CSECE4720/7720 Spring 2015
Introduction to Machine Learning and Pattern Recognition

Homework Assignment 4

due Thursday 4/16/2015

Please refer to the file ClassificationToolboxS2015.pdf regarding PCA and MDA.
Note: it is best that you implement your own PCA code by using the MATLAB function eig().

Problem 1

In the class of 3/17/2015, we have derived the equation

$$-\frac{2S_{Wx}\underline{w}(\underline{w}^{'}S_{Bx}\underline{w})}{(\underline{w}^{'}S_{Wx}\underline{w})^{2}}+\frac{2S_{Bx}\underline{w}}{\underline{w}^{'}S_{Wx}\underline{w}}=\underline{0} \qquad \text{Eq.(1)}$$

by maximizing the objective function of $J(\underline{w})=\frac{\underline{w}^{'}S_{Bx}\underline{w}}{\underline{w}^{'}S_{Wx}\underline{w}}$.

Complete the following two steps to verify the two-class LDA estimation equation:

$$\underline{w}=S_{Wx}^{-1}(\underline{m}_{x,1}-\underline{m}_{x,2}) \qquad \text{Eq.(2).}$$

(a) Show that Eq.(1) can be reduced to the following form:

$$-S_{Wx}\underline{w}\frac{\underline{w}^{'}(\underline{m}_{x,1}-\underline{m}_{x,2})}{\underline{w}^{'}S_{Wx}\underline{w}}+(\underline{m}_{x,1}-\underline{m}_{x,2})=\underline{0}$$

(b) Verify that the expression $\frac{\underline{w}^{'}(\underline{m}_{x,1}-\underline{m}_{x,2})}{\underline{w}^{'}S_{Wx}\underline{w}}=1$ when the LDA weight vector $\underline{w}$ defined by Eq.(2) is substituted into this expression.

Problem 2

(a) Carry out Principal component analysis (PCA) on the Iris training dataset (as defined in HW3). Note: if you are using the PCA function from the classification toolbox, for the Iris data, the train_patterns should be a 4xn matrix, where n is the number of training samples, and the input target vector is simply a null vector [].

(b) Use the PCA result of (a) to reduce the training and test data feature dimension from 4 to 2 (note: there should be only one transformation matrix W derived from the training data, and the test data should not be used in deriving W).

(c) Estimate the 2-D mean vectors and full covariance matrices for each class by using the program you wrote for HW3.

(d) Perform maximum likelihood classification on the 2-D test data by using the models you've estimated in (c).

(e) Summarize your results of (d) in a confusion table.

(f) Make a scatter plot for the PCA projected Iris data, where the data samples of different classes should be marked by different symbols or colors.

(g) Give interpretations of your classification results with reference to the scatter plot.

Problem 3

(a) Carry out multiple linear discriminant analysis (MDA) on the Iris training dataset (as defined in HW3) Note: for the Iris data, the train_patterns should be a 4xn matrix, and the input target vector should be a 1xn vector with each component equal to the class label of the corresponding data sample, where n is the number of training samples. For the three classes, you can use 1,2,3 to represent the class labels.

(b) Use the MDA derived transformaiton matrix to reduce the training and test data feature dimension from 4 to 2 (as in Problem 2, there should be only one transformation matrix W derived from the training data, and the test data should not be used in deriving W).

(c) Explain why we cannot keep 3 feature components by using MDA on this task.

(d) Estimate the 2-D mean vectors and covariance matrices for each class.

(e) Perform maximum likelihood classification on the 2-D test data by using the models you've estimated in (d).

(f) Summarize your results of (e) in a confusion table.

(g) Make a scatter-plot for the MDA projected Iris data, where the data samples of different classes should be marked by different symbols or colors.

(h) Give interpretations of your classification result with reference to the scatter plot.

Problem 4

You are to use PCA for image approximation (discussed in the class of 3/10/15). You need to read the 10 images from the input1 folder in the ImageCodeExample file and convert them into a vector format before PCA analysis. The ImageCodeExample file has example codes for data format conversion and image read, display, and write.

(a) Implement the image approximation algorithm (the equation for $\underline{\tilde{x}}$) in MATLAB. (Note, the approximated image vector $\underline{\tilde{x}}$ has the same dimension size as the original image vectors $\underline{x}$).

(b) Approximate the 10 images by using 1, 4, and 8 eigenvectors, respectively. Display the original and the approximated images.

(c) Compute the proportion of variances (β_k, discussed in the class of 3/5/15) and the approximation errors (e^2, discussed in the class of 3/17/15) for the three cases of (b) where 1, 4, and 8 eigenvectors are used in image approximation, respectively.

Problem 5

The K-nearest-neighbor classifier (Knn) is to be trained and evaluated on a face image dataset for face identification. The Knn classifier is implemented in Nearest_Neighbor.m in the Classification Toolbox: Please use the help function in MATLAB to find details of the function.

The face image dataset has five people and therefore five classes. You need to read the image files from the input1~input5 folders in the ImageFaceIDExample.rar file and convert each of them into a vector format as you did in Problem 4. The training and the test data sets are defined in the following way:

Training set: the first 5 images of each person;
Test set: the last 5 images of each person.

(a) From the 25 images in the training set, derive the PCA transformation matrices corresponding to keeping 1, 4, and 10 eigenvectors, respectively, and compute the proportion of variances for each of the 3 cases.
(b) Perform dimension reductions on both the training and the test data by using each of the 3 transformation matrices you've derived in (a).
(c) For each of the 3 cases of feature dimensions, use the Knn classifier to classify the test data, where three parameters for the number of neighbors are to be evaluated:

k = 1, 3, and 5. Summarize the test set classification error rate in the following table:

# of features / # of nearest neighbors	1	4	10
1			
3			
5			

Problem 6

The Gaussian-kernel based Parzen window method is to be used to estimate the probability density functions from the two datasets 'SalmonLightness.dat' and 'SeabassLightness.dat' that you used in HW1.

(a) Set the kernel function's standard deviation parameter to h = 0.8. Estimate the three pdfs $p(lightness \mid salmon)$, $p(lightness \mid seabass)$, and $p(lightness)$ for the lightness

value range of 0 through 12 with the increment of 0.1; plot the three pdfs in one figure, and distinguish them by different colors or line styles.

(b) Repeat (a) but use h = 0.2.

(c) Compare the pdf figures in (a) and (b) and state your observations.