ELE 382: Statistical Signal Processing Fall 2017

Homework 4

Due date: Wednesday, Oct. 18, 2017 (at the beginning of class)

You are allowed to drop 1 problem without penalty. If you attempt all problems, then the problem with the lowest score will automatically be dropped. In addition, up to 2 bonus points will be awarded to each problem for clean, well-organized, and elegant solutions.

1. Transformation of Gaussians (*10 points*)

Let

$$\boldsymbol{X} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} 1 & 2 & 2 \\ 2 & 4 & 4 \\ 2 & 4 & 9 \end{bmatrix}\right).$$

Find a deterministic transformation T such that

$$T(\boldsymbol{X}) \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix}\right).$$

You can use Matlab to perform basic matrix calculation.

2. Gaussian random vectors (*10 points*)

Suppose $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ is a Gaussian random vector with

$$\boldsymbol{\mu} = \begin{bmatrix} 1 \\ 5 \\ 2 \end{bmatrix} \quad \text{and} \quad \Sigma = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 4 & 0 \\ 0 & 0 & 9 \end{bmatrix}.$$

(a) Find the pdf of X_1.

(b) Find the pdf of $X_2 + X_3$.

(c) Find the pdf of $2X_1 + X_2 + X_3$.

(d) Find $\mathbb{P}\{2X_1 + X_2 + X_3 < 0\}$.

(e) Find the joint pdf of $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$, where $\boldsymbol{A} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & -1 & 1 \end{bmatrix}$.

3. Conditioning (*10 points*)

Consider the random vector $\boldsymbol{X}$ defined in Problem 2.

(a) Find the PDF of X_3 given (X_1, X_2).

(b) Find the PDF of (X_2, X_3) given X_1.

(c) Find the PDF of X_1 given (X_2, X_3).

4. Whitening (*10 points*)

Let U_1, U_2, and U_3 be independent random variables, where U_i has mean zero and variance $\sigma_{u_i}^2 = i$ for $i = 1, 2, 3$. Define

$$\boldsymbol{X} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix},$$

such that

$$X_i = \sum_{j=1}^{i} U_j, \quad i = 1, 2, 3.$$

(a) Find $\Sigma_{\boldsymbol{X}}$.

(b) Find a whitening matrix $\boldsymbol{A}$ for $\boldsymbol{X}$ such that $\mathsf{Cov}(\boldsymbol{AX}) = \boldsymbol{I}$.

5. MNIST (*10 points*)

Download the data package at `https://www.princeton.edu/~yc5/ele382_SSP/homeworks/hw4-data.zip`, where you'll find the MNIST handwritten digit dataset. The data package includes two image subsets (concerning digit 0 and digit 2) each containing 4999 instances of 28×28 pixels.

(a) In each subset, each image can be vectorized as a 784-dimensional vector $\mathbf{X}_i$ $(1 \leq i \leq 4999)$. Suppose that each image in subset l are independently drawn from some distribution $\mathbb{P}_l$ $(l = 1, 2)$. For each subset of n images, we can estimate the covariance matrix as follows

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n} \sum_{i=1}^{n} \left(\boldsymbol{X}_i - \frac{1}{n} \sum_{j=1}^{n} \boldsymbol{X}_j \right) \left(\boldsymbol{X}_i - \frac{1}{n} \sum_{j=1}^{n} \boldsymbol{X}_j \right)^{\top}.$$

Note that we remove the mean component $\frac{1}{n} \sum_{j=1}^{n} \boldsymbol{X}_j$ from each instance $\boldsymbol{X}_i$.

Compute an estimate of the covariance matrix for each subset of images (i.e. the set of digit-0 images and the set of digit-2 images) using any programming language you like.

(b) For each subset of images, compute and plot the first few eigenvectors (as images) of the estimated covariance matrix. These are often referred to as the principal components of the data.

(c) Compute an estimate of the covariance matrix for the entire set of images (including all digit-0 and digit-2 images). Plot its first few eigenvectors as images. Compare and contrast these plots with the ones you obtain in Part (b).

(d) Plot the eigenvalues for the covariance matrices computed in Part (a) and Part (c). Explain your findings in the plots. For instance, from the eigenvalues, what can you say about the "effective" dimensionality of the data? What does your observation suggest for feature selection for say recognizing digits?

Please include your code. You can use any programming language (e.g. Matlab, Python) for this problem, but you are not allowed to use off-the-shelf functions like "PCA" or "cov". For Matlab, you might find the functions *imshow* and *imread* useful in plotting and reading image data.