Project Part 1 [15 points] PCA, Density Estimation, and Bayesian Classification (Due Tuesday, Nov. 10, 11:59pm)

This part of the project uses a subset of images (with modifications) from the MNIST dataset. The original MNIST dataset (http://yann.lecun.com/exdb/mnist/)) contains 70,000 images of handwritten digits, divided into 60,000 training images and 10,000 testing images. We use only images for digit "0" and digit "1" in this project, and the images have been slightly modified to suit this project.

The data are store them in ".mat" files. You may use the following piece of code to read the dataset in Python (or you may use the *load filename* command in Matlab, since these are .mat files):

```
import scipy.io
Numpyfile= scipy.io.loadmat('matlabfile.mat')
```

Following are the statistics for the data you are going to use:

Number of samples in the training set: "0": 5923 ; "1": 6742.

Number of samples in the testing set : "0": 980; "1": 1135

For the classification task, we assume that the prior probabilities are the same (i.e., P(0) = P(1) =0.5), although you may have noticed that these two digits have different numbers of samples in both the training and the test sets.

In the original .mat file, each image is stored as a 28x28 array. We need to "vectorize" an image by concatenating its columns to form a 784-dimensional vector. In the 784-d space, it would be difficult to apply Bayesian decision theory (e.g., the minimum error rate classification). Hence we will use PCA to do dimensionality reduction first.

Specifically, you will practice doing the following five tasks in this project:

Task 1. Feature normalization (Data conditioning).

You need to normalize the data in the following way, before starting any subsequent tasks. Using all the training images (each viewed as a 784-d vector, $X = [x_1, x_2, \ldots, x_{784}]^t$, as explained), compute the mean m_i and standard deviation (STD) s_i for each feature x_i (remember that we have 784 features) from all the training samples. The mean and STD will be used to normalize all the data samples (training and testing): for each feature x_i, in any given sample, the normalized feature will be, $y_i = (x_i - m_i)/s_i$

Task 2. PCA using the training samples.

Use all the training samples to do PCA. You cannot use a built-in function pca or similar, if your platform provides such a function. You have to explicitly code the key steps of PCA: computing the covariance matrix, doing eigen analysis (you can use built-in functions for this), and then identify the principal components.

Task 3. Dimension reduction using PCA.

Consider 2-d projections of the samples on the first and second principal components. These are the new 2-d representations of the samples. Plot/Visualize the training and testing samples in this 2-d space. Observe how the two classes are clustered in this 2-D space. Does each class look like a normal distribution?

Task 4. Density estimation.

We further assume in the 2-d space defined above, samples from each class follow a Gaussian distribution. You will need to estimate the parameters for the 2-d normal distribution for each class/digit, using the training data. Note: You will have two distributions, one for each digit.

Task 4. Bayesian Decision Theory for optimal classification.

Use the estimated distributions for doing minimum-error-rate classification. Report the accuracy for the training set and the testing set respectively.

What to submit:

1. Your code for doing the above.
2. A report summarizing the results, e.g., the estimated parameters of the distributions and the final classification accuracy number. Include in your report any intermediate results that you deem helpful for illustrating the partial results, like the "eigen digits" in Task 2, and the samples in the 2-d space in Task 3, etc.

   Note: There is no minimum or maximum length requirement for the report. Writing the report is the opportunity for you to reflect on your understanding of the problems/tasks through organizing your results.

The data files for the project are uploaded in the Files/Assignments folder:
training_data_0.mat, testing_data_0.mat,
training_data_1.mat, testing_data_1.mat