

**Fall 2007, CPSC 689-607**  
**Special Topics in Pattern Classification and Clustering**  
**Homework #2**  
**Due date: 10/7**

*In recognition of the Texas A&M University policies of academic integrity, I certify that I have neither given nor received dishonest aid in this homework assignment.*

Name: \_\_\_\_\_

Signature: \_\_\_\_\_

**NOTES:**

1. Start each problem on a separate page
2. Provide hardcopies of all plots, MATLAB code and derivations
3. Download the compressed file '[hw2.zip](#)' from the course web page
4. Sign and return this page with your finished assignment.

**Problem 1 (25%)**

(1) Use a Gaussian kernel to generate a kernel density estimate (KDE) of the univariate dataset '[hw2p1\\_data.mat](#)'. Compute the estimate at a set of points evenly spaced over the range of the data. For comparison purposes, plot the KDE on the same graph as the normalized histogram of the data. Experiment with the value of the bandwidth  $h$  to obtain an appropriate smoothing effect. Show results for a "small", a "large" and your "best" bandwidth.

*NOTE: You are expected to implement your own KDE code.*

*HINT: To save trees, use the command 'subplot' to display the three plots ("small", "large" and "best") on the same window.*

(2) Next we will use the leave-one-out method outlined in Lecture 7 (page 16) to estimate the optimum bandwidth.

- (a) Split the dataset into two subsets: a training set containing all but the  $n^{\text{th}}$  example, and a validation set containing only the  $n^{\text{th}}$  example.
- (b) Build a KDE using only the training set, and estimate the log-likelihood of the validation sample (i.e., the logarithm of the KDE at the validation sample)
- (c) Repeat steps (a-b) for every example in the dataset, and compute the average log likelihood. Store this value.
- (d) Repeat steps (a-c) for 100 different bandwidths, logarithmically spaced between  $h_0/100$  and  $100h_0$  as follows:

$$h = \left\{ \frac{h_0}{100}, \alpha^1 \frac{h_0}{100}, \alpha^2 \frac{h_0}{100}, \alpha^3 \frac{h_0}{100}, \dots, \alpha^{98} \frac{h_0}{100}, 100h_0 \right\} \text{ with } h_0 = 0.9AN^{-1/5}; A = \min\left(\sigma_x, \frac{\text{iqr}(x)}{1.34}\right)$$

- (e) Generate a plot of the average log-likelihood in (c) versus the bandwidth.
- (f) Select the bandwidth which yields the highest log-likelihood, and
  - a. Generate a KDE with this bandwidth, and with the plug-in estimate  $h_0$ ,
  - b. Plot the two KDE on the same graph as the normalized histogram, and
- (g) Discuss your results.

## Problem 2 (10%)

Load the binary file 'hw2p2\_data.mat' into MATLAB. This will create a matrix "x", which consists of high-dimensional feature vectors arranged by rows:

- Generate a scatter plot of the first three dimensions. Rotate the reference frame (command 'rotate3d') and try to find structure in the data by changing the viewpoint. Plot the best orientation that you can find. Can you identify any structure?
- Compute the Principal Components of the data and generate a plot of the eigenvalues, sorted in decreasing order. How many eigenvalues are responsible for most of the variance in the data?
- Generate a scatter plot of the first three PCA projections (the ones with largest eigenvalues). Rotate the reference frame and try to find structure in the data by changing the viewpoint. Plot the best such orientation. Can you identify any structure? Where else in the data may the structure be?
- Discuss your findings.

NOTE: You are expected to implement your own PCA code.

## Problem 3 (10%)

Generate 150 examples for each of the four densities:

$$\begin{aligned} \mu_1 &= [0 \ 3 \ 3]^T & \mu_2 &= [4 \ 0 \ 3]^T & \mu_3 &= [0 \ -4 \ 0]^T & \mu_4 &= [-5 \ 2 \ 0]^T \\ \Sigma_3 &= \begin{bmatrix} 3 & 2 & -1.9 \\ & 3 & 0 \\ & & 3 \end{bmatrix} & \Sigma_2 &= \begin{bmatrix} 6 & -2.5 & -1.9 \\ & 6 & -0.5 \\ & & 4 \end{bmatrix} & \Sigma_1 &= \begin{bmatrix} 5 & -1.5 & -0.9 \\ & 1 & -0.1 \\ & & 1 \end{bmatrix} & \Sigma_4 &= \begin{bmatrix} 9 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 9 \end{bmatrix} \end{aligned}$$

To simulate noisy channels, augment the feature vector with 57 additional dimensions, each one from a Gaussian distribution  $N(\mu=0; \sigma=4)$  regardless of class. As a result, you will have 600 (60-dimensional) examples.

- Compute the first two principal components of the data, and generate a scatter plot of all the examples. Use the 'text' function to plot examples by their class label. Use the 'axis' function to adjust the limits of the plot.
- Compute the Fisher's linear discriminant projections of the data, and generate a scatter plot of all the examples. Use the 'tamu\_lda.m' implementation included in 'hw2.zip'
- Split the dataset 50/50 into training and test sets. Re-compute the LDA eigenvectors using only the training set, and use the eigenvectors to project both training and test data. Mark the test samples with an asterisk, so you can visually tell them apart from the training samples.
- Discuss your results.

*HINT: Use the MATLAB command 'mvnrnd' to generate data from a multivariate Gaussian pdf.*

*NOTE:  $\sigma=4$  is the standard deviation, NOT the variance.*

#### Problem 4 (10%)

Implement a Quadratic classifier for Problem 3

- (a) Split the data into training and test sets by randomly selecting 25% of the examples from each class for the test set.
- (b) Classify the test data in the original high-dimensional space.
- (c) Repeat the above steps several times. What is the average classification rate?
  
- (d) Compute the PCA projections using ONLY the training data
- (e) Project both training and test data using the first three PCA eigenvectors
- (f) Classify the test data in a 3-dimensional PCA subspace.
- (g) Repeat the above steps several times. What is the average classification rate?
  
- (h) Compute the LDA projections using ONLY the training data
- (i) Project both training and test data using the LDA eigenvalues
- (j) Classify the test data in the LDA subspace.
- (k) Repeat the above steps several times. What is the average classification rate?
- (l) Discuss your results

#### Problem 5 (10%)

Repeat Problem 4, but this time implementing a KNN classifier. Experiment with the value of  $k$ .

#### Problem 6 (35%)

In the previous problems you had the opportunity to develop and evaluate two classifiers using synthetic data generated on your own. Things get a little more interesting now, because we are going to use a blind test to evaluate the performance of your implementations.

The dataset 'hw2p6\_train.mat' from 'hw2.zip' contains the following matrices:

- x1: training set (row vectors)
- clab1: training set labels

The dataset 'hw2p6\_test.mat' from 'hw2.zip' contains the following matrices:

- x2: test set (row vectors)
- clab2: test set labels

You are to:

- (a) Perform dimensionality reduction to visualize the structure of the data. How many eigenvalues should be used for PCA and LDA? Which technique does a better job at unfolding the structure of the data? Why?
- (b) Split your training (or test) set as in Problems 4 and 5 to determine which classification approach works best.
- (c) Discuss your findings. Can you reconcile the results in part (b) with the structure of the data and what you know about the two classifiers?
- (d) Prepare a MATLAB program called 'hw2p6.m' that will load 'hw2p6\_test.mat' and classify each of the examples in the dataset x2. Once you submit your code, I will run your 'hw2p6.m' program with a separate 'hw2p6\_test.mat' file containing my own test data. Your grade will be based on the performance of your classifier on my test data, which will contain a very large number of examples so I can

approximate the true error rate. Needless to say, all datasets will be generated from the same distribution.

NOTES:

- Please submit your code using the “turnin” utility at <https://csnet.cs.tamu.edu>. You should submit a single ZIP file (**your\_last\_name.zip**). Please refer to the syllabus for the late submission policy.
- When your code loads my separate test set, the class label vector `clab2` will obviously have dummy values, so your program should not attempt to use them. On the other hand, the class label vector on the test set in ‘`hw2.zip`’ does have correct values, so you can use them for validation.
- Make sure your code can handle ANY number of examples in `x2`.
- Make sure your code works!!! Are all required files included? You will receive no credit if the program returns with an error. I will run your code on `unix.cs.tamu.edu`, so you should test it there before submitting.
- To facilitate grading, your ‘`hw2p6.m`’ file should create a COLUMN VECTOR called ‘`uclab`’ containing the predicted class labels for each of the rows in `x2`:

```
uclab = [  
    1  
    4  
    2  
    1  
    3  
    ...  
];
```

These are the class predictions that I will compare against the true class labels of my separate test set, which I have kept aside.