

1 Linear Regression

2 Introduction

Maximum Likelihood Estimation and Prediction

In this code, we use maximum likelihood estimation to find the best parameters for a linear model, and use those parameters to make predictions on a test set.

Training Set

First, we define our training set as arrays of inputs and outputs. The inputs are a 5x1 vector, and the outputs are a 5x1 vector. These are stored in the variables `X` and `y`, respectively.

```
X = np.array([-3, -1, 0, 1, 3]).reshape(-1,1) # 5x1 vector, N=5, D=1
y = np.array([-1.2, -0.7, 0.14, 0.67, 1.67]).reshape(-1,1) # 5x1 vector
```

Maximum Likelihood Estimation

Next, we define a function `maxlikestimate` that takes in the training inputs and outputs, and returns the maximum likelihood parameters for our linear model. We use the normal equation to estimate the parameters, which is implemented using the `solve` function from the `linalg` module in `scipy`.

```
def max_lik_estimate(X, y):

    bash
    Copy code
    # X: N x D matrix of training inputs
    # y: N x 1 vector of training targets/observations
    # returns: maximum likelihood parameters (D x 1)

    N, D = X.shape

    # Estimate the parameters using the normal equation
    theta_ml = np.linalg.solve(X.T @ X, X.T @ y)

    return theta_ml
```

Test Set

We define a test set as a 100x1 vector of test inputs, stored in the variable `Xtest`.

```
Xtest = np.linspace(-5,5,100).reshape(-1,1) # 100 x 1 vector of test inputs
```

Prediction

We define a function `predict_with_estimate` that takes in a set of test inputs and the maximum likelihood parameters, and returns the predicted outputs for those test inputs. The prediction is simply the dot product of the test inputs and the parameters.

```
def predict_with_estimate(Xtest, theta):  
  
    bash  
    Copy code  
    # Xtest: K x D matrix of test inputs  
    # theta: D x 1 vector of parameters  
    # returns: prediction of  $f(Xtest)$ ; K x 1 vector  
  
    prediction = Xtest @ theta  
  
    return prediction
```

We use this function to make predictions on our test set, using the maximum likelihood parameters we estimated earlier. We store these predictions in the variable `ml_prediction`.

```
ml_prediction = predict_with_estimate(Xtest, theta_ml)
```

Plotting

Finally, we plot our training data as points, and plot the predicted outputs for our test inputs as a line. We label our axes and add a legend.

```
plt.figure()  
plt.plot(X, y, '+', markersize=10)  
plt.plot(Xtest, ml_prediction)  
plt.xlabel("$x$")  
plt.ylabel("$y$")  
plt.legend(["Training Data", "Maximum Likelihood Prediction"])
```