FYS-STK Exercises week 38

We generate a data set of n=100 similar to that from week 36 to investigate the bias-variance tradeoff. The input data has polynomial features of degree 1 to 15, i.e., increasing complexity. Then we run an ordinary least squares (OLS) model on the data for with 100 resamplings for each polynomial degree, and estimate error, bias and variance of our model.

The following code is mostly made from scratch, with the bias, variance and mse calculations taken from the course book (Hjort-Jensen 2021).

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.utils import resample
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

def design_poly_n(x, n):
    X = np.zeros((len(x), n))
    X[:,0] = 1
    for i in range(1, n):
        X[:,i] = (x**i).T
    return X

np.random.seed(3490)
n = 100
# Make data set.
x = np.linspace(-3, 3, n).reshape(-1, 1)
y = np.exp(-x**2) + 1.5 * np.exp(-(x-2)**2)+ np.random.normal(0, 0.1, x.shape)

np.random.seed(5)
npoly = 15
X = design_poly_n(x, npoly + 1)[:,1:]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

```
polys = np.arange(npoly) + 1
mse = np.zeros(npoly)
bias = np.zeros(npoly)
variance = np.zeros(npoly)
nbootstraps = 100

# fit models with resampling

for i in polys:
    y_pred = np.zeros((y_test.shape[0], nbootstraps))
    X_train_deg = X_train[:,:i]
    for j in range(nbootstraps):
        x_tmp, y_tmp = resample(X_train_deg, y_train)
        lr = LinearRegression(fit_intercept=False).fit(x_tmp, y_tmp)
        y_pred[:,j] = lr.predict(X_test[:,:i]).ravel()
    mse[i-1] = np.mean( np.mean((y_test - y_pred)**2, axis = 1, keepdims = True) )
    bias[i-1] = np.mean((y_test - np.mean(y_pred, axis = 1, keepdims = True))**2)
    variance[i-1] = np.mean(np.var(y_pred, axis = 0, keepdims = True))

plt.clf()
plt.plot(polys, mse, label = "Error")
plt.plot(polys, bias, label = "Bias")
plt.plot(polys, variance, label = "Variance")
plt.legend()
plt.xlabel("Polynomial degrees")
plt.ylabel("Value")
plt.show()
```

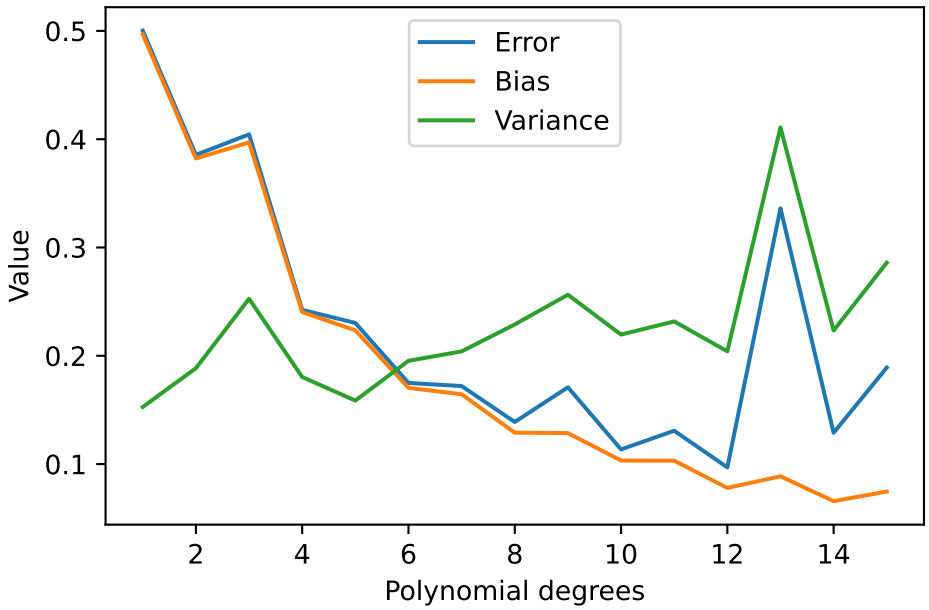

Figure 1: Error, bias and variance for a polynomial fit up to degree 15. The training data was bootstrapped with 100 resamplings per polynomial degree.

From Figure 1 we see that bias decreases with model complexity while variance increases, and that the error has some optimal complexity around degree 10-12. A more complex model will generally have lower bias, but is very sensitive to changes in the data used to fit the model, resulting in a higher model variance.

Hjort-Jensen, Morten. 2021. *Applied Data Analysis and Machine Learning.* Jupyter Notebook.