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](mailto: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](mailto:majh8@mail.sysu.edu.cn)) or our teaching assistant (孙淑婷, [sunsht3@mail2.sysu.edu.cn](mailto: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 , perform QR decomposition for  (e.g., the “qr” function in MATLAB could be used), update , until is close to an upper triangle matrix. Let us set the tolerance as  (or please specify in each of the following questions if it is set as another small positive number), i.e., if for any , the iteration stops.

1. Consider the following matrices,

, ,

* 1. Please show that the matrix sequence obtained by the QR algorithm  converges to an upper triangle matrix for the above matrices. (Results, 15%)
  2. Please compute the eigenvalues and matrix 2-norm condition number (i.e., ) 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%)

1. **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 , where  is the number of image pixels and is the number of face images. Each column of is a  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 .

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 , where . In the second step, the covariance matrix is computed by . 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  sorted eigenvectors are selected as principal component for dimensionality reduction.

* 1. In this problem, the number of image pixels is much larger than the number of face images , so we perform eigen-decomposition for  instead of the covariance matrix , i.e., , where  is a diagonal matrix with the eigenvalues of  as the diagonal elements and  is an orthogonal matrix whose columns are eigenvectors. Please show that the diagonal elements of contain all the non-zero eigenvalues of the covariance matrix and the columns of  are eigenvectors of the covariance matrix and orthogonal to each other. (Prove, 15%)
  2. Please compute the eigenvalues and eigenvectors of  by the QR algorithm mentioned at the beginning of this assignment. Please calculate how long it will take for eigen-decomposition of  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 as five black-and-white images. (Code and results, 20%)
  3. Before performing the QR algorithm for , let us convert  to a triangular matrix , i.e., where  is an orthogonal matrix (e.g., and can be calculated by “[P, H] = hess(A)” in MATLAB). Please report the time for such computation using triangular matrix . Please write out the largest 5 eigenvalues and show the corresponding eigenvectors of the covariance matrix as five black-and-white images. Please compare the results with those obtained in b). (Code and results, 20%)
  4. 