Solutions Exercise 1

Question 2

(a) By using the definition of variance

$$\mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2,$$

we can write $\mathrm{Var}(X+Y)$ as $\mathbb{E}[(X+Y)^2] - \mathbb{E}[X+Y]^2$. Due to linearity of expectation, the latter term is $(\mathbb{E}[X] + \mathbb{E}[Y])^2$ and by expanding both squares we get

$$\begin{aligned}\mathbb{E}[(X+Y)^2] - \mathbb{E}[X+Y]^2 &= \mathbb{E}[X^2] + \mathbb{E}[Y^2] + 2\mathbb{E}[XY] - \mathbb{E}[X]^2 - \mathbb{E}[Y]^2 - 2\mathbb{E}[X]\mathbb{E}[Y] \\ &= \mathrm{Var}(X) + \mathrm{Var}(Y) + 2(\mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]).\end{aligned}$$

Finally, we recognize that $(\mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]) = \mathrm{Cov}(X,Y)$ and hence

$$\mathrm{Var}(X+Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) + 2\mathrm{Cov}(X,Y).$$

What this means in practice is that for independent variables the variance of the sum is given by the sum of the variances, but for correlating variables the variance differs from that. If $X = Y$ then we get $\mathrm{Var}(X+Y) = 4\mathrm{Var}(X)$ since the covariance equals the variance; this equals the direct calculation $\mathrm{Var}(2X) = 4\mathrm{Var}(X)$.

(b) To compute the mean of the Gamma distribution $p_X(x|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1}e^{-\beta x}$ we simply start with the definition of mean:

$$\mathbb{E}[X] = \int_0^\infty xp(x|\alpha,\beta)dx = \int x\frac{\beta^\alpha}{\Gamma(\alpha)}x^{\alpha-1}e^{-\beta x}dx$$

and simplify it to get

$$\frac{\beta^\alpha}{\Gamma(\alpha)}\int x^\alpha e^{-\beta}dx.$$

We could now integrate this directly, but it is quite tedious. Instead, we can recognize that the term inside the integral corresponds to unnormalized pdf of another gamma density with parameters $\hat{\alpha} = \alpha + 1$ and β. We know the normalization term for that, and can write the expectation as

$$\frac{\beta^\alpha}{\Gamma(\alpha)}\frac{\Gamma(\alpha+1)}{\beta^{\alpha+1}}\int \frac{\beta^{\alpha+1}}{\Gamma(\alpha+1)}x^\alpha e^{-\beta}dx.$$

Its value did not change since we merely multiplied the equation with $\frac{\Gamma(\alpha+1)\beta^{\alpha+1}}{\beta^{\alpha+1}\Gamma(\alpha+1)} = 1$.

Now we know that the integral is one due to the definition of probability density functions. By further plugging in $\Gamma(\alpha+1) = \Gamma(\alpha)\alpha$ we can simplify the whole expectation to $\mathbb{E}[x] = \frac{\alpha}{\beta}$

(c) We notice that the pdf of $\mathbf{x}$ is multivariate normal distribution with mean $\mathbf{0}$ and covariance $\mathbf{I}$.

By using the laws of expectation we can write

$$\mathbb{E}[\mathbf{y}] = \mathbb{E}[\mathbf{A}\mathbf{x} + \mathbf{b}] = \mathbf{A}\mathbb{E}[\mathbf{x}] + \mathbf{b} = \mathbf{b},$$

since $\mathbb{E}[\mathbf{x}] = 0$.

Similarly the covariance is obtained by

$$\begin{aligned}\mathbb{E}[\mathbf{y}\mathbf{y}^T] - \mathbb{E}[\mathbf{y}]\mathbb{E}[\mathbf{y}]^T &= \mathbb{E}[(\mathbf{A}\mathbf{x} + \mathbf{b})(\mathbf{A}\mathbf{x} + \mathbf{b})^T] \\ &= \mathbb{E}[\mathbf{A}\mathbf{x}\mathbf{x}^T\mathbf{A}^T] + \mathbb{E}[\mathbf{A}\mathbf{x}\mathbf{b}^T] + \mathbb{E}[\mathbf{b}\mathbf{x}^T\mathbf{A}^T] + \mathbb{E}[\mathbf{b}]\mathbb{E}[\mathbf{b}]^T - \mathbb{E}[\mathbf{y}]\mathbb{E}[\mathbf{y}]^T.\end{aligned}$$

Here the last two terms cancel out since $\mathbb{E}[\mathbf{y}] = \mathbf{b}$, the middle terms are zero because $\mathbb{E}[\mathbf{x}] = 0$, and the first term simplifies into $\mathbf{A}\mathbb{E}[\mathbf{x}\mathbf{x}^T]\mathbf{A}^T = \mathbf{A}\mathbf{A}^T$ by plugging in the known covariance of $\mathbf{x}$.

The resulting pdf is that of a multivariate normal distribution with mean $\mathbf{b}$ and covariance $\mathbf{A}\mathbf{A}^T$, and hence is written as

$$p(\mathbf{y}) = (2\pi)^{-D/2}|\mathbf{A}\mathbf{A}^T|^{-1/2}e^{-\frac{1}{2}(\mathbf{y}-\mathbf{b})^T(\mathbf{A}\mathbf{A}^T)^{-1}(\mathbf{y}-\mathbf{b})}$$

Question 3

The code for solving the exercise is given on the next page and should be relatively straightforward – the point of this exercise was to just play around with numerical computation in Python.

Some things worth noting:

- Here $N = 200$ and $D = 5$. The fact that the file contained $\mathbf{X}^T$ instead of $\mathbf{X}$ might have been a bit confusing but it was intentional; you always need to pay attention to the exact definitions. Some of you had $N = 199$ – the csv file had no header but your code interpreted the first line as one.

- Pay attention to axis scaling for scatter-plots: If you did not scale the axes then the two-dimensional representation for the data looks like a spherical normal distribution, which is misleading.

- Also remember to label your axes and use the correct type of a plot (dots for the scatter-plot, line for the error curve) – always think of the presentation when preparing a plot.

- The different dimensions of the PCA representation are uncorrelated, seen clearly in the scatter-plot.

- For PCA the average reconstruction error (the error divided by N) is exactly the sum of the eigenvalues for dimensions that were not kept.

- Note that numpy.linalg.eig is not guaranteed to return the eigenvalues in any particular order, so you need to re-order them manually.

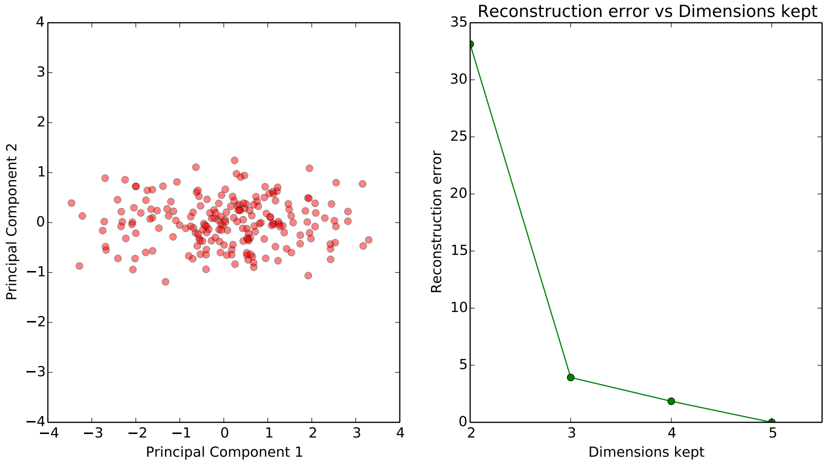

Figure 1: The left plot answers to task (d), showing the 200 samples in the new two-dimensional representation. The right plot shows the reconstruction error as a function of L. For $L = 5$ there is no error, and we see that $L = 2$ is not enough to capture the true data structure – the error is considerably larger than for $L = 3$. However, it is good to contrast these values to the original data variance – if we keep $L = 0$ dimensions the reconstruction error would be 480 and hence already $L = 2$ captures more than 90% of the structure.

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.backends.backend_pdf import PdfPages

def pcaExercise():
    ## Load data and call it X
    X = np.genfromtxt ('exercise_1_data.csv', delimiter=",")
    ## Compute the covariance of matrix X
    covMat = np.cov(X.T)
    ## Eigendecomposition of covariance matrix
    evals, evecs = np.linalg.eig(covMat)
    ## Sort the eigenvalues and eigenvectors in descending order
    tempvec = [(evals[i],i) for i in range(5)]
    tempvec.sort()
    tempvec.reverse()
    dsorder = [tempvec[i][1] for i in range(len(evecs))]
    ## Compute  principal component 1 and principal component 2
    # PC1 and PC2 projected onto eigenvectors
    PC1 = np.dot(X, evecs[:,dsorder[0]])
    PC2 = np.dot(X, evecs[:,dsorder[1]])
    ## Compute reconstruction errors as a function of dimensions kept
    rerror = []
    for k in range(len(evals)):
        Wmat1 = evecs[:,dsorder[:(k+1)]]
        Xr = np.dot(X, np.dot(Wmat1, Wmat1.T))
        Xerror = np.sum((X-Xr)**2)
        rerror.append(Xerror)
    ## plot
    plotPCA(rerror, PC1, PC2)

def plotPCA(rerror, PC1, PC2):
    xpts=[i+2 for i in range(len(rerror)-1)]
    fig,axes=plt.subplots(figsize=(12,6), nrows=1, ncols=2,squeeze=False)
    axes[0][1].plot(xpts, rerror[1:],"g-")
    axes[0][1].set_xlabel("Dimensions kept")
    axes[0][1].set_ylabel("Reconstruction error")
    axes[0][1].set_title("Reconstruction error vs Dimensions kept")
    axes[0][1].set_xlim(2, 5.5)
    axes[0][1].set_xticks(np.arange(2, 6, 1))
    axes[0][0].plot(PC1, PC2, 'ro', alpha=0.5)
    axes[0][0].set_xlabel("Principal Component 1")
    axes[0][0].set_ylabel("Principal Component 2")
    axes[0][0].set_ylim(-4, 4)
    #fig.tight_layout()
    # plt.show()
    with PdfPages('plot_exercise1.pdf') as pdf:
        pdf.savefig(fig)
    plt.close()
    return 0


if __name__ == '__main__':
    pcaExercise()
```

Question 4

Note: To simplify notation the vectors are not written in boldface here.

The likelihood for a single data point in logistic regression is given by

$$p((x,y),\theta) = P(y=1)^y \times P(y=0)^{1-y}.$$

Applying log on both sides, we get the log-likelihood $\log p((x,y),\theta)$

$$\log p((x,y),\theta) = y \log P(y=1) + (1-y) \log P(y=0).$$

Replacing $P(y=1)$ and $P(y=0)$ with the corresponding sigmoid probabilities we get

$$\log p((x,y),\theta) = y \log s(\theta^T x) + (1-y) \log(1 - s(\theta^T x)).$$

Hence, the loss function given in the exercise corresponds to the minus logarithmic likelihood of a Bernoulli model parameterized by the sigmoid function.

a) Before computing the $\nabla L((x,y),\theta)$, we first simplify $\dfrac{d}{dz}s(z)$:

$$\begin{aligned}
\frac{d}{dz}s(z) &= \frac{d}{dz}\left[\frac{1}{1+e^{-z}}\right] \\
&= \frac{d}{dz}\left(1+\mathrm{e}^{-z}\right)^{-1} \\
&= (1+e^{-z})^{-2}(e^{-z}) \\
&= \frac{e^{-z}}{(1+e^{-z})^2} \\
&= \frac{1}{1+e^{-z}}\cdot\left(1-\frac{1}{1+e^{-z}}\right) \\
&= s(z)\cdot(1-s(z)) \qquad (1)
\end{aligned}$$

Next we compute the derivative of the log-likelihood with respect to the ith element of the parameter vector, $\frac{\partial L}{\partial \theta_i}$:

$$\begin{aligned}
\frac{\partial L}{\partial \theta_i} &= -\frac{y}{s(\theta^T x)} \times \nabla_{\theta_i} s(\theta^T x) + \frac{1-y}{1-s(\theta^T x)} \times \nabla_{\theta_i} s(\theta^T x) \\
&= \left(-\frac{y}{s(\theta^T x)} + \frac{1-y}{1-s(\theta^T x)}\right) \times \nabla_{\theta_i} s(\theta^T x) \\
&= \left(-\frac{y}{s(\theta^T x)} + \frac{1-y}{1-s(\theta^T x)}\right) \times s(\theta^T x)(1-s(\theta^T x))\nabla_{\theta_i}(\theta^T x) \text{ using (1)} \\
&= \left(\frac{-y(1-s(\theta^T x)) + (1-y)s(\theta^T x)}{s(\theta^T x)(1-s(\theta^T x))}\right) \times s(\theta^T x)(1-s(\theta^T x))x_i \\
&= (s(\theta^T x)-y)x_i. \qquad (2)
\end{aligned}$$

The final gradient in vector is hence given by $\frac{\partial L}{\partial \theta} = (s(\theta^T x)-y)x$.

b) The calculation of the Hessian matrix can also be done for a single element:

$$\begin{aligned}
\frac{\partial L}{\partial \theta_j \partial \theta_i} &= \frac{\partial}{\partial \theta_j}(s(\theta^T x)-y)x_i \\
&= x_i s(\theta^T x)(1-s(\theta^T x))\nabla_{\theta_j}(\theta^T x) \\
&= x_i x_j s(\theta^T x)(1-s(\theta^T x)). \qquad (3)
\end{aligned}$$

The whole Hessian matrix is hence

$$H(L((x,y),\theta)) = xx^T s(\theta^T x)(1-s(\theta^T x))$$

c) Evaluating the gradient and Hessian at $\theta = [1,1]$, $\mathbf{x} = [-1,2]$ and $y = 1$ consists of simply plugging in the values. The gradient (2) is

$$\theta^T x = 1$$

$$s(\theta^T x) \approx 0.73$$

$$\nabla L = (s(\theta^T x) - y)x \approx [0.27, -0.54].$$

The Hessian is obtained by multiplying the outer-product

$$xx^T = \left(\begin{smallmatrix} 1 & -2 \\ -2 & 4 \end{smallmatrix}\right)$$

with $s(\theta^T x)(1 - s(\theta^T x)) = 0.1966$ resulting in

$$H \approx \left(\begin{smallmatrix} 0.197 & -0.393 \\ -0.393 & 0.786 \end{smallmatrix}\right).$$