Assignment # 1

SYDE 675 Winter 2019

The assignment can be done in Python or Matlab.
You need to submit both your report and the source code implementation for all questions as one Zip file. The report must be a single pdf and the source code must be a single .py or .m file. Please include brief comments in your code. Be sure to label all figures and include a legend where appropriate.
The due date for this assignment is **Feb 21st, 2019**. Please also note that late submission will be subject to a penalty of 20% deduction of the assignment mark per day.

---

1. Consider two classes described by the covariance matrices below (assume zero mean)

$$a.\ \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad b.\ \Sigma = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}$$

   a. For each matrix generate 1000 data samples and plot them on separate figures.
   b. For each case calculate first standard deviation contour as a function of the mean, eigenvalues, and eigenvectors. Show your calculation (Hint: consider distribution whitening from the tutorial). You may use preexisting functions for Eigen computation. Plot each contour on the respective plots from part (a).
   c. Calculate sample covariance matrices for each class using the data generated in part (a). Do not use a Python/Matlab function for computing the covariance.
   d. Compare the given covariance matrix for each class with the corresponding sample covariance matrix generated in (b).

2. Consider a 2D problem with 3 classes where each class is described by the following priors, mean vectors, and covariance matrices.

$P(C_1) = 0.2$	$P(C_2) = 0.3$	$P(C_3) = 0.5$
$\mu_1 = [3 \quad 2]^T$	$\mu_2 = [5 \quad 4]^T$	$\mu_3 = [2 \quad 5]^T$
$\Sigma_1 = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}$	$\Sigma_2 = \begin{bmatrix} 1 & -1 \\ -1 & 7 \end{bmatrix}$	$\Sigma_3 = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 3 \end{bmatrix}$

a. Create a program to plot the decision boundaries for a ML and MAP classifier. Plot the means and first standard deviation contours for each class. Discuss the differences between the decision boundaries.
b. Generate a 3000 sample dataset using the prior probabilities of each class. For both the ML and MAP classifiers: classify the generated dataset, calculate a confusion matrix, and calculate the experimental P(ε). Discuss the results.

3. The MNIST dataset contains a set of images containing the digits 0 to 9. Each image in the data set is a 28x28 image. The data is divided into two sets of images: a training set and a testing set. The MNIST dataset can be downloaded from http://yann.lecun.com/exdb/mnist/. Use only the training set to perform this part.

a) Program PCA that takes X(DxN) and returns Y(dxN) where N is the number of samples, D is the number of input features, and d is the number of features selected by the PCA algorithm. Note that you must compute the PCA computation method by yourself. You may use preexisting functions for Eigen computation.
b) Propose a suitable d using proportion of variance (POV) =95%.
c) Program PCA reconstruction that takes Y_{PCA}(dxN) and returns $\hat{X}$ (DxN) (i.e., a reconstructed image). For different values of d= {1, 2, 3, 4, …, 784} reconstruct all samples and calculate the average mean square error (MSE). Plot MSE (y-axis) versus d (x-axis). Discuss the results.
d) Reconstruct a sample from the class of number '5' and show it as a 'png' image for d= {1, 10, 50, 250, 784}. Discuss the results.
e) For the values of d= {1, 2, 3, 4, …, 784} plot eigenvalues (y-axis) versus d (x-axis). Discuss the results.

4. Once again, consider the MNIST dataset. Use the training set as your training data and the test set as your test data.

a) Classify the test data using a kNN classifier. Report the accuracy for k = {1,3,5,11}. Justify and compare the reported accuracies for the different values of k. Do not use kNN implemented function in Python/Matlab and implement it by yourself.
b) Apply PCA to MNIST to create a new dataset MNIST-d. Classify the test samples in MNIST-d using a kNN classifier. For each d = {5, 50, 100, 500} use k = {1,3,5,11}. Calculate and display the classification accuracy for each of the 16 combinations of d and k in a table. Discuss the results.
c) Compare the reported accuracies in part (a) and part (b)