An Unknown Signal Report

George Herbert
cj19328@bristol.ac.uk

March 25, 2021

Abstract

This report demonstrates my understanding of the methods I have used, the results results I have obtained and my understanding of issues such as overfitting for the 'An Unknown Signal' coursework.

1 Equations for linear regression

For a set of points that lie along a line with Gaussian noise $\mathbf{y} = \mathbf{X}\mathbf{w} + \epsilon$ where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, the maximum likelihood esimation is equivalent to the least square error estimation and is given by the equation:

$$\hat{\mathbf{w}} = (\mathbf{X}^\mathbf{T}\mathbf{X})^{-1}\mathbf{X}^\mathbf{T}\mathbf{y}.$$

I've implemented this equation in my code as the following to calculate the maximum likelihood estimation for my training data:

```
def regressionNormalEquation(self, X, y):
    ws = np.linalg.inv(X.T @ X) @ X.T @ y
    return ws
```

2 Choice of polynomial order

3 Choice of unknown function

4 Overfitting

Overfitting occurs when an algorithm produces a model that has learnt the noise in the data as if it represents the structure of the underlying model.

In the case of linear regeression, overfitting typically occurs when the model produced contains too complex a function class, such that it would fail to predict future observations.

Cross-validation can be used to detect overfitting. We can partition our data into two subsets: a training set and a validation set;

To prevent overfitting, I have used k-fold cross-validation for each line segment, the process for which is as follows:

1. Shuffle the dataset
2. Split the dataset provided into k equally sized subsamples
3. Perform cross-validation k times, using each subsample exactly once as the validation data

5 Procedure for determining function

6 Testing