Assignment #5

PCA & LDA

How TA evaluates your assignments:

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.

Source Code: create an m-file for any problem and write all your codes there. If a problem consists of several sub-problems, separate them by comments in your code. Finally, name your m-files according to the number of the problems.

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.

What to hand in:

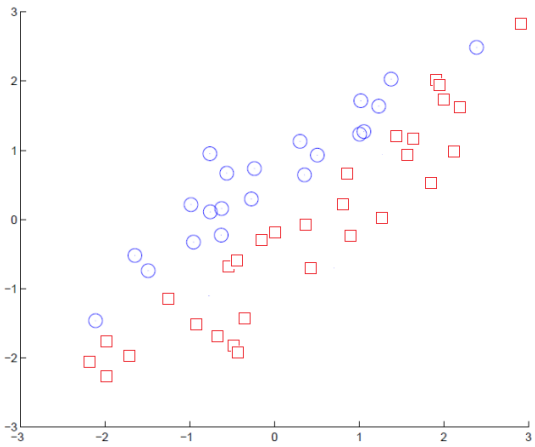
You must submit your report (.pdf) and source codes (m-files) for each assignment. Zip all your files into an archive file and use the following template to name it:

HW5_XXXXX.zip

where XXXXX must be replaced with your student ID. Your file size must not be bigger than 20MB. Send your files to *mohammadhme@gmail.com* with a subject of PR961_HW5_ XXXXX (replace XXXXX with your student number).

The Due Date for This Assignment is: Dey. 7th

1.In the following Figure, draw the first principal component direction and the first Fisher's linear discriminant direction. (For linear discriminant, consider round points as the positive class, and square points as the negative class)



2. Let's carry out a Principal Component Analysis by hand for a simple data set

$$X = \begin{bmatrix} 3 & 2 & 4 & 0 & 6 & 3 & 1 & 5 & -1 & 7 \\ 1 & 3 & -1 & 7 & -5 & 1 & 0 & 2 & -1 & 3 \end{bmatrix}$$

where each row corresponds to a dimension, and each column corresponds to a sample (observation). Let's say the first five columns belong to the positive class, and the second five columns belong to the negative class.

a. Plot the data points (You can plot the points by Matlab).

b. Create a new matrix Y by subtracting off the mean expression value (Anew, you may use Matlab to plot the new positions of data points).

c. Compute the 2 by 2 covariance matrix C using the data in Y. Compute the eigenvalues of C.

d. What fraction of the total variance of the data is accounted for by the first principal component of C?

e. Find the principal component eigenvectors and plot their directions on the same plot as the data points.

f. Re-express the matrix Y as a one-dimension dataset by projecting each data point (column) onto the PCs. (use Matlab for simplicity, but do not include code).

3. Given 3 data points in 2-d space, (1, 1), (2, 2) and (3, 3),

a. What is the first principle component?

b. If we want to project the original data points into 1-d space by principle component you choose, what is the variance of the projected data?

c. For the projected data in (b), now if we represent them in the original 2-d space, what is the reconstruction error? What is the variance of the projected data?

4. You must wrestle with LDA in this exercise. Consider the following 2-D dataset:

$$X_1 = \{(4,1),(2,4),(2,3),(3,6),(4,4)\}$$
$$X_2 = \{(9,10),(6,8),(9,5),(8,7),(10,8)\}$$

a. Plot the data points (You can use Matlab to plot points, and not for computations)

b. Compute and plot the LDA projection line.

c. Project all data points into the resulting subspace. Plot the projected data points.

d. Discuss the separability of classes in the projected subspace.

5. Let's see the difference between finding the principal components of uncentered vs. centered data. You should use Matlab to solve this problem, include your code. Suppose you have the following data matrix X:

$$X = \begin{bmatrix} 12.1 & 6.6 & 11.5 & 8.7 & 15.4 & 6.9 & 8.8 & 10.1 & 19.1 & 15.6 & 22.2 & 10.2 & 15.7 & 10.0 & 7.2 & 11.8 & 11.3 & 12.5 & 11.4 & 12.9 \\ 6.9 & 7.5 & 6.1 & 8.1 & 9.0 & 6.0 & 8.5 & 8.3 & 11.2 & 7.6 & 9.3 & 7.5 & 9.2 & 6.4 & 6.3 & 7.7 & 8.3 & 7.5 & 9.5 & 9.3 \end{bmatrix}$$

Where each row of X is a dimension and each column of X is a sample (i.e. data point); that is, the matrix

X has 20 points of 2-dimensional data.

a. Without centering, i.e. mean subtracting the data, find the covariance matrix of X.

b. Find the eigenvalues and eigenvectors of the covariance matrix. Reorder your eigenvectors and eigenvalues so that the eigenvector with the highest eigenvalue is in the first column

c. Transform the data X into the principal component space and plot each point, include a printout of your graph.

d. Now redo the previous three steps, but subtract the mean of each dimension from all of the data points. Include your code and the figure.

e. Describe the effect of mean subtracting in brief.

6. Let's see the behavior of PCA and LDA more intuitively. Consider a two-class problem and generate 1000 samples for each class, using the following mean and covariance matrix:

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

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.

e. Compute and draw the line on which LDA projects the data points.

f. Projects all data points onto the resulting LDA line and visualize the results.

g. Explain your observations.

7. We are given a set of labelled images of size 50 by 50 representing the faces of 10 people under different illumination conditions. Each image is thus represented by a 1D vector of 2500 dimensions. The goal is to recognize the faces by using the K Nearest Neighbor (KNN) algorithm. To avoid computing Euclidian distances in the 2500 dimensional spaces, the face images will be first projected into a smaller dimension space using PCA.

a. Using the 'Subset1YaleFaces.mat' data, compute the eigenvectors (eigenfaces) representation. Display the set of eigenvalues. How many non-zero eigenvalues do you obtain? Why is it so? Display the 'mean' face as well as the first 9 eigenfaces. Then, for different values of M, take a face, project it on the first M eigenvectors and visualize its reconstruction.

b. We are now going to use a classifier to recognize images of people in the database. As classifier, we will use a KNN classifier. In Matlab, you can use the function knnclassify. The training data will be the faces from Subset1YaleFaces.mat (cf above) (i.e. this set will be used to build the PCA projection matrix, and also as training data by using the corresponding identity labels). Using the validation dataset X_{valid} 'Subset2YaleFaces.mat', program and do the following:

b-1) project all faces from this dataset in order to obtain their PCA representation.

b-2) for different values of M and k (you can loop over these values), classify all faces using KNN (and the Subset1 dataset) from X_{valid}, and compute the recognition error.

b-3) select the values of M and k that give the best performance on this validation set. Then classify the faces from the Subset3YaleFaces.mat dataset, and compute the recognition error. You can check for the errors whether they make sense or not visually.