Assignment 4: CS 663

Due: 12th October before 11:55 pm

Remember the honor code while submitting this (and every other) assignment. You may discuss broad ideas with other students or ask me for any difficulties, but the code you implement and the answers you write must be your own. We will adopt a zero-tolerance policy against any violation.

Submission instructions: Follow the instructions for the submission format and the naming convention of your files from `http://www.cse.iitb.ac.in/~suyash/cs663/submissionStyle.pdf`. However, please do not submit the face image databases in your zip file that you will upload on moodle. Please see `http://www.cse.iitb.ac.in/~ajitvr/CS663_Fall2017/HW4/assignment4_SVD_FaceRecognition.rar`. Upload the file on moodle before 11:55 pm on 12th October. Policy for late submissions will be the same as in the aforementioned guidelines document. Please preserve a copy of all your work until the end of the semester.

1. In this part, you will implement a mini face recognition system. Download the ORL face database from `http://www.cl.cam.ac.uk/Research/DTG/attarchive/pub/data/att_faces.zip`. It contains 40 sub-folders, one for each of the 40 subjects/persons. For each person, there are ten images in the appropriate folder named 1.pgm to 10.pgm. The images are of size 92 by 110 each. Each image is in the pgm format. You can view the images in this format, either through MATLAB or through image viewers like IrfanView on Windows, or xv/display/gimp on Unix. Though the face images are in different poses, expressions and facial accessories, they are all roughly aligned (the eyes are in roughly similar locations in all images). For the first part of the assignment, you will work with the images of the first 32 people. For each person, you will include the first six images in the training set (that is the first 6 images that appear in a directory listing as produced by the dir function of MATLAB) and the remaining four images in the testing set. You should create an eigen-space (using the 'eig' function of MATLAB) from the training set as described during the lectures without explicitly computing the covariance matrix but instead using the $\boldsymbol{L}$ matrix as defined in class. Record the recognition rate using squared difference between the eigencoefficients while testing on all the images in the test set, for $k \in \{1, 2, 3, 5, 10, 15, 20, 30, 50, 75, 100, 150, 170\}$. Plot the rates in your report in the form of a graph. Now modify the required few lines of the code but using the 'svd' function of MATLAB instead of 'eig'.

   Repeat the same experiment (using just the svd routine) on the Yale Face database from `http://www.cse.iitb.ac.in/~ajitvr/CS663_Fall2016/HW4/CroppedYale/`. This database contains 60 images each of 38 individuals (labeled from 1 to 39, with number 14 missing). Each image is in pgm format and has size 192 by 168. The images are taken under different lighting conditions but in the same pose. Take the first 40 images of every person for training and test on the remaining 20 images (that is the first 40 images that appear in a directory listing as produced by the dir function of MATLAB). Plot in your report the recognition rates for $k \in \{1, 2, 3, 5, 10, 15, 20, 30, 50, 60, 65, 75, 100, 200, 300, 500, 1000\}$ based on (a) the squared difference between all the eigencoefficients and (b) the squared difference between all except the three eigencoefficients corresponding to the eigenvectors with the three largest eigenvalues. [40 points]

2. Display in your report the reconstruction of any one face image from the Yale database using $k \in \{2, 10, 20, 50, 75, 100, 125, 150, 175\}$ values. Plot the 25 eigenvectors (eigenfaces) corresponding to the 25 largest eigenvalues using the subplot or subimage commands in MATLAB. [10 points]

3. What will happen if you test your system on images of people which were not part of the training set? (i.e. the last 8 people from the ORL database). What mechanism will you use to report the fact that there is no matching identity? Work this out carefully and explain briefly in your report. Write code to test whatever you

propose on all the 32 remaining images (i.e. 8 people times 4 images per person), as also the entire test set containing 6 images each of the first 32 people. How many false positives/negatives did you get? [10 points]

4. Given a matrix $\boldsymbol{A}$ of size $m \times n$, write a MATLAB routine called MySVD which takes this matrix as input and outputs the left and right singular vectors (i.e. column vectors of $\boldsymbol{U}$ and $\boldsymbol{V}$ under usual notation) and the singular values (i.e. diagonal entries of $\boldsymbol{S}$) of $\boldsymbol{A}$. You are not allowed to use the svd or svds functions of MATLAB directly. You should use only the eigenvalue decomposition routines eig or eigs for this task. Cross-check your answer by verifying that $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$ based on your computation. [10 points]

5. Consider a set of N vectors $\mathcal{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_N}\}$ each in $\mathbb{R}^d$, with average vector $\bar{\boldsymbol{x}}$. We have seen in class that the direction $\boldsymbol{e}$ such that $\sum_{i=1}^{N} \|\boldsymbol{x_i} - \bar{\boldsymbol{x}} - (\boldsymbol{e} \cdot (\boldsymbol{x_i} - \bar{\boldsymbol{x}}))\boldsymbol{e}\|^2$ is minimized, is obtained by maximizing $\boldsymbol{e}^t\boldsymbol{C}\boldsymbol{e}$, where $\boldsymbol{C}$ is the covariance matrix of the vectors in $\mathcal{X}$. This vector $\boldsymbol{e}$ is the eigenvector of matrix $\boldsymbol{C}$ with the highest eigenvalue. Prove that the direction $\boldsymbol{f}$ perpendicular to $\boldsymbol{e}$ for which $\boldsymbol{f}^t\boldsymbol{C}\boldsymbol{f}$ is maximized, is the eigenvector of $\boldsymbol{C}$ with the second highest eigenvalue. For simplicity, assume that all non-zero eigenvalues of $\boldsymbol{C}$ are distinct and that $\text{rank}(\boldsymbol{C}) > 2$. [10 points]

6. Consider a matrix $\boldsymbol{A}$ of size $m \times n$. Define $\boldsymbol{P} = \boldsymbol{A}^T\boldsymbol{A}$ and $\boldsymbol{Q} = \boldsymbol{A}\boldsymbol{A}^T$. (Note: all matrices, vectors and scalars involved in this question are real-valued).

   (a) Prove that for any vector $\boldsymbol{y}$ with appropriate number of elements, we have $\boldsymbol{y}^t\boldsymbol{P}\boldsymbol{y} \geq 0$. Similarly show that $\boldsymbol{z}^t\boldsymbol{Q}\boldsymbol{z} \geq 0$ for a vector $\boldsymbol{z}$ with appropriate number of elements. Why are the eigenvalues of $\boldsymbol{P}$ and $\boldsymbol{Q}$ non-negative?

   (b) If $\boldsymbol{u}$ is an eigenvector of $\boldsymbol{P}$ with eigenvalue λ, show that $\boldsymbol{A}\boldsymbol{u}$ is an eigenvector of $\boldsymbol{Q}$ with eigenvalue λ. If $\boldsymbol{v}$ is an eigenvector of $\boldsymbol{Q}$ with eigenvalue μ, show that $\boldsymbol{A}^T\boldsymbol{v}$ is an eigenvector of $\boldsymbol{P}$ with eigenvalue μ. What will be the number of elements in $\boldsymbol{u}$ and $\boldsymbol{v}$?

   (c) If $\boldsymbol{v}_i$ is an eigenvector of $\boldsymbol{Q}$ and we define $\boldsymbol{u}_i \triangleq \dfrac{\boldsymbol{A}^T\boldsymbol{v}_i}{\|\boldsymbol{A}^T\boldsymbol{v}_i\|_2}$. Then prove that there will exist some real, non-negative γ_i such that $\boldsymbol{A}\boldsymbol{u}_i = \gamma_i\boldsymbol{v}_i$.

   (d) It can be shown that $\boldsymbol{u}_i^T\boldsymbol{u}_j = 0$ for $i \neq j$ and likewise $\boldsymbol{v}_i^T\boldsymbol{v}_j = 0$ for $i \neq j$ for correspondingly distinct eigenvalues.[1]. Now, define $\boldsymbol{U} = [\boldsymbol{v}_1|\boldsymbol{v}_2|\boldsymbol{v}_3|...|\boldsymbol{v}_m]$ and $\boldsymbol{V} = [\boldsymbol{u}_1|\boldsymbol{u}_2|\boldsymbol{u}_3|...|\boldsymbol{u}_n]$. Now show that $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Gamma}\boldsymbol{V}^T$ where $\boldsymbol{\Gamma}$ is a diagonal matrix containing the non-negative values $\gamma_1, \gamma_2, ..., \gamma_n$. With this, you have just established the existence of the singular value decomposition of any matrix $\boldsymbol{A}$. This is a key result in linear algebra and it is widely used in image processing, computer vision, computer graphics, statistics, machine learning, numerical analysis, natural language processing and data mining. [5 + 5 + 5 + 5 = 20 points]

[1] This follows because $\boldsymbol{P}$ and $\boldsymbol{Q}$ are symmetric matrices. Consider $\boldsymbol{P}\boldsymbol{u}_1 = \lambda_1\boldsymbol{u}_1$ and $\boldsymbol{P}\boldsymbol{u}_2 = \lambda_2\boldsymbol{u}_2$. Then $\boldsymbol{u}_2^T\boldsymbol{P}\boldsymbol{u}_1 = \lambda_1\boldsymbol{u}_2^T\boldsymbol{u}_1$. But $\boldsymbol{u}_2^T\boldsymbol{P}\boldsymbol{u}_1$ also equal to $(\boldsymbol{P}^T\boldsymbol{u}_2)^T\boldsymbol{u}_1 = (\boldsymbol{P}\boldsymbol{u}_2)^T\boldsymbol{u}_1 = (\lambda_2\boldsymbol{u}_2)^T\boldsymbol{u}_1 = \lambda_2\boldsymbol{u}_2^T\boldsymbol{u}_1$. Hence $\lambda_2\boldsymbol{u}_2^T\boldsymbol{u}_1 = \lambda_1\boldsymbol{u}_2^T\boldsymbol{u}_1$. Since $\lambda_2 \neq \lambda_1$, we must have $\boldsymbol{u}_2^T\boldsymbol{u}_1 = 0$.