ECE 269: Linear Algebra and Applications
Fall 2021

Homework # 4
Due: Friday, December 3 11:59pm,
via Gradescope

Collaboration Policy: This homework set allows limited collaboration. You are expected to try to solve the problems on your own. You may discuss a problem with other students to clarify any doubts, but you must fully understand the solution that you turn in and write it up entirely on your own. Blindly copying results from any resource (such as your friend, or the internet) will be considered a violation of academic integrity. *

Instructions for Programming Assignment: You can use MATLAB or Python for the programming assignment. You must submit the source code and a report with results and discussions. The source code must be presented in a way that it can be directly executed by the TAs. This can be done either by separately submitting the source code files, or embedding them in a Jupyter notebook. You cannot simply copy paste the source code into a .pdf or word file as part of the report.

---

1. **Problem 1: Moore–Penrose pseudoinverse.** A pseudoinverse of $\mathbf{A} \in \mathbb{R}^{m\times n}$ is defined as a matrix $A^+ \in \mathbb{R}^{n\times m}$ that satisfies

$$AA^+A = A, \quad A^+AA^+ = A^+$$

and AA^+ and A^+A are symmetric.

   (a) Show that A^+ is unique.

   (b) Show that $(A^TA)^{-1}A^T$ is the pseudoinverse and a left inverse of a full-rank tall matrix A

   (c) Show that $A^T(AA^T)^{-1}$ is the pseudoinverse and a right inverse of a full-rank fat matrix A.

   (d) Show that A^{-1} is the pseudoinverse of a full-rank square matrix A.

   (e) Show that A is the pseudoinverse of itself for a projection matrix A.

   (f) Show that $(A^T)^+ = (A^+)^T$.

   (g) Show that $(AA^T)^+ = (A^+)^TA^+$ and $(A^TA)^+ = A^+(A^+)^T$.

   (h) Show that $\mathcal{R}(A^+) = \mathcal{R}(A^T)$ and $\mathcal{N}(A^+) = \mathcal{N}(A^T)$.

   (i) Show that $P = AA^+$ and $Q = A^+A$ are projection matrices.

*For more information on Academic Integrity Policies at UCSD, please visit `http://academicintegrity.ucsd.edu/excel-integrity/define-cheating/index.html`

(j) Show that $y = Px$ and $z = Qx$ are the projections of x onto $\mathcal{R}(A)$ and $\mathcal{R}(A^T)$, respectively, where P and Q are defined as previously.

(k) Show that $x^* = A^+b$ is a least-squares solution to the linear equation $Ax = b$, i.e., $\|Ax^* - b\| \leq \|Ax - b\|$ for every other x.

(l) Show that $x^* = A^+b$ is the least-norm solution to the linear equation $Ax = b$, i.e., $\|x^*\| \leq \|x\|$ for every other solution x, provided that a solution exists.

2. **Problem 2: Eigenvalues.** Suppose that $A \in \mathbb{C}^{n\times n}$ has $\lambda_1, \lambda_2, \ldots, \lambda_n$ as its eigenvalues repeated according to their algebraic multiplicities.

(a) Show that $\det(A) = \lambda_1 \cdot \lambda_2 \cdots \lambda_n$.

(b) Show that $\text{Trace}(A) = \lambda_1 + \lambda_2 + \cdots + \lambda_n$.

(c) Show that the eigenvalues of A^k are $\lambda_1^k, \lambda_2^k, \ldots, \lambda_n^k$ for $k = 1, 2, \ldots$.

(d) Show that A is invertible if and only if it does not have a zero eigenvalue.

(e) Suppose that A is invertible. Show that the eigenvalues of A^{-1} are $\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_n^{-1}$.

3. **Problem 3: Nilpotent Matrices.** A matrix $A \in \mathbb{C}^{n\times n}$ is said to be nilpotent if $A^k = 0$ for some integer k.

(a) Show that all eigenvalues of a nilpotent matrix must be 0.

(b) Show that the smallest k for which $A^k = 0$, satisfies $k \leq n$. (*Hint: Use Cayley Hamilton*)

(c) Suppose that the smallest k for which $A^k = 0$, is $k = n$. Further suppose that $A^{n-1}x \neq 0$. Then show that $\{x, Ax, A^2x, \cdots, A^{n-1}x\}$ form a basis of $\mathbb{C}^n$.

4. **Problem 4: Spectral Norm.**

(a) Show that $\|A^HA\| = \|A\|^2$.

(b) Show that the spectral norm is *unitarily invariant*, namely, $\|UAV\| = \|A\|$ for any unitary matrices U and V.

(c) Show that

$$\left\| \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} \right\| = \max(\|A\|, \|B\|).$$

5. **Problem 5: Properties of PSD matrices.** Let $A = A^T \in \mathbb{R}^{n\times n}$ and $B = B^T \in \mathbb{R}^{n\times n}$. Prove the following statements.

(a) If $A \succeq 0$ and $B \succeq 0$, then $\text{Trace}(AB) \geq 0$.

(b) If $A \succeq 0$, then $A + B \succeq B$.

(c) If $A \succeq B$, then $-B \succeq -A$.

(d) If $A \succeq I$, then $I \succeq A^{-1}$.

(e) If $A \succeq B \succ 0$, then $B^{-1} \succeq A^{-1} \succ 0$.

Programming Assignment:
Face Recognition Using Eigenfaces and Principal Component Analysis

Please read the following paper which shows how to use Principal Component Analysis (PCA) for Face Recognition (uploaded on Canvas).

M. A Turk and A. P. Pentland, "Face Recognition Using Eigenfaces", Proceedings of IEEE CVPR 1991.

The paper puts forward a simple yet effective idea of using eigenfaces (obtained via PCA) to perform unsupervised face recognition. Read and understand the basic principle, and then conduct the following numerical experiments to implement and test the eigenface-based face recognition algorithm.

Data: The dataset ("**eigenface-dataset.zip**" uploaded on Canvas) consist of faces of 171 people and each person has two frontal images (one with a neutral expression, labeled with suffix letter "**a**" and the other with a smiling facial expression, labeled with suffix letter "**b**"), there are 342 full frontal face images manually registered and cropped.

Implementation and Experiments:

1. Compute the principal components (PCs) using 100 individuals', randomly selected neutral expression training images. Display the ten most representative eigenfaces. Plot the singular values of the data matrix and justify your choice of number of principal components. Repeat this procedure using the same 100 individuals' smiling expression training images.

2. Reconstruct one of 100 individuals' neutral expression image using different number of PCs you computed from Part 1. As you vary the number of PCs, display your reconstructed images and plot the mean squared error (MSE) of reconstruction versus the number of principal components to show the accuracy of reconstruction. Comment on your result.
   (**Note**: To calculate MSE and compare the results, you will need to make sure your reconstructed image is on the same numerical scale as the ground truth image across different PCs.)

3. Reconstruct one of 100 individuals' smiling expression image using different number of PCs. Again, display your reconstructed images and plot the MSE of reconstruction versus the number of principal components and comment on your result.

4. Now consider the remaining 71 individuals. Reconstruct one of the other 71 individuals' neutral expression image (this image is not in the training set of 100 images used in Part 1-3) using different number of PCs. Again, display your reconstructed images and plot the MSE of reconstruction versus the number of principal components and comment on your result. Repeat this procedure using corresponding individual's smailing expression image.

5. **Facial Expression Classification**: Now we want to perform facial expression classification on these images. Since the given dataset only contains two types of facial expressions, we label the smiling image as 0 and neutral image as 1. We are going to use the PCs trained from Part 1 as our eigenfaces.
   a). Generate your testing set: Start by randomly selecting 60 individuals that are not in the training set.
   b). Then new face images are first projected into "smiling face space" and "neutral face space" using the PCs you obtained from Part 1.
   c). Follow the reconstruction steps you performed earlier, reconstruct the images and calculate the MSE with its groundtruth. Since we have two classes here, you will obtain two reconstructed images per test image that were projected onto smiling face space and neutral face space.
   d). Compare the two MSEs. The face space with a smller MSE will be declared as the class of the test image.
   e). Repeat b)-d) for all 60 images. Report your classification accuracy rate for two classes separately.
   f). Pick one mistakenly labeled image from each class and display them in your report. Why do you think your eigenface algorithm fails to label this image? Any suggestions to improve this?