Assignment 4 - MAD

Mikkel Willén

November 2021

Exercise 1

Vi finder $\hat{\theta} \in (0,1)$, som maksimerer sandsynligheden for den observerede data.

$$\underset{\theta}{\text{maximize}} p\left(x_1, \ldots, x_N; \theta\right) = \underset{\theta}{\text{maximize}} \prod_{n=1}^{N} p\left(x_n; \theta\right) = \underset{\theta}{\text{maximize}} \prod_{n=1}^{N} (1-\theta)^{1-x_n} \theta$$

Så tager vi logaritmen af dette

$$I(\theta) = \log L(\theta) = \log \left(\prod_{n=1}^{N} (1-\theta)^{1-x_n} \theta \right) = \sum_{n=1}^{N} (x_n - 1) \log(1-\theta) + \log(\theta)$$

Så differentiere vi funktionen

$$\frac{\partial I}{\partial \theta} = \sum_{n=1}^{N} \frac{1}{\theta} - \frac{x_n - 1}{1-\theta}$$

Dette sætter vi så lig med nul og får

$$\sum_{n=1}^{N} \frac{1}{\theta} - \frac{x_n - 1}{1-\theta} = 0$$

$$\Leftrightarrow \quad \frac{1}{\theta} = \sum_{n=1}^{N} (x_n - 1) \frac{1}{1-\theta}$$

$$\Leftrightarrow \quad \frac{1}{\theta} - 1 = \sum_{n=1}^{N} (x_n - 1)$$

$$\Leftrightarrow \quad \theta = \frac{1}{\sum_{n=1}^{N} (x_n - 1)}$$

$$\Leftrightarrow \quad \theta_n = \frac{n}{\sum_{n=1}^{N} (x_n - 1)}$$

Siden

$$\frac{\partial^2 I}{\partial qq} = -\frac{\sum_{n=1}^{N} -1}{(1-\theta)^2} - \frac{1}{q^2} < 0$$

er det også et globalt maksimum.

Exercise 2

a)

Vi har PDF for fordelingen af stjerner set gennem vinduet:

$$f\theta(x,y) = \begin{cases} c & \text{if } x_{\min} \leq x \leq x_{\max} \text{ and } y_{\min} \leq y \leq y_{\max} \ 0 \\ \text{otherwise} \end{cases}$$

Vi kan nu udregner c, da vi ved, at når punkterne er IDD, så er:

$$P\left(x_{\min} \leq x \leq x_{\max}, y_{\min} \leq y \leq y_{\max}\right) = 1$$

$$\begin{aligned}
P\left(x_{\min} \leq x \leq x_{\max}, y_{\min} \leq y \leq y_{\max}\right) = 1 &= \int_{x_{\min}}^{x_{\max}} \int_{y_{\min}}^{y_{\max}} c dx dy \\
&= \int_{x_{\min}}^{x_{\max}} [c; y] \, {y_{\min}}^{y_{\max}} dx \\
&= \int_{x_{\min}}^{x_{\max}} c(y_{\max} - y_{\min}) dx \\
&= [c(y_{\max} - y_{\min})x]_{x_{\min}}^{x_{\max}} \\
&= c(y_{\max} - y_{\min})(x_{\max} - x_{\min}) \\
c &= \frac{1}{(y_{\max} - y_{\min})(x_{\max} - x_{\min})}
\end{aligned}$$

b)

Vi udregner sandsynligheden for de to set af parametre $\theta_1 = (-1, 4, -1, 3)$ og $\theta_2 = (-2, 5, -3, 6)$, ved først at udregne sandsynligheden for at se en stjerne.

$$c_{\theta_1} = \frac{1}{(3 - (-1)(4 - (-1))} = \frac{1}{20}$$

$$c_{\theta_2} = \frac{1}{(6 - (-3)(5 - (-2))} = \frac{1}{63} \tag{1}$$

Vi finder nu sandsynligheden for at se alle fire stjerner, ved at gange sandsynlighederne sammen.

$$L_{\theta_1} = \left(\frac{1}{20}\right)^4 = \frac{1}{160000} \tag{2}$$

$$L_{\theta_2} = \left(\frac{1}{63}\right)^4 = \frac{1}{15752961}$$

c)

Når værdierne er inde for nedenstående parametre, vil likelyhood være større end 0.

$$\hat{x}_{\text{min}} = \{x_1, x_2, \ldots, x_n\}$$
$$\hat{x}_{\text{max}} = \{x_1, x_2, \ldots, x_n\}$$
$$\hat{y}_{\text{min}} = \{y_1, y_2, \ldots, y_n\}$$
$$\hat{y}_{\text{max}} = \{y_1, y_2, \ldots, y_n\}$$

Jeg finder disse værdier, som er

$$\hat{x}_{\text{min}} = 0,\ \hat{x}_{\text{max}} = 2,\ \hat{y}_{\text{min}} = 0,\ \hat{y}_{\text{max}} = 2$$

Exercise 3

a)

Ved brug af prior og binomial sandsynlighed, kan vi beregne posterior

$$p(r) = 1, \quad (0 \leq r \leq 1)$$

ved brug af denne formel, som er givet til L7

$$p(r|y_N) \propto p(y_N|r)p(r)$$

Vi sætter ind

$$p(r|y_N) \propto r^{y_N+\alpha-1}(1-r)^{N-y_N+\beta-1} \propto r^{\delta-1}(1-r)^{\gamma-1}$$

og fra opgaveteksten

$$\delta = y_N + 1 \quad \vee \quad \gamma = N - y_N + 1$$

b)

Jeg ganger $2r$ på formlen fra slides.

$$\begin{aligned} r^{Y_N+\alpha-1}(1-r)^{N-Y_N+\beta-1} \cdot 2r &= 2r^{Y_N+\alpha}(1-r)^{N-Y_N+\beta-1} \\ &= 2rr^{\delta-1}(1-r)^{\gamma-1} \end{aligned}$$

Dette giver værdierne

$$\alpha = 2, \quad \beta = 1$$

c)

Exercise 4

a)

Sandsynligheden vil være:

$$p(t|X, w, \sigma^2) = \prod_{n=1}^{N} p(t_n|x_n, w, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}(w^T x_n, \sigma^2)$$

som er givet i L7

b)

c)

Codesnippet af den funktion jeg har lavet:

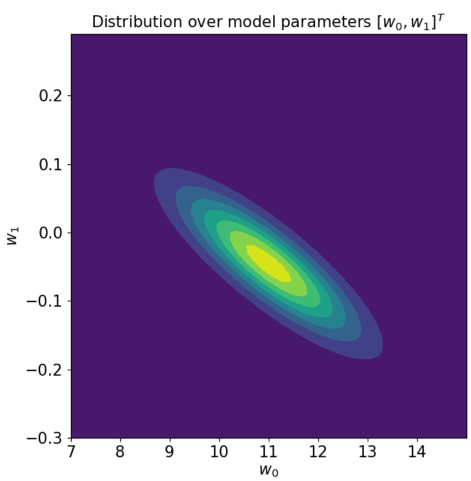
```
muZeros = np.zeros((2, 1))
sigmaZeros = np.array([[100, 0], [0, 5]])
variance = 10

def CorrPos(mu, sigma):
    sigW = np.linalg.inv(1/variance * np.dot(X.T, X) + np.linalg.inv(sigma))
    muW = np.dot(sigW, ((1/variance) * np.dot(X.T, t) + np.dot(np.linalg.inv(sigma), mu)))
    return muW, sigW
```

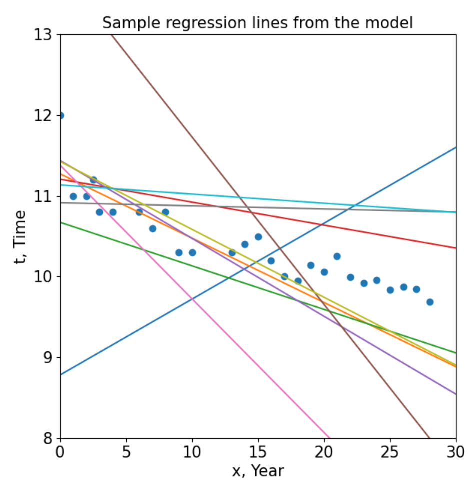
Se hele programmet i appendix.

d)

$$mean = \begin{bmatrix} 10.99417141 \\ -0.04578724 \end{bmatrix}$$

$$variance = \begin{bmatrix} 1.31233642 & -0.06718612 \\ -0.06718612 & 0.00478513 \end{bmatrix}$$





På det første billede kan man se fordelingen af parametrene w_0 og w_1, mens det andet billede viser, hvordan regressionen bliver bedre med hver iteration.

Appendix

A4-Ex4.py

```
# ### Exercise 4
# We mark suggestions for where you should add your code with
# TODO: Add your code here

# Loading packages
import numpy as np
import matplotlib.mlab as mlab
import matplotlib.pyplot as plt
from scipy.stats import multivariate_normal
import matplotlib
matplotlib.rcParams['figure.figsize'] = (15,7)
matplotlib.rc('font', size=15)
matplotlib.rc('axes', titlesize=15)

def visualize_model(mu, Sigma, xmin, xmax, ymin, ymax, x, t, Nsample):
    """This function visualize a normal distribution of linear model parameter (w_0, w_1) by a c
    and by plotting the data points and random samples of parameters (w_0, w_1) as lines.

    As input it takes the following parameters:
    mu - the mean (2x1) of the normal distribution in parameter space (w_0, w_1)
    Sigma - the covariance matrix (2x2) of the normal distribution in parameter space (w_0, w_1)
    xmin, xmax, ymin, ymax - the ranges of (w_0, w_1) to use when plotting the contour plot
    x - the input vector of years
    t - the target vector of running times
    Nsample - number of random samples to illustrate (each is plotted as a line)
    """
    # First, we visualize the model by visualizing the
    # prior/posterior normal distribution over the model parameters $[w_0, w_1]^T$.

    # Define a grid for visualizing normal distribution and define the normal distribution on th
    xx, yy = np.mgrid[xmin:xmax:.01, ymin:ymax:.01]
    pos = np.dstack((xx, yy))
    rv = multivariate_normal(mu.flatten(), Sigma)

    # Plot the normal distribution
    fig, ax = plt.subplots(1,2)
    ax[0].contourf(xx, yy, rv.pdf(pos))
    ax[0].set_xlabel('$w_0$')
    ax[0].set_ylabel('$w_1$')
    ax[0].set_title('Distribution over model parameters $[w_0, w_1]^T$')

    # Second, we visualize the model by drawing samples from the prior/posterior and
```

```
    # visualizing the corresponding regression lines

    # First, we scatter plot the observed data
    ax[1].scatter(x, t)

    # draw sample model parameters from the model
    w_0, w_1 = np.random.multivariate_normal(mu.flatten(), Sigma, Nsample).T

    # Plot the corresponding sample regression lines
    for i in range(Nsample):
        ax[1].plot([0, 30], [w_0[i] + 0*w_1[i], w_0[i] + 30*w_1[i]])

    ax[1].set_xlabel('x, Year')
    ax[1].set_ylabel('t, Time')
    ax[1].set_title('Sample regression lines from the model')
    ax[1].set_xlim(0,30)
    ax[1].set_ylim(8,13)

# Loading data
data = np.loadtxt('men-olympics-100.txt')
N, d = data.shape
print('N = ', N)
print('d = ', d)

x = (data[:,0]-data[0,0]).reshape(N,1) / 4 # Shift and rescale the input for visualization purpo
t = data[:,1].reshape(N,1)
one = np.ones((N,1))
X = np.concatenate((one, x), axis = 1)

plt.scatter(x,t)
plt.xlabel('x')
plt.ylabel('t')
plt.title('The olympic 100m dataset')

# **Exercise 4c)**

muZeros = np.zeros((2, 1))
sigmaZeros = np.array([[100, 0], [0, 5]])
variance = 10

def CorrPos(mu, sigma):
    sigW = np.linalg.inv(1/variance * np.dot(X.T, X) + np.linalg.inv(sigma))
    muW = np.dot(sigW, ((1/variance) * np.dot(X.T, t) + np.dot(np.linalg.inv(sigma), mu)))
    return muW, sigW
```

```python
# **Exercise 4d)**

# TODO: Add your code here and change these lines
muw, Sigmaw = CorrPos(muZeros, sigmaZeros)
print("muw = \n" + str(muw))
print("Sigmaw = \n" + str(Sigmaw))

# Call the function with your predictions of muw and Sigmaw
visualize_model(muw, Sigmaw, 7, 15, -0.3, 0.3, x, t, 10)

plt.show()
```