Homework #5
PCA & LDA
Statistical Pattern Recognition

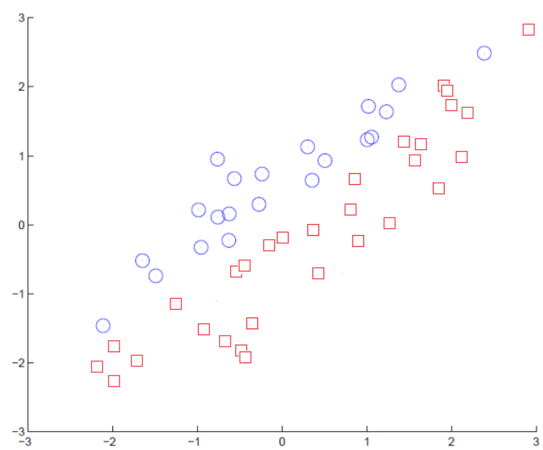
Due Dey 20th,1395
Format of homework file: Archive all files in a folder named as your student number and send it to mohammadhme@gmail.com. Send your emails with a subject of PR95F5_xxxxxx (replace xxxxxx by your student number)

---

1. In performing PCA, what happens if we set the number of dimensions in the new space equal to the number of dimensions in the original space? Does the analysis still hold?

---

2. 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)



---

3. Consider 3 data points in the 2-d space: $(-1,1), (0,0), (1,1)$
   a. What is the first principal component (write down the actual vector)?
   b. If we project the original data points into the 1-d subspace by the principal component you choose, what are their coordinates in the 1-d subspace? And what is the variance of the projected data?
   c. For the projected data you just obtained above, now if we represent them in the original 2-d space and consider them as the reconstruction of the original data points, what is the reconstruction error?

---

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

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

   a. Plot the data points.
   b. Create a new matrix Y by subtracting off the mean expression value.
   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).

---

5. [**computer project]** Consider a two-class problem and generate 1000 samples for each class, using the following mean and covariance matrix:

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

   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.

---

6. [**computer project]** In this assignment 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. Download The Face Dataset from:
   http://cs5785-cornell-tech-16fall.github.io/data/faces.zip
   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 $\mathbf{X}_{\text{test}}$ 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 $\mathbf{X}_{\text{test}}$ using the precomputed average face $\boldsymbol{\mu}$ in (c).

e. Eigenface: Perform Singular Value Decomposition (SVD) on training set **X** ($\boldsymbol{X} = U\Sigma V^T$**)** 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 $\widehat{\mathbf{X}}_{\mathbf{r}}$. The matrix $\widehat{\mathbf{X}}_{\mathbf{r}}$ is called rank-*r* approximation of **X**. Plot the rank-*r* approximation error $\left\|\boldsymbol{X} - \widehat{\mathbf{X}}_{\mathbf{r}}\right\|_F$ as a function of *r* when $r = 1,2, \dots ,200$.

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 $\{v_1, v_2, \dots , v_r\}$. Therefore, using the top *r* eigenfaces $\{v_1, v_2, \dots , v_r\}$. we can represent a 2500-dimensional face image z as an *r*-dimensional feature vector $\boldsymbol{F} = V^T z$. Write a function to generate *r*-dimensional feature matrix **F** and $\mathbf{F}_{\text{test}}$for training images **X** and test images $\mathbf{X}_{\text{test}}$, 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 $\mathbf{X}_{\text{test}}$).

h. Face Recognition: Extract training and test features for *r* = 10. Train a Logistic Regression model using **F** and test on $\mathbf{F}_{\text{test}}$**.** Report the classification accuracy on the test set. Plot the classification accuracy on the test set as a function of *r* when $r = 1,2, \dots ,200$.