CS 434b-654b
Assignment 3 (*Due March 20)*

Instructions: Hand in all the work you do. This should include matlab code and written answers. Do not hand in any plots that you do, just discuss them. Also, the matlab code should be emailed to me as scripts or function files. Make sure your email includes your name in the subject, and explanation which scripts go with which problem in the body of the email.

Problem 1 (40%): T

For this problem, use the face data from the previous homework

(a) Write a Matlab function for performing PCA. The function should be named pca(A,m). It should take a data matrix A of dimension N by d with examples piled as rows, and m is the dimension of the reduced space. The output should consist of 3 matrices, Y, V, and DataMean. Y should be a matrix which holds the smaller dimensional examples, that is Y should have dimension N by m. V should be a d by m matrix which holds the m eigenvectors used for dimensionality reduction. DataMean should be a d by 1 vector holding the mean of the input samples. You can use matlab function eig or eig for this part, but you cannot use the built-in matlab function for PCA.

(b) Perform PCA dimensionality reduction on the examples in the array faceTrain using m=15. Visualize the first 15 eigenvectors (or so called “eignefaces”) using the provided function visualize_pc(V), where input matrix V should have the first 15 eigenvectors arranged as columns of V. Give a once sentence description of what the eigenvectors look like.

(c) Perform PCA dimensionality reduction on the examples in the array faceTrain using m=3, 10, 20. After performing PCA, for each value of m, see how well the 30th face in the array faceTrain is approximated. To do this, compute the Euclidean distance between the 30th face and the 30th face approximated with the first m eigenvectors. The Euclidian distance should go down with larger m.

(d) Now compute how well the 10th face from the array faceTest is approximated with m = 3, 10, 20 using the m largest eigenvectors computed on the array faceTrain.

(e) Repeat part (d) using the 16 image from the array nonFaceTrain.

(f) Discuss the difference between (c),(d), and (e).

Problem 2 (20%): T

For this problem, use the face data from the previous homework

(a) Use the code from problem 1(a) to project data in array faceTrain and nonfaceTrain to15 dimensions, *separately* for each of these 2 classes. Assume that the projected data has multivariate Gaussian distribution. Classify samples in faceTest and nonfaceTest (you have to project the data to the lower dimensional subspace before classifying). Compute the confusion matrix (see the definition of confusion matrix from the previous homework).

(b) Repeat part (a) but project data to 1 dimension. Discuss the difference in results between (a) and (b).

Problem 3 (40%):

For this problem, load the data in P3.mat. This data consists of 3 classes from real plant data. Each sample has 4 features corresponding to measurements on plants. Each class corresponds to a different type of a plant.

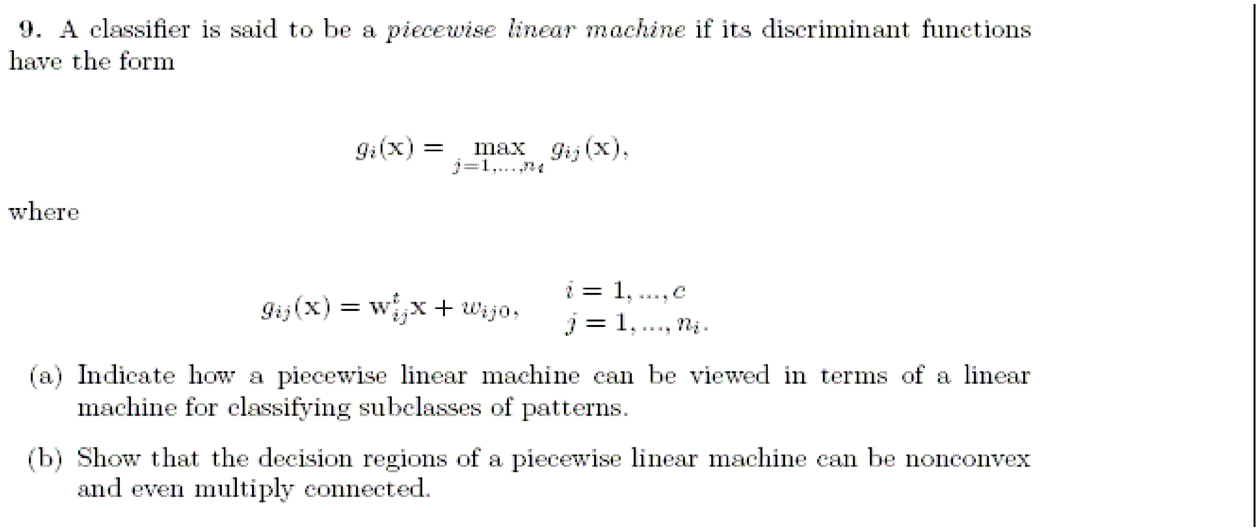
(a) Write a matlab function [Y,V] = lda(class1,class2,class3,dim) which takes as an input 3 matrices, each holding samples from a single class piled as rows. The input dim should be an integer equal to either 1 or 2, to project the samples either to 1 or 2 dimensions (remember that for 3 classes, we can only project the data to 1 or 2 dimensions). The function should perform LDA and output the reduced samples in Y and the projection matrix in V. You can use matlab function eig or eigs.

(b) Use the function you wrote in part (a) to project the data to 1 and 2 dimensions. Visualize your results separately for each dimension using function scatter and different color for each class. Are the samples well separated in one dimension? In two dimensions?

(c) Assume in the low dimensional space features have gaussian distribution. Use leave-one-out cross-validation to compute the confusion matrix when using LDA to project to dimension 1 and dimension 2

(d) Use MSE for multiple classes to classify the samples, using again leave-one-outcross validation. Compute the confusion matrix. Discuss the difference (if any) between (c) and (d).

Problem 4 (20%): Problem 10(b) on page 272,



9. A classifier is said to be a *piecewise linear machine* if its discriminant functions have the form

$$g_i(\mathbf{x}) = \max_{j=1,...,n_i} g_{ij}(\mathbf{x}),$$

where

$$g_{ij}(\mathbf{x}) = \mathbf{w}_{ij}^t\mathbf{x} + w_{ij0}, \qquad \begin{array}{l} i = 1, ..., c \\ j = 1, ..., n_i. \end{array}$$

(a) Indicate how a piecewise linear machine can be viewed in terms of a linear machine for classifying subclasses of patterns.

(b) Show that the decision regions of a piecewise linear machine can be nonconvex and even multiply connected.