Dr. Nadim Ghaddar
University of Toronto
Tuesday, October 1, 2024
ECE367 - Matrix Algebra and Optimization

Problem Set #3
Due Date: Tuesday, October 15, 2024, at 11:59 PM

Homework policy: Problem sets must be turned by the due date. In the following, the course text "Optimization Models" is abbreviated as "OptM" and "Introduction to Applied Linear Algebra" as "IALA".

Problems are categorized as:

- **"Theory" problems:** These are mostly mathematical questions designed to give you deeper insight into the fundamentals of the ideas introduced in this class.
- **"Application" problems:** These questions are designed to expose you to the breadth of application of the ideas developed in class and to introduce you to useful numerical toolboxes. Problems of this sort often ask you to produce plots and discuss your results; said plots and discussions should be included in your submission – think of your submitted solution like a lab book. Your attached code simply provides back-up evidence.
- **"Optional" problems:** Optional problems provide extra practice or introduce interesting connections or extensions. They need not be turned in. We will not grade them, but we will assume you have reviewed and understood the solutions to the optional problems when designing the exams.

Submission and Grading:

- **Submission:** Your submission of the "Theory" and "Application" questions must be uploaded via Quercus by the due date.
- **Grading:** We will grade one problem from the "theory" problems and one problem from the "application" problems. Your score will be the sum of the two grades.
- **For fairness, late submissions will not be accepted!**

Theory Problems

1. *Practice computing eigenvalues and eigenvectors.* In this problem you consider the eigenvalues and eigenvectors of each of the following four matrices:

   (a) $A = \begin{bmatrix} 3 & 1 \\ 2 & 2 \end{bmatrix}$, (b) $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, (c) $A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, (d) $A = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$.

   For each of the matrices in parts (a)-(d) do the following:

   (i) compute the eigenvalues of the matrix,

   (ii) compute the eigenvectors of the matrix,

   (iii) specify both the algebraic and geometric multiplicity of each distinct eigenvalue, and

   (iv) *if* the matrix is diagonalizable express the matrix in its diagonalized form. In other words, if A is diagonalizable express it as $A = V\Lambda V^{-1}$ where V is the matrix of eigenvectors and Λ is a diagonal matrix of eigenvalues. Specify V and Λ for each matrix that is diagonalizable.

2. *Eigenvectors of a symmetric* 2×2 *matrix.* OptM Book, Exercise 4.1.

3. *Quadratic constraints.* OptM Book, Exercise 4.2. (In this problem, it might be helpful to consider the spectral decomposition of A).

4. *Lower bound on the rank.* OptM Book, Exercise 4.9, parts 1 and 2 only (not part 3). For these parts, you only need to assume that A is a symmetric matrix. (The following two facts might be useful to solve this problem: 1) For any two matrices A and B, $\text{trace}(AB) = \text{trace}(BA)$, and 2) The rank of a square matrix is equal to the number of non-zero eigenvalues.)

Application Problems

1. *Google's PageRank algorithm.* In this problem, you will implement and analyse approaches to the eigenvector computation that is at the heart of Google's PageRank algorithm (cf. discussion in Examples 3.5 and 7.1 of OptM). Download `pagerank_urls.txt` and `pagerank_adj.mat` files from the course website. The first is a plain text file in which each line consists of a URL of a web page. We refer to the web pages by their respective URL-line numbers starting from 1. For example, web page 9 is the page that the URL in line 9 points to. The URLs are of the internal web pages of Hollins University in Roanoke, Virginia. The provided data files consist of a modified version of web crawling data downloaded from `https://www.limfinity.com/ir`. The original dataset has been created in January, 2004 therefore you might notice that some of the links are inactive.

   Let N be the number of URLs (therefore the number of web pages) in the first file and $i, j \in [N]$. Execute the command `load 'pagerank_adj.mat'` in MATLAB to load the content of the second file into your MATLAB workspace. You will see that a new $N \times N$ variable `J` has been imported. This variable represents an *adjacency matrix* $J \in \{0,1\}^{N\times N}$ which describes the relationships between the web pages. Specifically, the element in ith row and jth column $J_{i,j} = 1$ if there exists a link from web page j to web page i, and $J_{i,j} = 0$ otherwise. Data have been carefully filtered so that $J_{j,j} = 0$ and $\sum_{i=1}^{N} J_{i,j} > 0$ for all $j \in [N]$. In other words, we do not allow links from a web page to itself, and we do not allow for dangling pages, that is pages with no outgoing links.

   Use J to obtain the *link matrix* A where

$$A_{i,j} = \frac{J_{i,j}}{\sum_{k=1}^{N} J_{k,j}}.$$

   Also, let $x \in \mathbb{R}^N$ be a vector with all entries equal to 1. Use the matrix A and the vector x the same way as described in Examples 3.5 and 7.1 to solve the following.

   (a) Verify that each column in the provided matrix A sum to 1. What is the importance of this property for the Google PageRank algorithm?

   (b) In the following we are consistent with the notation used in OptM. Let $x(k+1)$ be the approximation of the eigenvector in the $k+1$th iteration. We define the approximation error in the $k+1$th iteration as $e(k+1) = \|Ax(k+1) - x(k+1)\|_2$. Using MATLAB, implement the power iteration algorithm described in Section 7.1.1 in OptM. Run the algorithm for 10 iterations and plot $\log(e(k+1))$ versus k.

   (c) Implement the shift-invert power iteration and Rayleigh quotient iteration algorithms presented in Sections 7.1.2 and 7.1.3 of OptM. For the shift-invert power method use $\sigma = 0.99$. In the Rayleigh quotient iteration method, use $\sigma_1 = \sigma_2 = 0.99$ for the first two iterations and $\sigma_k = \frac{x^*(k)Ax(k)}{x^*(k)x(k)}$ for $k > 2$, in a similar manner as the discussion in Example 7.1. Repeat your experiment of part (b) for these two algorithms. Plot $\log(e(k+1))$ for each of your three algorithms on a single plot. Check whether your results are consistent with Example 7.1. Include your MATLAB code with the answers.

   (d) List the (page index, PageRank score) pairs of the top 5 and bottom 5 pages according to your PageRank scores. Compare them with the provided web page links and briefly explain whether the ranking seems intuitively correct.

(Note: While we write "MATLAB" in the above, you are, as usual, welcome to use any software you would like to, just not to use built-in functions that accomplish the objective.)

2. *Eigenfaces and ℓ_2 projection.* In this problem, you will familiarize with the concept of Eigenfaces and its uses. Download the dataset `yalefaces.mat` from the course website. This dataset consists of 32×32 gray scale images of faces taken from the *Extended Yale Face Database B* (`http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html`). Load the dataset into your MATLAB environment by executing the command `load('yalefaces.mat')`. You will see a new variable `M` of dimension $32 \times 32 \times 2414$ which consists of 2414 grayscale images, 32×32 pixels each. The pixel values of the images range from 0 to 255. You can view the loaded images by making use of MATLAB built-in functions `imshow` or `imagesc`. As an example, the first image of the dataset can be displayed by executing `imshow(M(:,:,1)/255)`.

   Let N be the number of images in the dataset and let $d = 1024$, the total number of pixels in each image. An image can be thought of as a matrix with 32 columns and 32 rows consisting of entries in the range $[0, 255]$. In this exercise we consider the images in vector form. Let $\mathcal{J}_i$ be the ith image in the dataset where $i \in [N]$. We formulate a column vector $x^{(i)} \in \mathbb{R}^d$ by flattening the matrix that makes up $\mathcal{J}_i$. In our case we stack all 32 columns vertically to form a d-dimensional vector $x^{(i)}$. In MATLAB you can do this efficiently using the command `reshape`. The 'average face' vector of the dataset $\bar{x}$ can be computed as $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x^{(i)}$. We can (roughly) visualize the $x^{(i)}$s as forming a cloud of data points where the cloud is centered at $\bar{x}$ in d-dimensional space. Translate the cloud center to the origin by subtracting $\bar{x}$ from each $x^{(i)}$. Let the resulting vectors be denoted as $\bar{x}^{(i)} = x^{(i)} - \bar{x}$, which we will refer to as centered image vectors. Construct a matrix $X \in \mathbb{R}^{d \times N}$ by concatenating all the centered vectors $\bar{x}^{(i)}$, i.e., $X = \left[\bar{x}^{(1)}, \bar{x}^{(2)}, \ldots, \bar{x}^{(N)}\right]$. The matrix $C = XX^T$ is N times the covariance matrix of the data samples $x^{(i)}, i \in [N]$.

   (a) Compute the eigenvalue/eigenvector pairs of C and arrange them in decreasing order of the magnitude of the eigenvalues. Let the jth pair in the ordered set be denoted as $\lambda_j \in \mathbb{R}, v^{(j)} \in \mathbb{R}^d$ for $j \in [d]$. You may find the MATLAB command `eig` helpful. Comment whether the eigenvalues are real and briefly explain why. Plot $\log \lambda_j$ against $j \in [d]$. Include your plot in your solution.

   (b) The set of eigenvectors you computed in the previous part form a basis for the centered image vectors. Reshape each eigenvector $v^{(j)}$ to obtain a 32×32 matrix. These matrices can be considered as images and they are known as *eigenfaces*. Include plots of two sets of eigenfaces, those corresponding to the largest 10, and those corresponding to the smallest 10, eigenvalues. Do you observe any difference between the two sets of eigenfaces you plotted? If so briefly explain the reason for this difference.

   (c) In this part, consider the images $\mathcal{J}_i$ for $i \in \{1, 1076, 2043\}$. Let $\mathcal{B}_j = \{v^{(1)}, v^{(2)}, \ldots, v^{(j)}\}$ where $j = \{2^1, 2^2, 2^3, \ldots, 2^{10}\}$, i.e., j indexes sets each consisting of a different number of the eigenvectors. Let us denote the ℓ_2 projection of $\bar{x}^{(i)}$ onto the basis vector set $\mathcal{B}_j$ as $\bar{y}^{(i,j)}$. Compute $\bar{y}^{(i,j)}$ for the given i, j pairs using MATLAB. (I.e., do this using your numerical tool and not by hand.) Note that $\bar{y}^{(i,j)}$ vectors are computed using the centered image vectors. Translate these vectors back (un-center them) by the cloud center shift $\bar{x}$ to get the image vectors $y^{(i,j)} = \bar{y}^{(i,j)} + \bar{x}$. Reshape the $y^{(i,j)}$ vectors into 32×32 matrices and plot them as images. Note that you will need to plot 30 images and these can be compactly plotted using `subplot` command in MATLAB.

   (d) In this part you will learn how the eigenfaces can be used for the task of *face recognition*.

Consider the two sets of indices $\mathcal{I}_1 = \{1, 2, 7\}$ and $\mathcal{I}_2 = \{2043, 2044, 2045\}$. The faces in the set $\mathcal{J}_i$ for $i \in \mathcal{I}_1$ belong to one person and those for $i \in \mathcal{I}_2$ belong to a second person. We have carefully picked the image indices so that the corresponding images are taken under similar lighting conditions. Consider $\mathcal{B}_{25}$ where $\mathcal{B}_j$ is defined as in part (c). Compute the projection coefficients obtained by projecting $\bar{x}^{(i)}, i \in \mathcal{I}_1 \cup \mathcal{I}_2$ onto the eigenvectors in $\mathcal{B}_{25}$. Let $c^{(i)} \in \mathbb{R}^{25}$ be the vector that consists of coefficients obtained for ith image. The vector $c^{(i)}$ can be thought of as a representation of the corresponding original image $\mathcal{J}_i$. Intuitively, images that belong to same person should have similar coefficient vectors. To verify this, compute the pairwise Euclidean distances between the $c^{(i)}$ vectors. Tabulate the values and comment whether the distance between any two $c^{(i)}$ vectors that belong to the same person is smaller than those belonging to the other person. Briefly explain how you can use this to build a simple face recognition scheme.

Optional Problems

1. *Ellipses, eigenvalues, eigenvectors, and volume.* Make neat and clearly-labelled *sketches* (i.e., draw by hand) of the ellipsoid $\mathcal{E} = \{x | (x - x_c)^T P^{-1} (x - x_c) = 1\}$ for the following sets of parameters:

   (a) Center $x_c = [0\ 0]^T$ and $P = [1.5\ \ -0.5; -0.5\ 1.5]$.

   (b) Center $x_c = [1\ \ -2]^T$ and $P = [3\ 0; 0\ 1]$.

   (c) Center $x_c = [-2\ 1]^T$ and $P = [9\ \ -2; -2\ 6]$.

   For each part (a)–(c) also compute each pair of eigenvalues and corresponding eigenvectors.

   (d) Recall the geometrically meaningful property of the determinant of a square real matrix A: its magnitude $|\det A|$ is equal to the volume of the parallelepiped $\mathcal{P}$ formed by applying A to the unit cube $\mathcal{C} = \{x | \, 0 \leq x_i \leq 1, i \in [n]\}$. In other words, if $\mathcal{P} = \{Ax | x \in \mathcal{C}\}$ then $|\det(A)|$ is equal to the volume of $\mathcal{P}$. Furthermore, recall that the determinant of a matrix is zero if any of its eigenvalues are zero. Explain how to interpret this latter fact in terms of the interpretation of $|\det(A)|$ as the volume of $\mathcal{P}$.