Statistical Machine Learning
Assignment 2

Inez Wijnands (s4149696) & Guido Zuidhof (s4160703)

Radboud University Nijmegen

04/11/2015

The entire code listing is in a separate file. The listings shown here are merely code snippets.

1 Sequential learning

1.1 Obtaining the prior

1.

$$\tilde{\boldsymbol{\Lambda}}_{a,b} = \tilde{\boldsymbol{\Sigma}}_{a,b}^{-1} \tag{1.1}$$

$$= \left(\begin{array}{cc|cc} 60 & 50 & -48 & 38 \\ 50 & 50 & -50 & 40 \\ \hline -48 & -50 & 52.4 & -41.4 \\ 38 & 40 & -41.4 & 33.4 \end{array}\right) \tag{1.2}$$

Using the precision matrix $\tilde{\boldsymbol{\Lambda}}$ we can use equations 2.69, 2.73 and 2.75 from Bishop to obtain the mean and covariance of the conditional distribution $p([x_1, x_2]^T | x_3 = x_4 = 0)$.

$$\boldsymbol{\Sigma}_p = \boldsymbol{\Lambda}_{aa}^{-1} \tag{Bishop 2.73}$$

$$\boldsymbol{\Lambda}_{aa} = \begin{pmatrix} 60 & 50 \\ 50 & 50 \end{pmatrix} \tag{1.3}$$

$$\boldsymbol{\Sigma}_p = \begin{pmatrix} 0.1 & -0.1 \\ -0.1 & 0.12 \end{pmatrix} \tag{1.4}$$

$$\boldsymbol{\mu}_p = \boldsymbol{\mu}_{a|b} = \boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{aa}^{-1}\boldsymbol{\Lambda}_{ab}(\boldsymbol{x}_b - \boldsymbol{\mu}_b) \tag{Bishop 2.75}$$

We can fill in this equation, since $\tilde{\boldsymbol{\mu}}$ and $\boldsymbol{x}_b$ (the second partition of $\boldsymbol{x}$) are known.

$$\boldsymbol{\mu}_p = \begin{pmatrix} 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 60 & 50 \\ 50 & 50 \end{pmatrix}^{-1} \begin{pmatrix} -48 & 38 \\ -50 & 40 \end{pmatrix} (\begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 \\ 2 \end{pmatrix}) \tag{1.5}$$

$$= \begin{pmatrix} 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0.1 & -0.1 \\ -0.1 & 0.12 \end{pmatrix} \begin{pmatrix} -48 & 38 \\ -50 & 40 \end{pmatrix} (\begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 \\ 2 \end{pmatrix}) \tag{1.6}$$

$$= \begin{pmatrix} 0.8 \\ 0.8 \end{pmatrix} \tag{1.7}$$

2. Using the prior $\boldsymbol{\mu}_p$ and $\boldsymbol{\Sigma}_p$, we used the numpy-equivalent in Python for the MATLAB-function `mvnrnd` to obtain the $\boldsymbol{\mu}_t$ we used for the remainder of this assignment:

```
np.random.multivariate_normal(mu_p, sigma_p, 1)
```

This resulted in:

$$\boldsymbol{\mu}_t = \begin{pmatrix} 0.28584241 \\ 1.42626702 \end{pmatrix} \tag{1.8}$$

3. The probability density is highest at the mean (as illustrated in Figure 1.1). The density decreases quickly as both x and y change, but less so when x XOR y change. In $\boldsymbol{\Sigma}_p$, the values for the x XOR y are lower, so this is consistent with our density plot.

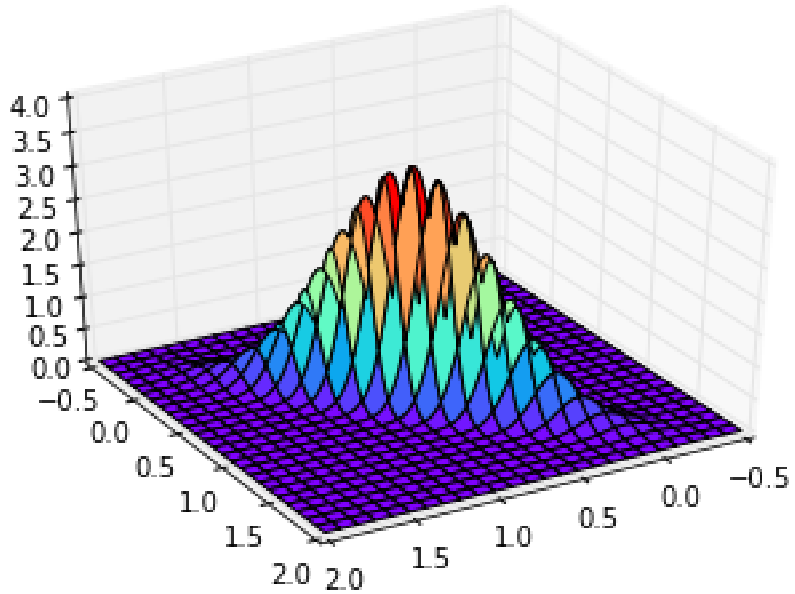

Figure 1.1: The probability density of the distribution.

1.2 Generating the data

1. We used the following function to generate our data:

```
np.random.multivariate_normal(mu_t, sigma_t, 1000)
```

2.

$$\boldsymbol{\mu}_{ML} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n \tag{Bishop 2.121}$$

$$= [\frac{1}{1000}\sum_{n=1}^{1000} \boldsymbol{x}_n, \frac{1}{1000}\sum_{n=1}^{1000} \boldsymbol{y}_n] \tag{1.9}$$

$$= \begin{pmatrix} 0.25383138 \\ 1.38260838 \end{pmatrix} \tag{1.10}$$

$$\boldsymbol{\Sigma}_{ML} = \frac{1}{N}\sum_{n=1}^{N} (\boldsymbol{x}_n - \boldsymbol{\mu}_{ML})(\boldsymbol{x}_n - \boldsymbol{\mu}_{ML})^T \tag{Bishop 1.22}$$

$$= \begin{pmatrix} \frac{1}{1000}\sum_{n=1}^{1000}(\boldsymbol{x}_n - \boldsymbol{\mu}_{ML}(1))(\boldsymbol{x}_n - \boldsymbol{\mu}_{ML}(1))^T & \frac{1}{1000}\sum_{n=1}^{1000}(\boldsymbol{x}_n - \boldsymbol{\mu}_{ML}(1))(\boldsymbol{y}_n - \boldsymbol{\mu}_{ML}(2))^T \\ \frac{1}{1000}\sum_{n=1}^{1000}(\boldsymbol{y}_n - \boldsymbol{\mu}_{ML}(2))(\boldsymbol{x}_n - \boldsymbol{\mu}_{ML}(1))^T & \frac{1}{1000}\sum_{n=1}^{1000}(\boldsymbol{y}_n - \boldsymbol{\mu}_{ML}(2))(\boldsymbol{y}_n - \boldsymbol{\mu}_{ML}(2))^T \end{pmatrix} \tag{1.11}$$

$$= \begin{pmatrix} 1.90513804 & 0.72479489 \\ 0.72479489 & 3.81690496 \end{pmatrix} \tag{1.12}$$

This is calculated using the following code:

```
mu_ml = sum(data)/len(data)

sse = [0,0]
for point in data:
        point = np.matrix(point)
        sse += (point-mu_ml).T*(point-mu_ml)
sigma_ml =  sse/len(data)
```

The differences with the 'true' values are:

$$\boldsymbol{\mu}_t - \boldsymbol{\mu}_{ML} = \begin{pmatrix} 0.28584241 \\ 1.42626702 \end{pmatrix} - \begin{pmatrix} 0.25383138 \\ 1.38260838 \end{pmatrix} = \begin{pmatrix} 0.03201103 \\ 0.04365864 \end{pmatrix} \tag{1.13}$$

$$\boldsymbol{\Sigma}_t - \boldsymbol{\Sigma}_{ML} = \begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 4.0 \end{pmatrix} - \begin{pmatrix} 1.90513804 & 0.72479489 \\ 0.72479489 & 3.81690496 \end{pmatrix} = \begin{pmatrix} 0.09486196 & 0.07520511 \\ 0.07520511 & 0.18309504 \end{pmatrix} \tag{1.14}$$

1.3 Sequential learning algorithms

1.

2.

$$p(\boldsymbol{x}|D_{n-1}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \tag{Bishop 2.113}$$

where: $\boldsymbol{x} = \boldsymbol{\mu}, \boldsymbol{\mu} = \boldsymbol{\mu}_{(n-1)}, \boldsymbol{\Lambda}^{-1} = \boldsymbol{\Sigma}_{(n-1)}$

$$p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}, \boldsymbol{L}^{-1}) \tag{Bishop 2.114}$$

where $\boldsymbol{y} = \boldsymbol{x}_n, \boldsymbol{A} = \boldsymbol{I}, \boldsymbol{x} = \boldsymbol{\mu}, \boldsymbol{b} = 0, \boldsymbol{L}^{-1} = \boldsymbol{\Sigma}^t$

$$p(\boldsymbol{x}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\Sigma}\{\boldsymbol{A}^T\boldsymbol{L}(\boldsymbol{y}-\boldsymbol{b}) + \boldsymbol{\Lambda}\boldsymbol{\mu}\}, \boldsymbol{\Sigma}) \tag{Bishop 2.116}$$

$$\boldsymbol{\Sigma} = (\boldsymbol{\Lambda} + \boldsymbol{A}^T\boldsymbol{L}\boldsymbol{A})^{-1} \tag{Bishop 2.117}$$

Matching the variables we get the following equations:

$$p(\boldsymbol{\mu}|\boldsymbol{x}_n) = \mathcal{N}(\boldsymbol{\mu}|\boldsymbol{S}\{\boldsymbol{I}^T\boldsymbol{\Sigma}_t^{-1}(\boldsymbol{x}_n - 0) + \boldsymbol{\Sigma}_{(n-1)}^{-1}\}, \boldsymbol{S}) \tag{1.15}$$

$$= \mathcal{N}(\boldsymbol{\mu}|\boldsymbol{S}\{\boldsymbol{I}^T\boldsymbol{\Sigma}_t^{-1}\boldsymbol{x}_n + \boldsymbol{\Sigma}_{(n-1)}^{-1}\}, \boldsymbol{S}) \tag{1.16}$$

$$= \mathcal{N}(\boldsymbol{\mu}|\boldsymbol{S}\{\boldsymbol{\Sigma}_t^{-1}\boldsymbol{x}_n + \boldsymbol{\Sigma}_{(n-1)}^{-1}\}, \boldsymbol{S}) \tag{1.17}$$

$$\boldsymbol{S} = (\boldsymbol{\Sigma}_{(n-1)}^{-1} + \boldsymbol{I}^T\boldsymbol{\Sigma}_t^{-1}\boldsymbol{I})^{-1} \tag{1.18}$$

$$= (\boldsymbol{\Sigma}_{(n-1)}^{-1} + \boldsymbol{\Sigma}_t^{-1})^{-1} \tag{1.19}$$

$\boldsymbol{\mu}_n$ is the mean of the distribution $p(\boldsymbol{\mu}|\boldsymbol{x}_n)$, so the functions we use for our sequential learning algorithm are:

$$\boldsymbol{\Sigma}_n = \boldsymbol{S} \tag{1.20}$$

$$\boldsymbol{\mu}_n = \boldsymbol{\Sigma}_n\{\boldsymbol{\Sigma}_t^{-1}\boldsymbol{x}_n + \boldsymbol{\Sigma}_{n-1}^{-1}\}, \boldsymbol{\Sigma}_n) \tag{1.21}$$

3.

4.

2 The faulty lighthouse

2.1 Constructing the model

1.

2.

3.

4.

2.2 Generate the lighthouse data

1.

2.

2.3 Find the lighthouse

1.

2.

3.