Department of Mathematics, University of Utah

Analysis of Numerical Methods I

MATH 6610 – Section 001 – Fall 2020

Homework 1

Basic linear algebra and eigenvalues

Due Friday, September 18, 2020 by 11:59pm MT

Submission instructions:
Create a private repository on `github.com` named `math6610-homework-1`. Add your LaTeX source files and your Matlab/Python code and push to Github. To submit: grant me (username `akilnarayan`) write access to your repository.
You may grant me write access before you complete the assignment. I will not look at your submission until the due date+time specified above. If you choose this route, I will only grade the assignment associated with the last commit before the due date.
<u>All commits timestamped after the due date+time will be ignored</u>
<u>All work in commits before the final valid timestamped commit will be ignored.</u>

Problem assignment:
Trefethen & Bau III, Lecture 1: # 1.3
Trefethen & Bau III, Lecture 2: # 2.1, 2.2, 2.6
Trefethen & Bau III, Lecture 3: # 3.1, 3.2, 3.3
Trefethen & Bau III, Lecture 24: # 24.1, 24.2 (a and c)

P1. (Inner and outer products) Let $A \in \mathbb{R}^{K\times n}$ and $B \in \mathbb{R}^{n\times L}$ be given matrices. The matrix product $AB \in \mathbb{R}^{K\times L}$ has entries

$$(AB)_{k,\ell} = \sum_{q=1}^{n} (A)_{k,q}(B)_{q,\ell}, \qquad\qquad k = 1,\ldots,K,\ \ell = 1,\ldots,L.$$

Using only this definition, prove the following:

(a) If $R \in \mathbb{R}^{K\times n}$ has rows $\{r_k\}_{k=1}^{K}$ and $C \in \mathbb{R}^{n\times L}$ has columns $\{c_\ell\}_{\ell=1}^{L}$, then

$$(RC)_{k,\ell} = r_k^T c_\ell.$$

(b) If $R \in \mathbb{R}^{n\times L}$ has rows $\{r_j\}_{j=1}^{n}$ and $C \in \mathbb{R}^{K\times n}$ has columns $\{c_j\}_{j=1}^{n}$, then

$$CR = \sum_{j=1}^{n} c_j r_j^T.$$

(c) What is the maximum possible rank of CR from part b? Justify your answer.

P2. (Unitary matrices and norms) Let $U \in \mathbb{C}^{n\times n}$ be a unitary matrix. Prove the following:

(a) $\|Ux\|_2 = \|x\|_2\ \forall x \in \mathbb{C}^n$

(b) $\|U\|_2 = 1$

(c) $\|U\|_F = \sqrt{n}$

(d) $\|UA\|_2 = \|A\|_2$ for arbitrary $A \in \mathbb{C}^{n\times m}$ and arbitrary $m \in \mathbb{N}$
(e) $\|UA\|_F = \|A\|_F$ for arbitrary $A \in \mathbb{C}^{n\times m}$

Above $\|\cdot\|_2$ is the 2-norm on vector or the induced 2-norm on matrices, and $\|\cdot\|_F$ is the Frobenius norm.

P3. (Norms of projections) Let $M \in [1,\infty)$ and $n \geq 2$ be arbitrary. Explicitly construct a projection matrix $P \in \mathbb{R}^{n\times n}$ such that $\|P\|_2 = M$.

P4. Prove the spectral theorem for skew-Hermitian matrices: If $A \in \mathbb{C}^{n\times n}$ satisfies $A = -A^*$, then A is unitarily diagonalizable with purely imaginary eigenvalues.

P5. (Permutation matrices) Let $P \in \mathbb{R}^{n\times n}$ be a permutation matrix.

(a) Prove that P^T is also a permutation matrix.
(b) Prove that $P^T = P^{-1}$.
(c) Prove that if P_1 and P_2 are both permutation matrices, then P_1P_2 is also a permutation matrix.
(d) Is it true in general that $P^2 = P$? If so, prove it. If not, give a counterexample.
(e) Suppose $P \in \mathbb{C}^{n\times n}$ is both a projection matrix and a permutation matrix. Show that such a matrix P is unique, and explicitly compute its entries.

P6. (Commutativity and simultaneous diagonalizability) Two square matrices M_1 and M_2 *commute* if $M_1M_2 = M_2M_1$. Two square matrices M_1 and M_2 are *simultaneuosly diagonalizable* if there is a single square, invertible matrix V such that

$$M_1 = V\Lambda_1V^{-1}, \qquad\qquad M_2 = V\Lambda_2V^{-1},$$

where Λ_1 and Λ_2 are diagonal matrices. Assume that square matrices A and B are both ("individually") diagonalizable, and that all the eigenvalues of A and B are simple. Prove that they are simultaneously diagonalizable if and only if they commute. (The simple eigenvalue assumption is actually not needed, but the proof in this case is more involved.)

P7. (Generalized eigenvalue problems) Assume that A and B are $n\times n$ Hermitian matrices, and that B is positive-definite. Consider the following *generalized* eigenvalue problem,

$$\text{Find } \lambda \in \mathbb{C},\ v \in \mathbb{C}^n\backslash\{0\} \text{ such that } Av = \lambda Bv$$

Prove the following:

(a) There are n real-valued (generalized) eigenvalues and n linearly independent (generalized) eigenvectors.
(b) The n eigenvectors $\{v_j\}_{j=1}^n$ are B-orthogonal: $v_j^* B v_k = \delta_{j,k}$.
(c) The Courant-Fischer-Weyl min-max principle for this problem is that the eigenvalues $\{\lambda_j\}_{j=1}^n$, ordered such that $\lambda_1 \leq \lambda_n$, satisfy

$$\lambda_k = \max_{\substack{V\subset\mathbb{C}^n \\ \dim V = n-k+1}} \min_{x\in V} R_{A,B}(x), \qquad\qquad R_{A,B}(x) := \frac{\langle Ax, x\rangle}{\langle Bx, x\rangle}$$

P8. (Principal Component Analysis) Let $B \in \mathbb{C}^{m\times n}$ be comprised of columns b_j, i.e.,

$$B = \begin{pmatrix} | & | & & | \\ b_1 & b_2 & \cdots & b_n \\ | & | & & | \end{pmatrix}.$$

In this problem, we will view each column as a single piece of data in $\mathbb{C}^m$-dimensional space. In particular, we'll consider these columns as random realizations from some probability distribution on vectors. Under this model, the (empirical) mean of our set of realizations is

$$b_0 \coloneqq \frac{1}{n} \sum_{j=1}^{n} b_j,$$

which is the "average" data point.

The ultimate goal of Principal Component Analysis (PCA) is to define a compression matrix $Q \in \mathbb{C}^{m\times r}$ such that the compressed matrix $Q^* B$ contains as much of the "variance" of B as possible.

In particular, we consider the unnormalized variance of B when each data column is projected onto the direction $x \in \mathbb{C}^m \backslash \{0\}$:

$$\mathrm{var}_x(B) \coloneqq \mathrm{var}\,(\hat{x}^* B), \qquad\qquad \hat{x} \coloneqq \frac{x}{\|x\|_2},$$

where $\mathrm{var}\,(\cdot)$ on row vectors is the unnormalized variance,

$$\mathrm{var}\,(y) \coloneqq (y - \bar{y})\,(y - \bar{y})^* \,.$$

We will call $\mathrm{var}_x(B)$ the (unnormalized) "directional variance" of B in the direction x; a normalized variance only differs by a $\frac{1}{n-1}$ multiplicative factor. Qualitatively, $\mathrm{var}_x(B)$ measures the amount of "information" or "energy" that the dataset contains in the direction of $x \in \mathbb{C}^m$. Given a subspace $W \subset \mathbb{C}^m$, the smallest directional variance within W is

$$\min_{x\in W} \mathrm{var}_x(B).$$

Restating the goal of PCA: Given some rank $r \in \{1, \ldots, m\}$, find a subspace W whose smallest directional variance is as large as possible. I.e., to find W that maximizes the variance when B is projected into W. Below, $R_C(\cdot)$ is the Rayleigh quotient of a square, symmetric matrix C.

(a) Show that $\mathrm{var}_x(B) = R_{AA^*}(x)$, where $A \coloneqq B - b_0\, 1_{1\times n}$, where $1_{1\times n}$ is an n-dimensional row vector with all entries equal to 1.

(b) For a fixed $r \in \{1, \ldots, m\}$, prove that the subspace V that maximizes the smallest directional variance within a rank-r subspace,

$$V \coloneqq \operatorname*{argmax}_{\substack{W\in\mathbb{C}^m \\ \dim W = r}} \min_{x\in W} \mathrm{var}_x(B),$$

is given by $\mathrm{span}\{v_n, v_{n-1}, \ldots, v_{n-r+1}\}$, where $\{(\lambda_j, v_j)\}_{j=1}^m$ are the eigenpairs of AA^* ordered as $\lambda_j \le \lambda_{j+1}$. (The matrix AA^* is called the unnormalized covariance of the data.)

Thus, if each v_j is unit-length, the sought PCA compression matrix is $Q = [v_n \ldots, v_{n-r+1}] \in \mathbb{C}^{m\times r}$. The vector v_n is typically called the first principal component; v_{n-1} the second principal component; etc.

Computing assignment:

C1. (PCA and Eigenfaces) In a programming language of your choice, compute and plot principal components for the Yale Face Database:
`http://vision.ucsd.edu/content/yale-face-database`
For this problem, grayscale images can be viewed as a matrix of numbers, where the dimensions of the matrix correspond to the pixel dimensions of the image. In order to treat a given image as a single data point (b_j in problem P8), vectorize its matrix representation (i.e., unwind the matrix into a vector). Plot the first three principal components for the subset of the data that are "normal" faces. Repeat this experiment for the "glasses" faces, and finally for all the facial images. Comment on the results that you see. For this type of experiment, the principal components are frequently called "eigenfaces".

`Note`: "Plotting" principal components does not mean plotting a vector. Plot them as images, as you would the original data. They should be interpretable. This exercise requires you to import and manipulate data from hard disk, and also to load and manipulate image data, so you will have to learn how to accomplish this in your language of choice. Finally, submit your code, but <u>not</u> the Yale Face Database data files. In your code provide an easy way for me to point to a particular folder on my local hard drive that contains the data.