hw01-selecting_and_fitting_models

January 14, 2019

1a

A flexible method here would perform better than an inflexible method. A flexible method will fit the data better with the large sample size. In addition, small number of predictors will limit model variance.

1b

A flexible method here would perform worse than an inflexible method. Since the number of predicictors is extremely large, using a flexible method might overfit the small number of observations.

1c

A flexible method would perform better than an inflexible method here. Since inflexible methods can only produce a relatively small range of shapes, a flexible moethod would fit better when predictors and response is highly non-linear with less restrictive shape.

1d

A flexible method here would perform worse than an inflexible method because the variance of the error is extremely high and a flexible method would fit to the error terms as well and increase the variance.

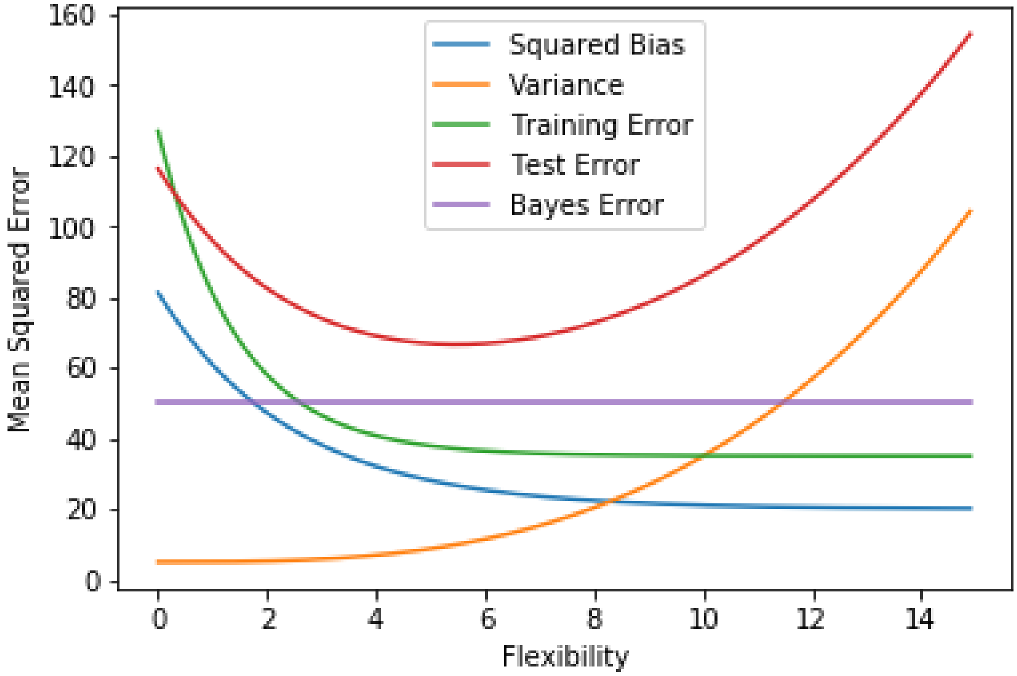
```
In [208]: #2a
          import matplotlib.pyplot as plt
          import numpy as np

          x = np.arange(0, 15, 0.1)

          y1 = 50 / 1.5 ** (x - 0.5) + 20
          y2 = 0.03 * x ** 3 + 5
          y3 = 65 / 2 ** (x - 0.5) + 35
          y4 = 50 / 1.5 ** (x - 0.5) + 20 + 0.03 * x ** 3 + 5 + 30
          y5 = x * 0 + 50

          plt.plot(x, y1, label = 'Squared Bias')
          plt.plot(x, y2, label = 'Variance')
          plt.plot(x, y3, label = 'Training Error')
          plt.plot(x, y4, label = 'Test Error')
          plt.plot(x, y5, label = 'Bayes Error')

          plt.xlabel('Flexibility')
          plt.ylabel('Mean Squared Error')
plt.legend()
plt.show()
```



2b

(1) Bias refers to the error that “is introduced by approximating a real-life problem” (ISLR, p35). It means that the more complicated model we use, the less bias it will introduce because the shape of the fit is less strict. Therefore, in the above plot, the blue line shows the Squared Bias curve: when flexibility increases, bias will decrease.

(2) Variance refers to “the amount by which f would change if we estimated it using a different training data set” (ISLR, p34). Therefore, if a method is highly flexible, then changing data points will result in large changes in f. In above plot, the orange line shows the variance curve: when flexibility increases, variance will increase as well.

(3) As the method becomes more and more complex, it will try to fit more and more data and hence reducing the training error. In the plot, the green line shows the training error curve: as the flexibility increases, the training error will decrease.

(4) The test error should appear as a U-shape because there is a trade-off between variance and bias. When the method is inflexible, the bias is large while variance is small. When the method becomes more flexible, the bias will decrease while variance will increase.

(5) The irreducible error, as the name implies, does not change when flexibility changes because it only has to do with the data but not with the models that we choose. Therefore, it appears as a line parallel to the x-axes.

```
In [209]: #3a
          np.random.seed(251)

          #3b
          x1 = np.random.uniform(-1,1,200)
          x2 = np.random.uniform(-1,1,200)

          #3c
          mu = 0
          sigma = 0.5
          e = np.random.normal(mu, sigma, 200)
          y = x1 + x1*x1 + x2 + x2*x2 + e

          #3d
          prob_suc = np.exp(y) / (1 + np.exp(y))

          #3e
          mask_suc = prob_suc > 0.5
          plt.scatter(x1[mask_suc],x2[mask_suc], label = 'success')
          mask_fail = prob_suc <= 0.5
          plt.scatter(x1[mask_fail], x2[mask_fail], label = 'failure')
          plt.legend()

          #3f
          X, Y = np.meshgrid(np.linspace(-1, 1, 200), np.linspace(-1, 1, 200))
          yy = X + X*X + Y + Y*Y
          prob = np.exp(yy)/(1 + np.exp(yy))
          plt.contour(X, Y, (prob>0.5).astype(np.int), linestyles = 'dashed')

          #3g
          plt.xlabel('x1')
          plt.ylabel('x2')
          plt.title('Illustration of the Bayes classifier')
          plt.show()
```

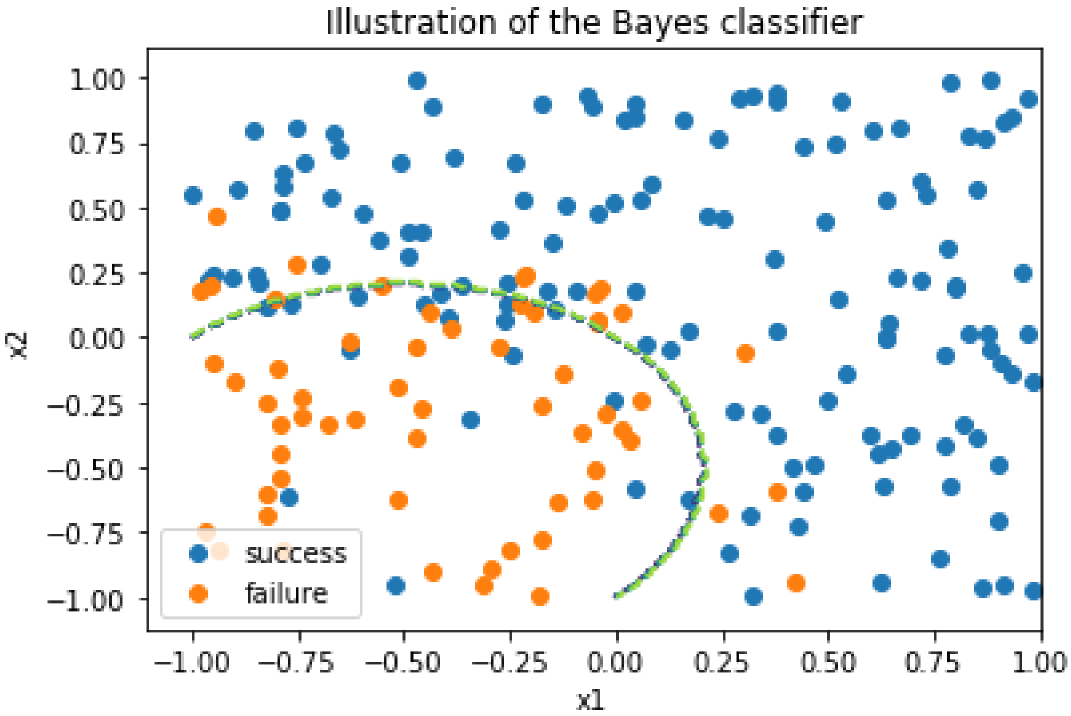


Illustration of the Bayes classifier
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
x2
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
x1
success
failure