Project 1 FYS-STK4155

Reinert Sæbø Eldøy

October 11, 2021

Exercise 1

We are tasked with applying Ordinary Least Squares to the Franke Function. I use the code example provided in the project pdf to visualize the function(see figures below), and will operate with gaussian noise centered at 0 with standard deviation $\sigma = 0.1$. The 20 (x, y) datapoints are uniformly sampled from 0 to 1 and I use the PolynomialFeatures method from scikit-learn to build design matrices. I also employ standard scaling of the predictors(subtracting the mean and dividing by the standard deviation) as it provides the clearest performance delineation between the different polynomial fits. The MSEs and R2 scores are

```
MSE for deg=1 polynomial:  0.04820281617529043
R2 score for deg=1 polynomial:  0.539437339256952
MSE for deg=2 polynomial:  0.09147807448621305
R2 score for deg=2 polynomial:  0.3599027012522056
MSE for deg=3 polynomial:  0.011074184883117325
R2 score for deg=3 polynomial:  0.8904889770818888
MSE for deg=4 polynomial:  0.006518623103414819
R2 score for deg=4 polynomial:  -0.11874175914782148
MSE for deg=5 polynomial:  0.006227642791600526
R2 score for deg=5 polynomial:  0.9495083915566792
```

from which we infer that a 5th degree polynomial fit performs best as its MSE is lowest and its R2 score is closest to 1, which means the model is readily generalizable to unseen data.

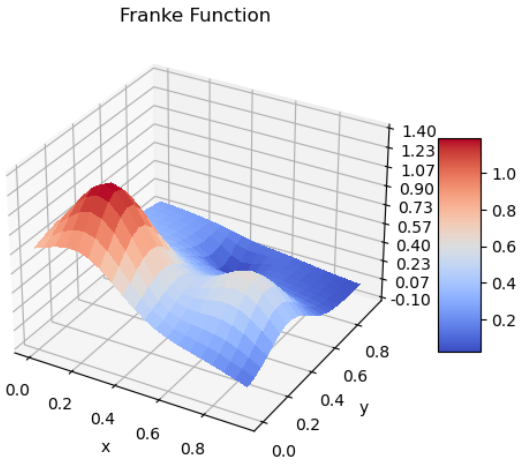

Figure 1: The Franke Function visualized.

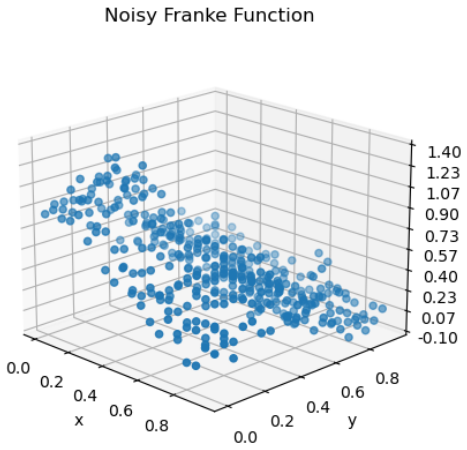

Figure 2: Franke Function with added noise. This is the data we'll fit against.

Exercise 2

Given $\mathbf{y} = f(\mathbf{x}) + \epsilon$, the bias-variance tradeoff is derived as follows[1]:

$$\mathbb{E}\left[(\mathbf{y} - \tilde{\mathbf{y}})^2\right] = \mathbb{E}\left[(\mathbf{f} + \epsilon - \tilde{\mathbf{y}})^2\right] = \mathbb{E}$$

The first term represents the bias, i.e. the average squared distance between the estimator's mean value $\mathbb{E}[\tilde{\mathbf{y}}]$ and the actual data points f_i. In a sense it's an artifact of our assumptions about the model. For instance a linear estimator will exhibit large bias when the true relationship is more cubic in character. The second term is the variance, which as the name implies represents the overall variability of the estimator, or more precisely its average deviation from the expectation value. The final term is known as the "irreducible error" and is defined as the variance of the noise, $\sigma^2 = Var[\epsilon]$.

In the code i increase to up to 9 degree polynomial fits and compute the MSE for each model, predicting on both the test set and the training set, and plot them together like figure 2.11 in Hastie et. al. for three different datapoint numbers. The result is shown below.

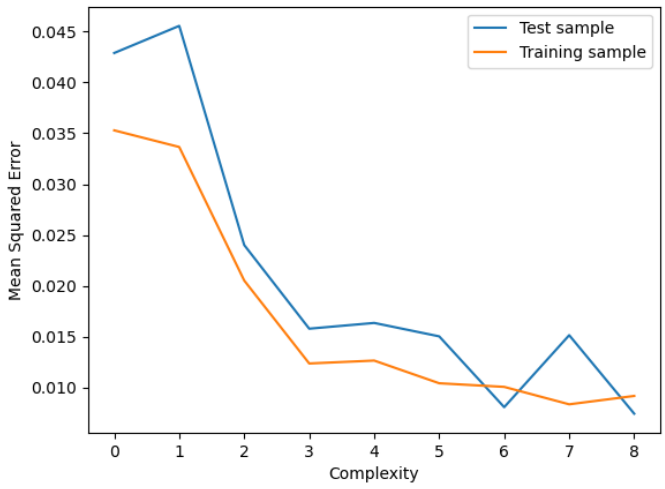

Figure 3: 200 datapoints

[1] Writing f instead of $f(\mathbf{x})$

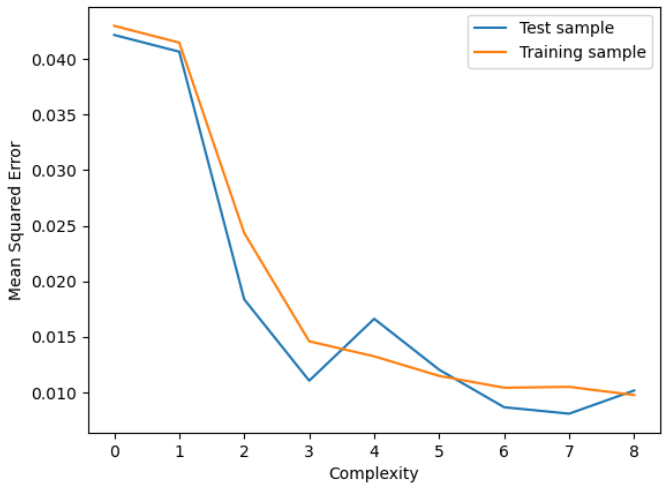

Figure 4: 300 datapoints

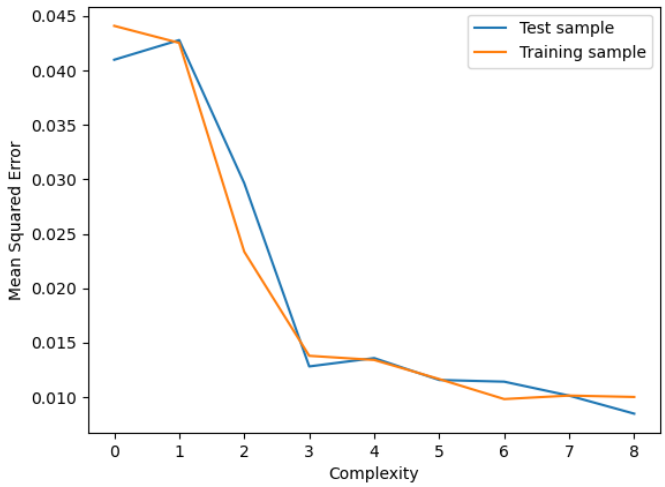

Figure 5: 500 datapoints

The immediate takeaway is that predictions on test samples tend to yield greater errors than training samples, which is due to it generally having a greater irreducible error. Furthermore the test sample in this case exhibits somewhat more erratic behaviour, dipping up and down here and there, while the training

sample's MSE is pretty much monotonically decreasing. The reason for this is that the test sample consists of "unseen data" which might take a different form than the training set and its curve is therefore more sensitive to increased bias with complexity. We also see that the curves get closer together the higher number of datapoints we include. This is because increased datapoints means their respective irreducible errors are closer together.

Exercise 3

Exercise 4

Exercise 5

Exercise 6