tailored to the training data, potentially capturing noise or outliers as part of the learned decision boundaries. This leads to high variance, as the model may not generalize well to unseen data.

We used NumPy, Matplotlib and SciKit-Learn to train decision trees on a random sample obtained from our ground truth function. Then, we plotted the $Bias^2$ and Variance vs Complexity graphs, where we used the Maximum depth of the tree as the measure for complexity.

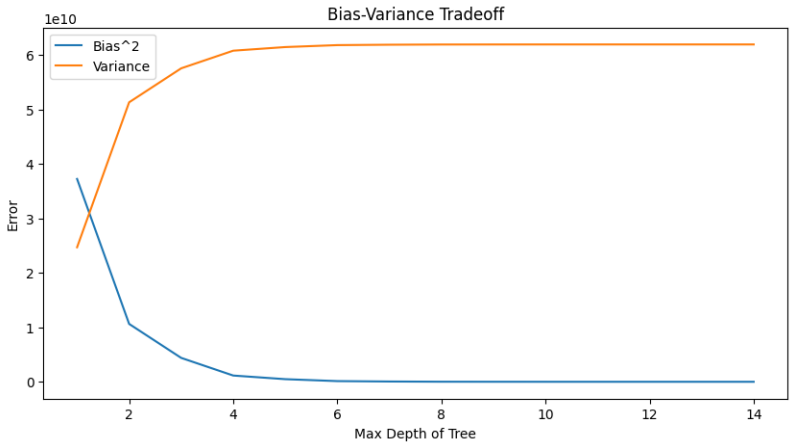


Figure 4: Bias-Variance Curve vs Complexity

It is obvious from the graph that $Bias^2$ decreases with increasing complexity while Variance increases. Further, the following table gives us a precise look at what's happening.

	Max Depth	MSE (Train)	MSE (Test)
0	1	3.724011e+10	4.156037e+10
1	2	1.062243e+10	1.065019e+10
2	3	4.386108e+09	4.686844e+09
3	4	1.146989e+09	1.118288e+09
4	5	4.805855e+08	5.657601e+08
5	6	1.237637e+08	1.627896e+08
6	7	5.030327e+07	6.236260e+07
7	8	1.297852e+07	2.208883e+07
8	9	5.236254e+06	1.239611e+07
9	10	1.421084e+06	9.888868e+06
10	11	4.509839e+05	9.123044e+06
11	12	1.450740e+05	8.930466e+06
12	13	4.099451e+04	8.861505e+06
13	14	1.004901e+04	8.834743e+06

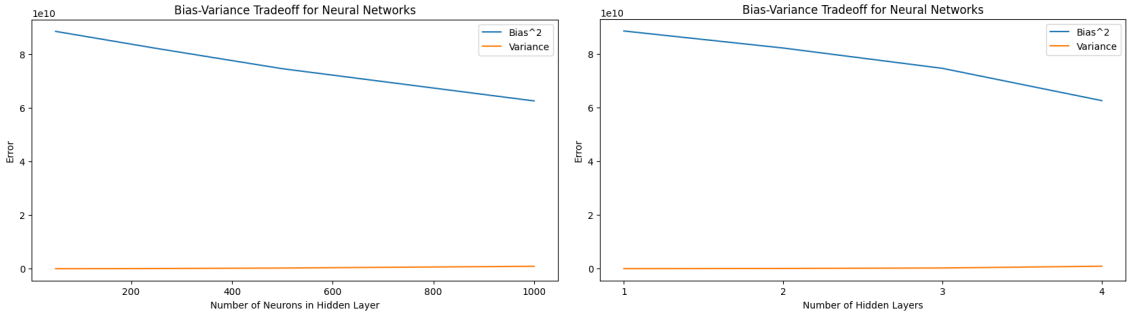
Here, the MSE(train) and the MSE(test) decrease with an increase in Maximum Depth.

3.4 Testing on Neural Networks

For Neural Networks, complexity can vary by either changing the number of neurons in hidden layers or increasing the number of hidden layers. Adding more neurons increases the capacity of the neural network to learn complex patterns in the data. This can potentially reduce bias since the model becomes more flexible and can capture more intricate relationships in the data. However, increasing the number of neurons also increases the risk of overfitting. This leads to high variance in the model's predictions, and it may perform well on the training data but poorly on unseen data.

The number of hidden layers in a neural network also influences the bias-variance tradeoff. Adding more hidden layers increases the depth of the neural network, allowing it to learn hierarchical representations of the data. Deeper networks are also more prone to overfitting, especially if they are not properly regularized. With more layers, the model has more parameters, increasing the risk of memorizing noise in the training data and thus leading to higher variance.

We used NumPy, Matplotlib and SciKit-Learn to train Neural Networks on a random sample obtained from our ground truth function. Then, we plotted the $Bias^2$ and Variance vs Complexity graphs, where we used the number of neurons and the number of hidden layers as the measure for complexity.



(a) On Increasing Neurons

(b) On Increasing Hidden Layers

Figure 5: Bias-Variance Curve vs Complexity

As can be seen from the two graphs, $Bias^2$ decreases with increasing complexity while Variance increases. Further, the following table gives us a precise look at what's happening.

	Hidden Layer Size	MSE (Train)	MSE (Test)
0	50	7.961713e+10	8.508854e+10
1	250	7.375113e+10	7.883896e+10
2	500	6.682427e+10	7.150008e+10
3	1000	5.592024e+10	5.985539e+10

Clearly, the MSE(train) and the MSE(test) decrease with an increase in the size of the hidden layers that govern the Neural Network.

4 Conclusion

In conclusion, our investigation into the effect of increasing complexity on the bias-variance curve reveals valuable insights into the trade-off between model simplicity and performance. We observe a characteristic pattern as complexity rises: bias typically decreases while the variance increases. As complexity increases, the model becomes more flexible and can capture more intricate patterns, thus reducing bias. However, it also becomes sensitive to small fluctuations in the training data, leading to high variance and poor generalization of unseen data.

Understanding this relationship is crucial for effective model selection and regularization techniques, ensuring optimal balance between bias and variance for robust and generalizable machine learning models. Further research could explore advanced regularization methods or alternative model architectures to mitigate the adverse effects of increasing complexity, thereby enhancing predictive performance in real-world applications.