# $3^{\text{rd}}$ Homework

Chalkiopoulos Georgios | $p3352124$

January 27, 2022

**Exercise 1.** Prove that:

$$R_{\mathbf{x}} = cov(\mathbf{x}) + \boldsymbol{\mu}\boldsymbol{\mu}^T$$

It is:

$$\begin{aligned}
cov(\mathbf{x}) &= E[(\mathbf{x} - \boldsymbol{\mu}) \cdot (\mathbf{x} - \boldsymbol{\mu})^T] \\
&= E\left[\begin{bmatrix} x_1 - \mu_1 \\ \vdots \\ x_l - \mu_l \end{bmatrix} \begin{bmatrix} x_1 - \mu_1 & \cdots & x_\ell - \mu_\ell \end{bmatrix}\right] \\
&= E\begin{bmatrix} (x_1 - \mu_1)(x_1 - \mu_1) & \dots & (x_1 - \mu_1)(x_\ell - \mu_\ell) \\ \vdots & \ddots & \vdots \\ (x_\ell - \mu_\ell)(x_1 - \mu_1) & \dots & (x_\ell - \mu_\ell)(x_\ell - \mu_\ell) \end{bmatrix} \\
&= \begin{bmatrix} E[(x_1 - \mu_1)(x_1 - \mu_1)] & \dots & E[(x_1 - \mu_1)(x_\ell - \mu_\ell)] \\ \vdots & \ddots & \vdots \\ E[(x_\ell - \mu_\ell)(x_1 - \mu_1)] & \dots & E[(x_\ell - \mu_\ell)(x_\ell - \mu_\ell)] \end{bmatrix} \\
&= \begin{bmatrix} cov(x_1 x_1) & \dots & cov(x_1 x_\ell) \\ \vdots & \ddots & \vdots \\ cov(x_\ell x_1) & \dots & cov(x_\ell x_\ell) \end{bmatrix} \qquad (1.1)
\end{aligned}$$

Moreover:

$$\begin{aligned}
R_{\mathbf{x}} &= E[\mathbf{x} \cdot \mathbf{x}^T] \\
&= E\left[\begin{bmatrix} x_1 \\ \vdots \\ x_l \end{bmatrix} \begin{bmatrix} x_1 & \cdots & x_\ell \end{bmatrix}\right]
\end{aligned}$$

$$= \begin{bmatrix} E(x_1x_1) & \dots & E(x_1x_\ell) \\ \vdots & \ddots & \vdots \\ E(x_\ell x_1) & \dots & E(x_\ell x_\ell) \end{bmatrix} \tag{1.2}$$

We also know that the **Correlation** between two **rv's** x and y is:

$$r_{xy} = E(xy) = cov(xy) + E[x]E[y] \tag{1.3}$$

Furthermore:

$$\begin{aligned} \boldsymbol{\mu}\boldsymbol{\mu}^T &= \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_l \end{bmatrix} \begin{bmatrix} \mu_1 & \cdots & \mu_\ell \end{bmatrix} \\ &= \begin{bmatrix} \mu_1\mu_1 & \dots & \mu_1\mu_\ell \\ \vdots & \ddots & \vdots \\ \mu_\ell\mu_1 & \dots & \mu_\ell\mu_\ell \end{bmatrix} \end{aligned} \tag{1.4}$$

Using (1.2), (1.3) and (1.4) we can write:

$$\begin{aligned} R_{\mathbf{x}} &= \begin{bmatrix} cov(x_1x_1) + \mu_1\mu_1 & \dots & cov(x_1x_\ell) + \mu_1\mu_\ell \\ \vdots & \ddots & \vdots \\ cov(x_\ell x_1) + \mu_\ell\mu_1 & \dots & cov(x_\ell x_\ell) + \mu_\ell\mu_\ell \end{bmatrix} \\ &= \begin{bmatrix} cov(x_1x_1) & \dots & cov(x_1x_\ell) \\ \vdots & \ddots & \vdots \\ cov(x_\ell x_1) & \dots & cov(x_\ell x_\ell) \end{bmatrix} + \begin{bmatrix} \mu_1\mu_1 & \dots & \mu_1\mu_\ell \\ \vdots & \ddots & \vdots \\ \mu_\ell\mu_1 & \dots & \mu_\ell\mu_\ell \end{bmatrix} \\ &= cov(\mathbf{x}) + \boldsymbol{\mu}\boldsymbol{\mu}^T \end{aligned}$$

**Exercise 2.**

**(a) Bernoulli Distribution Mean and Variance.**

It is:

$$E[X] = \sum_{i=0}^{n} x p_X(x) = 1 \cdot p + 0 \cdot (1-p) = p$$

Moreover:

$$E[X^2] = \sum_{i=0}^{n} x^2 p_X(x) = 1^2 \cdot p + 0^2 \cdot (1-p) = p$$

And:

$$E[X]^2 = p^2$$

Thus:

$$\sigma_X^2 = E[X^2] - E[X]^2 = p - p^2 = p(1-p)$$

**(b) Binomial Distribution Mean.**

The binomial random variable can be thought of as the sum of $n$ independent Bernoulli random variables, each with mean $p$ and variance $p(1-p)$. Let $U_1, \ldots, U_n$ be independent Bernoulli random variables. We can calculate the mean of the Binomial as follows:

$$E[X] = E[U_1 + \cdots + U_n] = E[U_1] + \cdots + E[U_n] = np$$

Another way of calculating the mean is using the definition of the mean. It is:

$$\begin{aligned}
E[X] &= \sum_{x=0}^{n} x p_X(x) \\
&= \sum_{x=0}^{n} k \binom{n}{k} p^k q^{n-k} \\
&= \sum_{x=1}^{n} k \binom{n}{k} p^k q^{n-k}, \quad with : k\binom{n}{k} = n\binom{n-1}{k-1} \\
&= np \sum_{x=1}^{n} \binom{n-1}{k-1} p^{k-1} q^{(n-1)-(k-1)}, \quad with : m = n-1, j = k-1 \\
&= np \sum_{j=0}^{m} \binom{m}{j} p^j q^{(m-j)} = np
\end{aligned}$$

**Exercise 3.**

It is:

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{\ell/2}|\Sigma|^{1/2}} \exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2}\right) \tag{3.1}$$

We will write all components as products. We can first calculate:

$$\frac{1}{(2\pi)^{\ell/2}} = \prod_{i=1}^{\ell} \frac{1}{\sqrt{2\pi}} \tag{3.2}$$

Moreover since $\Sigma$ is diagonal:

$$\begin{aligned}
\frac{1}{|\Sigma|^{1/2}} &= \frac{1}{\begin{vmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ddots & \sigma_\ell^2 \end{vmatrix}^{1/2}} \\
&= \frac{1}{(\sigma_1^2 \dots \sigma_\ell^2)^{1/2}} \\
&= \prod_{i=1}^{\ell} \frac{1}{\sqrt{\sigma_i^2}} \qquad (3.3)
\end{aligned}$$

Regarding the exponential component we have that:

$$(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) = \begin{bmatrix}(x_1-\mu_1) & \dots & (x_\ell-\mu_\ell)\end{bmatrix} \begin{bmatrix} \sigma_1^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \vdots & \sigma_\ell^2 \end{bmatrix}^{-1} \begin{bmatrix}(x_1-\mu_1) \\ \vdots \\ (x_\ell-\mu_\ell)\end{bmatrix}$$

$$= \begin{bmatrix}\frac{(x_1-\mu_1)}{\sigma_1^2} & \dots & \frac{(x_\ell-\mu_\ell)}{\sigma_\ell^2}\end{bmatrix} \begin{bmatrix}(x_1-\mu_1) \\ \vdots \\ (x_\ell-\mu_\ell)\end{bmatrix}$$

$$= \frac{(x_1-\mu_1)^2}{\sigma_1^2} + \cdots + \frac{(x_\ell-\mu_\ell)^2}{\sigma_\ell^2}$$

$$= \sum_{i=1}^{n} \frac{(x_i-\mu_i)^2}{\sigma_i^2} \Rightarrow$$

$$\begin{aligned}\exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2}\right) &= \exp\left(\sum_{i=1}^{n} -\frac{(x_i-\mu_i)^2}{2\sigma_i^2}\right)\\ &= \prod_{i=1}^{n}\exp\left(-\frac{(x_i-\mu_i)^2}{2\sigma_i^2}\right) \qquad (3.4)\end{aligned}$$

Using (3.1), (3.2), (3.3) and (3.4), we get that:

$$\begin{aligned}p(\boldsymbol{x}) &= \frac{1}{(2\pi)^{\ell/2}|\Sigma|^{1/2}}\exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2}\right)\\ &= \prod_{i=1}^{\ell}\frac{1}{\sqrt{2\pi}}\prod_{i=1}^{\ell}\frac{1}{\sqrt{\sigma_i^2}}\prod_{i=1}^{n}\exp\left(-\frac{(x_i-\mu_i)^2}{2\sigma_i^2}\right)\\ &= \prod_{i=1}^{\ell}\frac{1}{\sqrt{2\pi\sigma_i^2}}\exp\left(-\frac{(x_i-\mu_i)^2}{2\sigma_i^2}\right) \qquad (3.5)\end{aligned}$$

Equation (3.5) indicates that random variables $\mathbf{x}$, $\mathbf{x}_i$ are statistically independent. This conclusion that was reached by assuming that the random variables are mutually uncorrelated.

**Exercise 4.**

Considering the linear model $y = \theta \cdot x + n$ we derive the following dataset:

$$X = \{(y_1, x_1), \ldots, (y_N, x_N)\}$$

where $x = [x_1 \ldots x_n]^T$, $y = [y_1 \ldots y_n]^T$ and $\theta$ a scalar. We will estimate $\theta$ using the Least Squares Criterion. It is:

$$\begin{aligned} J(\theta) &= \sum_{n=1}^{N} (y_n - \theta x_n) \Rightarrow \\ \frac{\partial J(\theta)}{\partial \theta} &= -2 \sum_{n=1}^{N} (y_n - \theta x_n) x_n \\ &= -2 \sum_{n=1}^{N} (y_n x_n - \theta x_n x_n) \end{aligned}$$

Setting the gradient equal to zero we obtain:

$$\begin{aligned} \frac{\partial J(\theta)}{\partial \theta} &= 0 \Leftrightarrow \\ -2 \sum_{n=1}^{N} (y_n - \theta x_n) x_n &= 0 \Leftrightarrow \\ \sum_{n=1}^{N} (y_n x_n) &= \sum_{n=1}^{N} (\theta x_n x_n) \Leftrightarrow \\ \sum_{n=1}^{N} (y_n x_n) &= \theta \sum_{n=1}^{N} (x_n^2) \Leftrightarrow \\ \theta &= \frac{\sum_{n=1}^{N} (y_n x_n)}{\sum_{n=1}^{N} (x_n^2)} \\ &= \frac{X^T y}{X^T X} \end{aligned}$$

# Exercise 5

## 5.a Generate the set.

[1]:
```
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
import matplotlib.cm as cm

import pandas as pd

# for creating a responsive plot
%matplotlib inline
```

[2]:
```
def generate_data():
    # Construct X matrix [1, x1, x2, x1*x2]
    X = np.random.uniform(low=0,high=10,size=(30,1))

    # define theta
    theta = 2

    # define normal error
    n = np.random.normal(0,np.sqrt(64),len(X))

    # Define y using only x1, x2
    y = theta * (X.T) + n

    #prin X and y
    return(np.concatenate((X, y.reshape(-1,1), n.reshape(-1,1)), axis=1))

def yield_index(ds, num):
    """
    Function to return a virtual dataset from a np array with 30 data␣
 ↪points per dataset
    Input:  an array containing all dataset, in order
            the requested dataset point (ex. in order to fetch dataset 30␣
 ↪num should be 30)
    Output: the range in which the specific dataset can be found
    """
    return ds[num*30-30: num*30]

# Geneerate 50 datasets
data = np.empty((1,3))
```

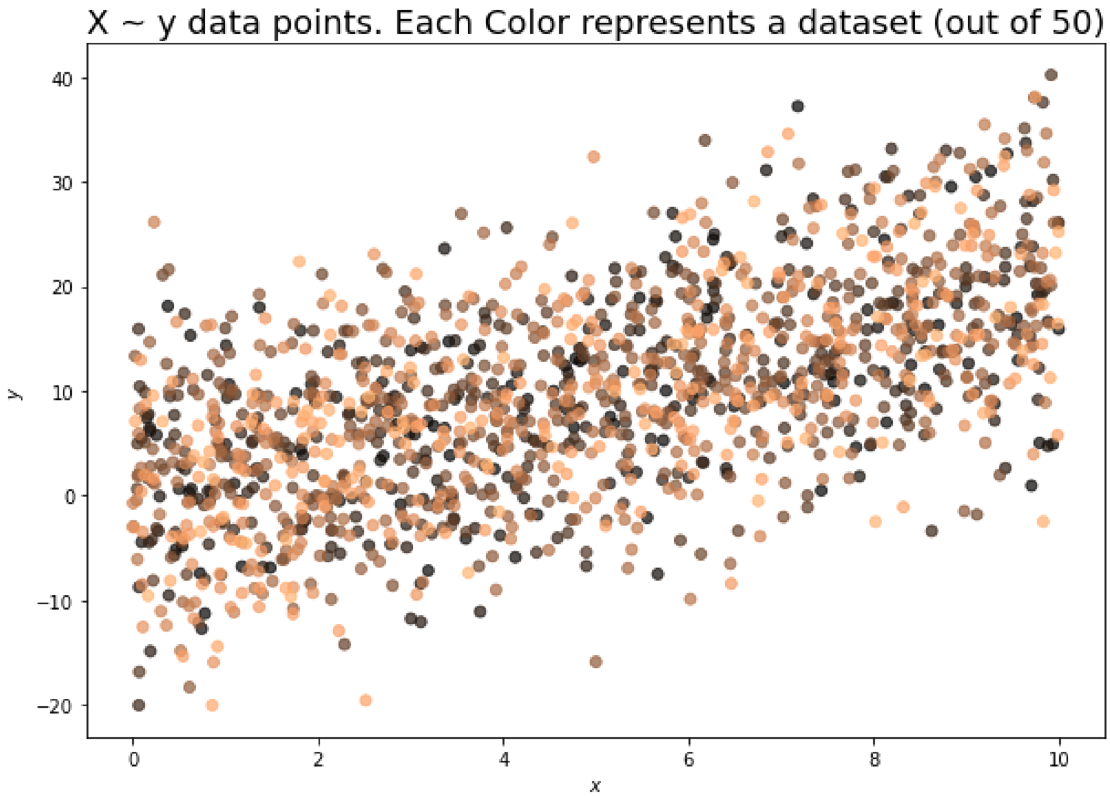
```
for i in range(50):
    data = np.concatenate((data, generate_data()))
data = data[1:]

X_all = data[:,0]
y_all = data[:,1]
data[:,:5]
```

[2]:
```
array([[ 4.54342735,  9.96753916,  0.88068447],
       [ 7.56474221, 10.85811453, -4.27136988],
       [ 8.11494008, 24.4602244 ,  8.23034424],
       ...,
       [ 0.16736766, -9.56594507, -9.9006804 ],
       [ 8.43933605, 25.98851864,  9.10984655],
       [ 0.73239391, -3.45024141, -4.91502922]])
```

[3]:
```
# Create 50 shades of color
colormap = plt.cm.copper #nipy_spectral, Set1,Paired
colorst = [colormap(i) for i in np.linspace(0, 0.9,50)]

#plot X data
fig = plt.figure(figsize=(10,7))
ax = fig.add_subplot(111)
for i in range(50):
    ax.scatter(yield_index(X_all, i+1),yield_index(y_all, i+1),␣
 ↪c=[colorst[i]]*30,marker='o', alpha = 0.7)
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')
ax.set_title('X ~ y data points. Each Color represents a dataset (out of␣
 ↪50)', fontsize=18)
plt.show()
```




## 5.b Calculate LS estimates of $\theta$

[4]:
```
# Calculate \theta
theta = []
for i in range(50):
    X = yield_index(X_all, i+1)
    y = yield_index(y_all, i+1)
    XX = X.dot(X.T)
    Xy = X.dot(y.T)
    theta.append(Xy/(XX))
theta = np.array(theta).reshape(-1,1)
theta[:5]
```

[4]:
```
array([[1.89790379],
       [1.9932325 ],
       [1.422784  ],
       [2.55931934],
       [1.82579796]])
```

### 5.c

#### 5.c1 Estimate the $MSE = E[(\hat{\theta} - \theta_0)^2]$

[5]:
```
mse = np.power((np.full((50), 2) - theta),2).mean()
print(f"The MSE is: {mse:.3f}")
```

```
The MSE is: 0.066
```

#### 5.c2 depict graphically the values of $\hat{\theta}_1, \ldots, \hat{\theta}_d$ and comment.

Looking at the histogram below, we could say that the estimates of $\theta$ follow (kind of) a normal distribution, spread around the value of 2, which is the actual value of $\theta$. This is explained due to the noise, which follows a normal distibution with a mean of zero and a standard deviation of 64. Comparing the histograms of the noise and the theta estimates we can see that the standard deviation in also the same (althought the scale differs).

[6]:
```
fig, ax1 = plt.subplots(figsize=(8, 6))
ax2 = ax1.twiny().twinx()
ax2.hist(theta, color='C1', alpha=0.7, label='Theta')
ax1.hist(data[:,2], color='C0', alpha=0.7, label='Noise')
fig.legend()
ax1.set_title('Histograms of Noise and Theta', fontsize=18)
plt.show()
```