

Assignment 4: CS 663

Due: 26th 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 the submission guidelines file in the homework folder. However, please do not submit the face image databases in your zip file that you will upload on moodle. Please see `assignment4.SVD.FaceRecognition.rar`. Upload the file on moodle before 11:55 pm on 26th October. We will not penalize submission of the files till 10 am on 27th October. **No late assignments will be accepted after this time.** Please preserve a copy of all your work until the end of the semester.

1. A student is trying to write his/her own routine to compute the SVD of a matrix \mathbf{A} of size $m \times n$. For this, the student computes $\mathbf{A}^T \mathbf{A}$ and assigns the eigenvectors of $\mathbf{A}^T \mathbf{A}$ (computed using the `eig` routine in MATLAB) to be the matrix \mathbf{V} consisting of the right singular vectors of \mathbf{A} . Then the student also computes $\mathbf{A} \mathbf{A}^T$ and assigns the eigenvectors of $\mathbf{A} \mathbf{A}^T$ (computed using the `eig` routine in MATLAB) to be the matrix \mathbf{U} consisting of the left singular vectors of \mathbf{A} . Finally, the student assigns the non-negative square-roots of the eigenvalues (computed using the `eig` routine in MATLAB) of either $\mathbf{A}^T \mathbf{A}$ or $\mathbf{A} \mathbf{A}^T$ to be the diagonal matrix \mathbf{S} consisting of the singular values of \mathbf{A} . He/she tries to check his code and is surprised to find that $\mathbf{U} \mathbf{S} \mathbf{V}^T$ is not equal to \mathbf{A} . Why could this be happening? What should (s)he do to rectify this error, if any? [15 points]
2. Consider a set of N vectors $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ each in \mathbb{R}^d , with average vector $\bar{\mathbf{x}}$. We have seen in class that the direction \mathbf{e} such that $\sum_{i=1}^N \|\mathbf{x}_i - \bar{\mathbf{x}} - (\mathbf{e} \cdot (\mathbf{x}_i - \bar{\mathbf{x}}))\mathbf{e}\|^2$ is minimized, is obtained by maximizing $\mathbf{e}^T \mathbf{C} \mathbf{e}$, where \mathbf{C} is the covariance matrix of the vectors in \mathcal{X} . This vector \mathbf{e} is the eigenvector of matrix \mathbf{C} with the highest eigenvalue. Prove that the direction \mathbf{f} perpendicular to \mathbf{e} for which $\mathbf{f}^T \mathbf{C} \mathbf{f}$ is maximized, is the eigenvector of \mathbf{C} with the second highest eigenvalue. For simplicity, assume that all non-zero eigenvalues of \mathbf{C} are distinct and that $\text{rank}(\mathbf{C}) > 2$. [15 points]
3. Suppose you are standing in a well-illuminated room with a large window, and you take a picture of the scene outside. The window undesirably acts as a semi-reflecting surface, and hence the picture will contain a reflection of the scene inside the room, besides the scene outside. While solutions exist for separating the two components from a single picture, here you will look at a simpler-to-solve version of this problem where you would take two pictures. The first picture g_1 is taken by adjusting your camera lens so that the scene outside (f_1) is in focus (we will assume that the scene outside has negligible depth variation when compared to the distance from the camera, and so it makes sense to say that the entire scene outside is in focus), and the reflection off the window surface (f_2) will now be defocussed or blurred. This can be written as $g_1 = f_1 + h_2 * f_2$ where h_2 stands for the blur kernel that acted on f_2 . The second picture g_2 is taken by focusing the camera onto the surface of the window, with the scene outside being defocussed. This can be written as $g_2 = h_1 * f_1 + f_2$ where h_1 is the blur kernel acting on f_1 . Given g_1 and g_2 , and assuming h_1 and h_2 are known, your task is to derive a formula to determine f_1 and f_2 . Note that we are making the simplifying assumption that there was no relative motion between the camera and the scene outside while the two pictures were being acquired, and that there were no changes whatsoever to the scene outside or inside. Even with all these assumptions, you will notice something inherently problematic about the formula you will derive. What is it? [8+7 = 15 points]
4. In this part, you will implement a mini face recognition system. Download the ORL face database from the homework folder. 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 implement the recognition system by using the `svd` function of MATLAB on an appropriate data matrix. 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 `eig` function of MATLAB (on the \mathbf{L} matrix as defined in class) instead of `svd`.

Repeat the same experiment (using just the `svd` routine) on the Yale Face database from the homework folder. This database contains about 64 images each of 38 individuals (*labeled from 1 to 39, with number 14 missing; some folders have slightly less than 64 images*). 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 24 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. [30 points]

5. Display in your report the reconstruction of any one face image from the ORL 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]
6. 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? [15 points]