ECE 301 (Section 001) Homework 8
Spring 2023, Dr. Chau-Wai Wong
TA in Charge: Mushfiqur Rahman

Problem 1 (Projection and PCA) You are given a data matrix

$$\mathbf{X} = \begin{bmatrix} -0.92 & 1.09 & -1.35 & 2.06 & -0.60 & -0.28 \\ -0.08 & 1.15 & -1.67 & 1.08 & -1.14 & 0.66 \end{bmatrix}.$$

and its principal component $\mathbf{u} = [0.7474,\ 0.6644]^T$.

a) Plot the data points using circles and the principal component using a line segment with one end at $[0, 0]^T$ on a 2D plane using Matlab.

b) Calculate the projection c_i for each data point $\mathbf{x}_i$ to $\mathbf{u}$. Note that c_i can be negative.

c) Calculate the sample variance of $\{c_i\}_{i=1}^n$, namely,

$$\widehat{\mathrm{Var}}\left(\{c_i\}\right) = \frac{1}{n-1}\sum_{i=1}^{n}\left(c_i - \overline{c}\right)^2,$$

where $\overline{c} \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} c_i$ is the sample mean.

d) Repeat b) and c) using another unit vector $\mathbf{u}_a = [0.8,\ 0.6]^T$. Is the newer sample variance smaller than that in b)? Can you explain why?

Problem 2 (Orthogonal and Unitary Matrices)

You have learned in class that a real-valued *orthogonal matrix* $\mathbf{P}$ has the property that $\mathbf{P}^T\mathbf{P} = \mathbf{P}\mathbf{P}^T = \mathbf{I}$. In signal processing, oftentimes signals are represented in complex values. We can define for a complex-valued square matrix a similar concept called the *unitary matrix*. A unitary matrix $\mathbf{Q}$ satisfies the property that $\mathbf{Q}^H\mathbf{Q} = \mathbf{Q}\mathbf{Q}^H = \mathbf{I}$, where "^{H}" is the Hermitian operator that is a combination of both the transpose, "^{T}," and the complex conjugate, "*."

a) The discrete Fourier transform (DFT) matrix for a signal of length 3 is defined as $\mathbf{Q}_3 = [q_{kn}]_{k,n\in\{0,1,2\}}$, where $q_{kn} = \frac{1}{\sqrt{3}}\exp(-j\frac{2\pi}{3}kn)$. Explicitly write out this 3-by-3 matrix. Simplify each entry but do not evaluate numerically the complex exponentials.

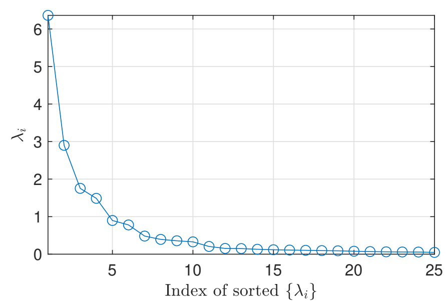
b) Verify that $\mathbf{Q}_3$ is a unitary matrix.

c) Compute the forward discrete Fourier transform for signal $\mathbf{x} = [1.01, 0.99, 0.97]^T$ to obtain a transformed signal $\mathbf{z} = \mathbf{Q}_3\,\mathbf{x}$, where $\mathbf{z} = [z_0, z_1, z_2]^T$. How large are the norms for z_0, z_1, and z_2?

d) Now, zero out z_2 to obtain a new vector $\mathbf{z}_{\text{compressed}}$. Compute the inverse transform using $\hat{\mathbf{x}} = \mathbf{Q}_3^H\,\mathbf{z}_{\text{compressed}}$. Is $\hat{\mathbf{x}}$ similar to $\mathbf{x}$? Can you guess why?

Problem 3 (PCA on Downsampled Yale Face Database) In this problem, we will explore PCA as a visualization tool for Yale Face Database. Download the `.m` files and the database. Extract the face image files into a folder named `yalefaces` and put the `.m` files at the same level of the folder. Call this folder "problem_pca_face/". Open Matlab, change your "Current Folder" to "problem_pca_face/", and open `main_pca_visualization.m`.

- **a)** Run the code corresponding to this part only, describe the data structure of variable `img_buffer`. Set `preview_img_flag` to `1`, re-run the code to visually inspect the whole database.
- **b)** Complete Matlab function `[V, Lambda_mat] = PcaViaKlt(data)` by implementing PCA using eigendecomposition on a sample covariance matrix of the face data. The detailed information about the input and outputs are given in the comments of the incomplete function. You may use built-in function `eig` for eigendecomposition. If your implementation is correct, after running the code of b), you will obtain a plot similar to the following.



- **c)** Run the code of c) to visualize a few dominating eigenvectors. Comment on whether they reflect some characteristics of the faces you saw in a).
- **d)** The code of d) projects each face image (coming from one of the four selected classes) onto a 2D space. Comment on PCA's data visualization performance in this specific example.

Problem 4 (Deep Learning with Matlab) In recent updates, Matlab has put together well-guided tutorials for deep learning. This is one set of tutorials on "Deep Learning with Images". Complete the following tutorials by running the example code:

"Classify Webcam Images Using Deep Learning."

Write a concise report consisting of key source code, images, and your explanations.

For more tutorials, see the left menu on this page: `https://www.mathworks.com/help/deeplearning/getting-started-with-deep-learning-toolbox.html`

These tutorials may give you ideas about your projects in other courses.

Group Study (1', bonus) Zoom: Take a screenshot of the whole team with everyone's webcam capturing his/her face. One of you will share the screen showing the specific homework assignment sheet that you are working on. In-Person: Take a selfie with all group members' faces in the photo. Capture the homework assignment sheet in the photo.

Include the screenshot/selfie in your own homework submission as the last "problem." Your screenshot/selfie gets you 1 bonus point; your group members need to do it separately to earn their bonus points.