Project Part 1 [15 points] PCA, Density Estimation, and Bayesian Classification (Due Tuesday, Oct. 29, 11:59pm)

This part of the project uses a subset of images (with modifications) from the Fashion-MNIST dataset. The original Fashion-MNIST dataset contains 70,000 images of objects, divided into 60,000 training images and 10,000 testing images. We use only images for class "T-shirt" and class "Sneaker" in this project, and the images have been slightly modified to suit this project.

The data is stored 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
data = scipy.io.loadmat('matlabfile.mat')
```

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

Number of samples in the training set: "T-shirt": 6000; "Sneaker": 6000

Number of samples in the testing set: "T-shirt ": 1000; "Sneaker": 1000

For the classification task, we assume that the prior probabilities are the same (i.e., P(0) = P(1) =0.5).

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, using the training data. Note: You will have two distributions, one for each class.

Task 5. 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 with the following format-
   a. Introduction – start with problem statement, data description etc.
   b. Method – implementation details, steps followed etc.
   c. Results and observation – the results asked in each of the steps, e.g., the estimated parameters of the distributions and the final classification accuracy number (any intermediate results for each of the tasks you want to show) along with your observations
   d. Conclusion

   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.
3. The report should be typed *(handwritten reports are not allowed)* and in a .pdf format (to be submitted as separate document, not included within the code file).
4. Do not submit a .zip file. Submit multiple individual files on Canvas instead.

The data files for the project are uploaded in the Files/Assignments folder:
train_data.mat, test_data.mat