MDA Part 2, Assignment 1

Marinos Marios

December 10, 2020

Exercise 1

To show that $\theta|y$ we need to evaluate the posterior probability Bayes Theorem.

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta)p(\theta)d\theta} \qquad [1.1]$$

We do know the distributions of both $p(y|\theta)$ and prior $p(\theta)$ as it is given in Exercise.

$$y|\theta \sim N_n(H\theta, \sigma^2 I_n) \ and \ \theta \sim N_2(m_0, P_0) \qquad [1.2]$$

Now we need to calculate the posterior and we can drop the denominator as it **does not depend on theta and thus, it is a constant term**.

$$p(\theta|y) \propto p(y|\theta)p(\theta) \qquad [1.3]$$

$$p(y|\theta)p(\theta) \propto (2\pi\sigma^2)^{-n/2} exp\left(-\frac{1}{2}(y - H\theta)^T(\sigma^2 I_n)^{-1}(y - H\theta)\right) x$$

$$(2\pi P_0)^{-n/2} exp\left(-\frac{1}{2}(\theta - m_0)^T P_0^{-1}(\theta - m_0)\right)$$

Everything that does not include θ can be dropped as it can be consider as constant. So we get :

$$p(y|\theta)p(\theta) \propto exp\left(-\frac{1}{2\sigma^2}(y - H\theta)^T(y - H\theta)\right) x \qquad [1.4]$$

$$exp\left(-\frac{1}{2}(\theta - m_0)^T P_0^{-1}(\theta - m_0)\right)$$

By expanding the inner products and doing the calculations we end up with the following expression :

$$p(y|\theta)p(\theta) \propto exp\left(-\frac{1}{2}\left(-\frac{2y^T H\theta}{\sigma^2} + \frac{\theta^T H^T H\theta}{\sigma^2} + \theta^T P_0^{-1}\theta - 2m_0^T P_0^{-1}\theta\right)\right) \qquad [1.5]$$

Finally, as we want to prove that $\theta|y \sim N_2(\nu, C)$ we know that :

$$p(\theta|y) = exp\left(-\frac{1}{2}\left(\theta - \nu\right)^T C^{-1}(\theta - \nu))\right) \propto exp\left(-\frac{1}{2}(\theta^T\theta - \theta^T\nu - \nu^T\theta + \nu^T\nu)C^{-1}\right)$$

$$\propto exp\left(-\frac{1}{2}(\theta^T\theta C^{-1} - 2\theta^T\nu C^{-1} + \nu^T\nu C^{-1}))\right) \qquad [1.6]$$

$$\propto exp\left(-\frac{1}{2}(\theta^T\theta C^{-1} - 2\nu^T\theta C^{-1})\right)$$

In 1.6 we drop the term $v^2 vC^{-1}$ as it does not depend on θ. Now, the only thing that is left to do is to equate the quadratic terms and the linear terms from 1.6 with the respective terms that we derived in 1.5.

$$\begin{aligned}
\theta^T \theta C^{-1} &= \frac{\theta^T H^T H \theta}{\sigma^2} + \theta^T P_0^{-1} \theta \\
&= \theta^T \theta (\frac{H^T H}{\sigma^2} + P_0^1) \\
C^{-1} &= \frac{H^T H}{\sigma^2} + P_0^{-1}
\end{aligned}$$

And for the linear term :

$$\begin{aligned}
-2v^T \theta C^{-1} &= -\frac{2y^T H \theta}{\sigma^2} - 2m_0^T P_0^{-1} \theta \\
-2v^T \theta C^{-1} &= -2\theta (\frac{y^T H}{\sigma^2} + m_0^T P_0^{-1}) \\
v^T C^{-1} &= \frac{y^T H}{\sigma^2} + m_0^T P_0^{-1} \\
v^T &= C(\frac{y^T H}{\sigma^2} + m_0^T P_0^{-1}) \\
v &= C(H^T \sigma^{-2} y + P_0^{-1} m_0)
\end{aligned}$$

Exercise 2

$$P_k^{-1} = H_k' \sigma^{-2} H_k + P_{k-1}^{-1} \qquad [2.1]$$

$$(A + UCV)^{-1} = A^{-1} - A^{-1} U (C^{-1} + VA^{-1} U)^{-1} VA^{-1} \qquad [2.2]$$

Using 2.1 we can derive :

$$P_k = (H_k' \sigma^{-2} H_k + P_{k-1}^{-1})^{-1} \qquad [2.3]$$

Using Woodlbury identity, 2.2 with $A = P_{k-1}^{-1},\ U = H_K^T,\ V = H_k,\ C = \sigma^{-2}$ get us :

$$P_k = P_{k-1} - P_{k-1} H_k^T (H_k P_{k-1} H_k^T + \sigma^2)^{-1} H_k P_{k-1} \qquad [2.4]$$

In most cases when we have to deal with matrices operations the most time-consuming part of the computations is to find the inverse matrix. In this case the P_k matrix has pxp elements which for problems with many parameters can get very big. With the Wooldbury identity, we avoid that by simply use previous step matrix that we have calculated already. Therefore, when with formula 2.1 we would need to do 2 inverses for pxp matrices one for term P_{k-1}^{-1} and one for all the right-hand-side of formula 2.1 to get P_k whereas with formula 2.4 only one pxp inverse is needed.

Exercise 3

1. Covariance matrix and means after 10 observations :

$$m_{10} = \begin{bmatrix} 40058.09794588 \\ 198.7961864 \\ 567.33226474 \\ 34.11850302 \end{bmatrix}, P_{10} = \begin{bmatrix} 98.99731623 & -1.95964335 & -3.57467222 & 0.2673962 \\ -1.95964335 & 95.3294411 & -10.38902585 & -0.1321786 \\ -3.57467222 & -10.38902585 & 72.06425626 & -8.32226437 \\ 0.2673962 & -0.1321786 & -8.32226437 & 1.07892547 \end{bmatrix}$$

Covariance matrix and means after all observations :

$$m_{30} = \begin{bmatrix} 40072.17057539 \\ 355.67982027 \\ 1324.71598938 \\ -53.82422696 \end{bmatrix}, P_{30} = \begin{bmatrix} 98.74096776 & -2.7336052 & -5.02986195 & 0.43689653 \\ -2.7336052 & 91.88701872 & -21.13117985 & 1.17950301 \\ -5.02986195 & -21.13117985 & 26.78638907 & -2.63424452 \\ 0.43689653 & 1.17950301 & -2.63424452 & 0.28046664 \end{bmatrix}$$

2. Fitted curve is shown in Figure 1.

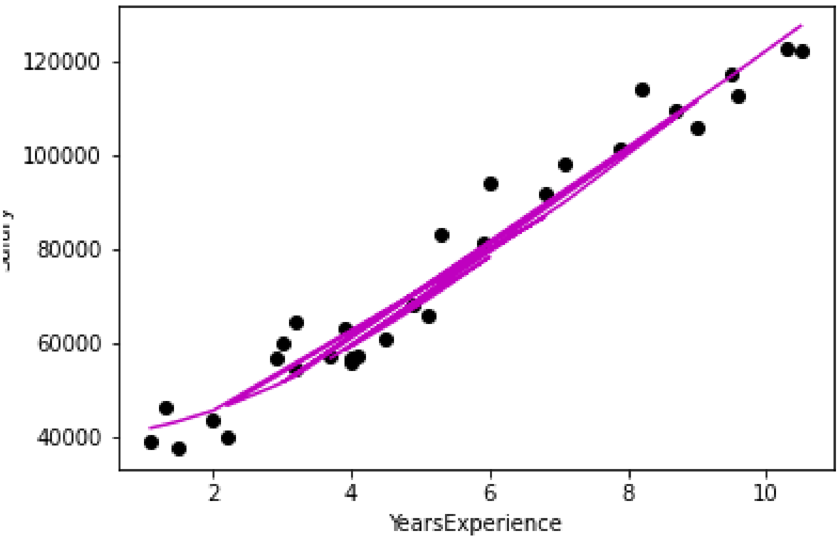

Figure 1: Plotting the fitted curve on the data points.

Exercise 4

Firstly, we need to derive the distribution of $\theta_k|y_{1:k-1}$. which can be obtained as the marginal distribution of $(\theta_k, \theta_{k-1})|y_{1:k-1}$

$$p(\theta_k, \theta_{k-1}|y_{1:k-1}) = \frac{p(\theta_k, \theta_{k-1}, y_{1:k-1})}{p(y_{1:k-1})} \quad [4.1]$$

Applying the probability chain rule we get :

$$\frac{p(\theta_k|\theta_{k-1}, y_{1:k-1})p(\theta_{k-1}|y_{1:k-1})p(y_{1:k-1})}{p(y_{1:k-1})} \quad [4.2]$$

But from definition $\theta_k = A\theta_{k-1} + q_{k-1}$ **does not** depend on $y_{1:k-1}$ so we can drop it from the term $p(\theta_k|\theta_{k-1}, y_{1:k-1})$.

$$So\ p(\theta_k, \theta_{k-1}|y_{1:k-1}) = p(\theta_{k-1}|y_{1:k-1})p(\theta_k|\theta_{k-1}) \quad [4.3]$$

It holds that :

$$\theta_{k-1} \sim N(m_{k-1}, P_{k-1}) \qquad [4.4]$$

$$\theta_k|\theta_{k-1} \sim N(A\theta_{k-1} + q_{k-1}, Q) \qquad [4.5]$$

So we can apply Lemma 1 and conclude :

$$\begin{bmatrix} \theta_{k-1} \\ \theta_k|\theta_{k-1} \end{bmatrix} \sim N\left(\begin{bmatrix} m_{k-1} \\ Am_{k-1} + q_{k-1} \end{bmatrix}, \begin{bmatrix} P_{k-1} & P_{k-1}A^T \\ AP_{k-1} & AP_{k-1}A^T + Q \end{bmatrix}\right) \qquad [4.6]$$

Finally, as hint 1 is pointing out, to get the distribution of $\theta_k|y_{1:k-1}$ we need to obtain the marginal distribution of formula 4.3. Taking the marginal with respect to θ_{k-1} we are left with the second row of the distribution which defines the distribution of $\theta_k|\theta_{k-1}$ and eventually the distrubtion of $\theta_k|y_{1:k-1}$.

$$m_{\bar{k}} = Am_{k-1} + q_{k-1}$$
$$P_{\bar{k}} = AP_{k-1}A^T + Q$$

But q_{k-1} has mean of zero so can be eliminated.

A Appendix Code

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from numpy.linalg import inv
# set the seed so we get the same values for theta
np.random.seed(50)


# loading the data into a dataframe.
Salary_df = pd.read_csv('SalaryData.csv')
Salary_df

# Define constants.
Prior_Means = np.array([[40000],[0], [0], [0]])

# eye creates an identity 4x4 matrix.
Prior_Cov = np.eye(4) * 100
measurement_sigma = 250
# predictions of y's.
y_hats = np.zeros((30,1))


# Loop thorugh each observation to update the priors thetas.
for index, row in Salary_df.iterrows():
    t = row['YearsExperience']
    y = row['Salary']
    H_k = np.array([[1, t, t**2, t**3]])
    current_Means = Prior_Means
    current_Covariances = Prior_Cov
    """current_Cov and Means is respectively P_k-1 and m_k-1,
    H_k is the current data point. Operator @
    is matrix multiplication and .T is transpose. """
    Prior_Cov = current_Covariances - current_Covariances @ H_k.T \
    @ (H_k @ current_Covariances @ H_k.T + measurement_sigma**2)**(-1) \
    @  (H_k @ current_Covariances)
    Prior_Means = Prior_Cov @ (H_k.T * measurement_sigma**(-2) * y  + \
        inv(current_Covariances) @ current_Means)
    # after 10 observations stop.
    # if index==9: break
# final m_k, P_k matrices.
print(Prior_Cov)
print(Prior_Means)

"""firstly we need to derive the thetas from a
multivariate normal distribtuion with the new m_k,
P_k and then predict y's hat based on random thetas
```

```python
drawn from the multivarite normal distribution"""
thetas = np.random.multivariate_normal(Prior_Means.reshape(4,), Prior_Cov)
# for each example we predict the y_hat.
for index, row in Salary_df.iterrows():
    t = row['YearsExperience']
    y_hats[index] = thetas[0] + thetas[1] * t + \
    thetas[2] * t**2 + thetas[3] * t**3

#plot the fitted curve on the data.

plt.plot(Salary_df['YearsExperience'], Salary_df['Salary'], 'o', color = 'black'
plt.plot(Salary_df['YearsExperience'], y_hats, color = 'm' )
plt.xlabel('YearsExperience')
plt.ylabel('Salary')
plt.savefig('Fitted_Curve.png')
```