02_Exercise2_MaxL

April 22, 2018

0.1 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 [55]: import numpy as np
         import scipy as sp
         import matplotlib.pyplot as plt
         import pandas as pd

In [68]: noise = np.random.normal(0, 3, 100)
         noise_mean, noise_variance = sp.stats.norm.fit(noise)

         def log_likelihood(params):
             (y_intercept, slope) = params

             y_mean = y_intercept + (slope * x) + noise_mean
             log_likelihood = -np.sum(np.log(sp.stats.norm.pdf(y, loc=y_mean, scale=noise_varian

             return(log_likelihood)

         x = np.linspace(0, 100, num=100)
         y = 4 + (1.4 * x) + noise

         [y_intercept1, slope1] = sp.optimize.minimize(log_likelihood, (1, 1)).x

         x = np.linspace(0, 100, num=100)
         y = 1 + ( 0.5 * x) + noise

         [y_intercept2, slope2] = sp.optimize.minimize(log_likelihood, (1, 1)).x

         data = {'Y Intercept': ['4', '1'],
         'Y Intercept (calc)': [y_intercept1, y_intercept2],
```

```
        'Slope': ['1.4', '0.5'],
        'Slope (calc)': [slope1, slope2]}

        pd.DataFrame(data = data)
```

Out[68]:

	Slope	Slope (calc)	Y Intercept	Y Intercept (calc)
0	1.4	1.400469	4	3.976573
1	0.5	0.500469	1	0.976573