LA_RubanrajRavichandran_180417_02_Exercise2_MaxL

April 22, 2018

1 Team members:

1.1 Ramesh kumar

1.2 Ravikiran Bhat

1.3 Rubanraj Ravichandran

1.4 Mohammad Wasil

1.5 Task1

Implement in Python (you can use SciPy library) the Maximum Likelihood Estimator to estimate the parameters for example mean and variance of some data. Your steps are: * Create a data set: - Set x-values for example: *x = np.linspace(0, 100, num=100)*, - Set observed y-values using a known slope (1.4), intercept (4), and sd (3), for example *y = 4 + 1.4x + np.random.normal(0, 3, 100)* * Create a likelihood function which arguments is a list of initial parameters * Test this function on various data sets (Hint: you can use minimize from scipy.optimize and scipy.stats to compute the negative log-likelihood)

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        %matplotlib inline
        import scipy
        from scipy import stats
        from scipy.optimize import minimize
```

```
In [26]: x = np.linspace(0, 100, num=100)
         y = 4 + 1.4*x + np.random.normal(0, 3, 100)

         print "Given slope 1.4"
         print "Given intercept 4"
         print "Given standard deviation 3"
         print "----------------------------------"
         def likelihood_function(params):
             intercept = params[0]
             slope = params[1]
             predicted_mean = intercept + slope*x
             std = params[2]
```

```
            log_likelihood = -np.sum( stats.norm.logpdf(y, predicted_mean, std) )
            return(log_likelihood)

        init_params = [1, 1, 1]
        results = minimize(likelihood_function, init_params, method='nelder-mead')
        print "Prredicted slope using likelihood function " + str(results.x[1])
        print "Prredicted intercept likelihood function  " + str(results.x[0])
        print "Prredicted standard deviation likelihood function " + str(results.x[2])
        print "---------------------------------"
```

```
Given slope 1.4
Given intercept 4
Given standard deviation 3
---------------------------------
Prredicted slope using likelihood function 1.3834785529725473
Prredicted intercept likelihood function  4.6386866214471265
Prredicted standard deviation likelihood function 3.0274974100796475
---------------------------------
```

In [27]:
```
y_prediction = results.x[0] + results.x[1]*x
```

In [29]:
```
plt.title("observed and prediction comparison")
plt.plot(x, y, color='red', label="Observed with noise")
plt.plot(x, y_prediction, color='blue', label="Prediciton")
plt.legend()
plt.show()
```