Engineering Mathematics Project

Report: half of your score will be graded proportional to the quality of your report. You should provide a distinct section for each problem, include the desired outputs and explain what you've done. Don't forget to discuss your results as well. It is not necessary to accommodate your source codes in your reports unless you want to refer to them. Compactness, expressiveness and neatness are of high importance.
As you have to upload your submission electronically, it is of high interest to prepare your reports using Microsoft Office tools or Latex. However, scanned handwritten solutions are also acceptable as long as they are readable, neat and expressive.

Consider a two-class problem and generate 1000 samples of Gaussian distribution for each class, using the following mean vectors and covariance matrices:

$$\mu_1 = [10, 10]^T \quad , \mu_2 = [22, 10]^T \quad , \Sigma_1 = \Sigma_2 = \begin{bmatrix} 4 & 4 \\ 4 & 9 \end{bmatrix}$$

a. Compute and draw the line on which PCA projects the data points.

b. Projects all data points onto the resulting PCA line and visualize the results.

c. Do you see what you already expected? Explain your observation.

d. Reconstruct the data points to the two-dimensional space and compute the reconstruction error.

In this part you will implement the Eigenface method for recognizing human faces. You will use face images from The Yale Face Database B, where there are 64 images under different lighting conditions per each of 10 distinct subjects, 640 face images in total. With your implementation, you will explore the power of the Singular Value Decomposition (SVD) in representing face images.

a. After you unzip faces.zip, you will find a folder called images which contains all the training and test images; train.txt and test.txt specifies the training set and test (validation) set split respectively, each line gives an image path and the corresponding label.

b. Load the training set into a matrix X: there are 540 training images in total, each has 50x50 pixels that need to be concatenated into a 2500-dimensional vector. So the size of X should be 540×2500, where each row is a flattened face image. Pick a face image from X and display that image in grayscale. Do the same thing for the test set. The size of matrix X_test for the test set should be 100×2500.

c. Average Face: Compute the average face μ from the whole training set. Display the average face as a grayscale image.

d. Mean Subtraction: Subtract average face μ from every column in X. Pick a face image after mean subtraction from the new X and display that image in grayscale. Do the same thing for the test set X_test using the precomputed average face μ in (c.)

e. Eigenface: Perform Singular Value Decomposition (SVD) on training set X (X=UΣV^T) to get matrix V^T, where each row of V^T has the same dimension as the face image. We refer to v_i, the i-th row of V^T, as i-th eigenface. Display the first 10 eigenfaces as 10 images in grayscale.

f. Low-rank Approximation: Since Σ is a diagonal matrix with non-negative real numbers on the diagonal in non-ascending order, we can use the first r elements in Σ together with first r columns in U and first r rows in V^T to approximate X̂_r. The matrix X̂_r is called rankr approximation of X. Plot the rank-r approximation error ‖X - X̂_r‖_F as a function of r when r=1,2,…,200.

g. Eigenface Feature: The top r eigenfaces span an r –dimensional linear subspace of the original image space called face space, whose origin is the average face μ, and whose axes are the eigenfaces {v_1,v_2,…,v_r}. Therefore, using the top r eigenfaces {v_1,v_2,…,v_r}. we can represent a 2500-dimensional face image z as an r-dimensional feature vector F=V^T. Write a function to generate r-dimensional feature matrix F and F_test for training images X and test images X_test, respectively (to get F, multiply X to the transpose of first r rows of V^T , F should have same number of rows as X and r columns; similarly for X_test.)

h. Face Recognition: Extract training and test features for r = 10. Train a Logistic Regression model using F and test on F_test. Report the classification accuracy on the test set. Plot the classification accuracy on the test set as a function of r when r=1, 2, …, 200.