Matrix Analysis, Spring 2018

(Due on June 3, 2018 at 11:59pm)

Submission Instructions

We do not do reverse engineering, so please DO NOT provide MATLAB (or other programming language) codes WITHOUT **method description**. You should also declare in the assignment that **the MATLAB (or other programming language) code was written by you, not by others either partially or fully**.

You should submit your answers as a write-up in PDF format to matrix_analysis_18@163.com. The email title is formatted as “prog_asgmt_学号_姓名”.

If you are in doubt, talk to me (majh8@mail.sysu.edu.cn) or our teaching assistant (孙淑婷, sunsht3@mail2.sysu.edu.cn) to understand more.

QR Algorithm

Please implement the QR algorithm as an Eigen-Decomposition function and provide the code for the implementation (10%). The QR algorithm is described on Page 108 in our textbook《矩阵论简明教程》, i.e., let $A_1 = A$, perform QR decomposition for $A_k = Q_k R_k$ (e.g., the “qr” function in MATLAB could be used), update $A_{k+1} = R_k Q_k$, until A_{k+1} is close to an upper triangle matrix. Let us set the tolerance as $\varepsilon = 10^{-10}$ (or please specify ε in each of the following questions if it is set as another small positive number), i.e., if $\left|a_{ij}^{(k)}\right| < \varepsilon$ for any $i > j$, the iteration stops.

1. Consider the following matrices,

$$A = \begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix}, \qquad B = \begin{pmatrix} 2 & 3 & 4 & 5 & 6 \\ 4 & 4 & 5 & 6 & 7 \\ 0 & 3 & 6 & 7 & 8 \\ 0 & 0 & 2 & 8 & 9 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \qquad H_6 = \begin{pmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{7} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{6} & \frac{1}{7} & \cdots & \frac{1}{11} \end{pmatrix}$$

   a) Please show that the matrix sequence obtained by the QR algorithm $\{A_k\}$ converges to an upper triangle matrix for the above matrices. (Results, 15%)

   b) Please compute the eigenvalues and matrix 2-norm condition number (i.e., $\mathrm{cond}_2(A) = \|A\|_2 \|A^{-1}\|_2$) for each of the above matrices. (Please indicate how to calculate the matrix 2-

norm condition number for each matrix and provide the results, 20%)

2. **Principal Component Analysis (PCA) for dimensionality reduction in face analysis applications.**

Download from the course website the data file "yale_face.mat". Load it by the command load('yale_face.mat'). It contains a matrix $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m) \in \mathbb{R}^{n \times m}$, where n is the number of image pixels and m is the number of face images. Each column of X is a 64×64 black-and-white face image resized to a vector. For example, use the command "imshow(reshape(X(:,1),[64 64]),[])" to view the first face image $\boldsymbol{x}_1$.

Your task is to perform PCA on this face image set. In the first step, each feature vector of the face image is centralized at the origin, i.e., compute $\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - \boldsymbol{\mu}$, where $\boldsymbol{\mu} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x}_i$. In the second step, the covariance matrix Σ is computed by $\Sigma = \frac{1}{\mathrm{m}}\sum_{i=1}^{m} \tilde{\boldsymbol{x}}_i \tilde{\boldsymbol{x}}_i^T = \frac{1}{\mathrm{m}} \tilde{X}\tilde{X}^T$. In third step, eigenvalues and eigenvectors are computed for the covariance matrix Σ. Finally, the eigenvectors are sorted by the eigenvalues in descend order. The first k sorted eigenvectors are selected as k principal component for dimensionality reduction.

a) In this problem, the number of image pixels n is much larger than the number of face images m, so we perform eigen-decomposition for $\frac{1}{\mathrm{m}}\tilde{X}^T\tilde{X}$ instead of the covariance matrix Σ, i.e., $\frac{1}{\mathrm{m}}\tilde{X}^T\tilde{X} = VDV^T$, where D is a diagonal matrix with the eigenvalues of $\frac{1}{\mathrm{m}}\tilde{X}^T\tilde{X}$ as the diagonal elements and V is an orthogonal matrix whose columns are eigenvectors. Please show that the diagonal elements of D contain all the non-zero eigenvalues of the covariance matrix Σ and the columns of $\tilde{X}V$ are eigenvectors of the covariance matrix Σ and orthogonal to each other. (Prove, 15%)

b) Please compute the eigenvalues and eigenvectors of $\frac{1}{\mathrm{m}}\tilde{X}^T\tilde{X}$ by the QR algorithm mentioned at the beginning of this assignment. Please calculate how long it will take for eigen-decomposition of $\frac{1}{\mathrm{m}}\tilde{X}^T\tilde{X}$ by the QR algorithm (e.g., "tic" and "toc" can be used in MATLAB). Please write out the largest 5 eigenvalues and show the corresponding eigenvectors of the covariance matrix Σ as five 64×64 black-and-white images. (Code and results, 20%)

c) Before performing the QR algorithm for $\frac{1}{\mathrm{m}}\tilde{X}^T\tilde{X}$, let us convert $\frac{1}{\mathrm{m}}\tilde{X}^T\tilde{X}$ to a tridiagonal matrix H, i.e., $\frac{1}{\mathrm{m}}\tilde{X}^T\tilde{X} = PHP^T$ where P is an orthogonal matrix (e.g., P and H can be calculated by “[P, H] = hess(A)” in MATLAB). Please report the time for such computation using triangular matrix H. Please write out the largest 5 eigenvalues and show the corresponding eigenvectors of the covariance matrix Σ as five 64×64 black-and-white images. Please compare the results with those obtained in b). (Code and results, 20%)