ECE763 Project 01

Due 2/16/2018

How to submit your solutions: put your reports (word or pdf) and results images (.png, if had) in a folder named *[your_unityid]_project01* (e.g., twu19_project01), and then compress it as a **zip** file (e.g., twu19_project01.zip). Submit the zip file through **moodle**.

Importance of working on Project 01 individually: Getting hands-on experience is important, which benefits you the most if you play with it individually and independently.

Objectives: Face image classification using Gaussian model, Mixture of Gaussian model, t-distribution, Mixture of t-distribution, Factor Analysis and Mixture of Factor Analyzer

Data Preparation. Prepare training and testing data. Go to https://github.com/betars/Face-Resources and select one of the provided 17 face datasets which has face bounding boxes annotated. Download the dataset (note that some of the 17 datasets might need registration to download and you can skip those to save time). Extract $n = 1000$ training images for face and non-face respectively, and $m = 100$ testing images for face and non-face respectively, both at 60×60 resolution (resizing accordingly). Make sure training face images and testing face images are separate, that is no face testing images are from the same person in the training set of face images. And, non-face images should be cropped randomly from background in the provided images in the dataset you selected.

Organized the extracted image patches Write a self-contained data i/o module in python or matlab or c/c++.

- Why doing this step? To get familiar with loading datasets, extracting image patches based on provided annotations. The datasets usually provide some toolkit for manipulating image and annotation which you can re-use and modify. This is the first step in almost of computer vision problems.
- You can try to use more training and testing images at your will.
- You can also try to use smaller or bigger patches so you need less or more computing power and memory footprint, depending on your laptop or workstation.
- You might need to extract more images initially, then manually prune those to make your dataset less messy. E.g., containing relatively clean faces, frontal and profile, without different types of occlusions (e.g., big glasses, mask, etc).

Tasks: With your own face dataset created, you can train your models and test the performance. For each model, report results as follows.

- *Visualize the estimated mean(s) and covariance matrix for face and non-face respectively*; Use RGB images, but you are welcome to try on other things such as gray images, gray images with histogram equalized, and HSI color space, etc.
- *Evaluate the learned model on the testing images using 0.5 as threshold for the posterior. Compute false positive rate (#negatives being classified as faces / #total negatives), and*

false negative rate (#positives being classified as non-face / #total positives), and the misclassification rate ((#false-positives + #false-negative)/#total testing images)

- *Plot the ROC curve where x-axis represents false positive rate and y-axis true positive rate (i.e, 1-false negative rate). To plot the curve, you change the threshold for posterior from $+\infty$ (use maximum posterior across testing images in practice, then all being classified as non-faces) to $-\infty$ (use minimum posterior across testing images in practice, then all being classified as faces) with for example1000 steps. (ref: https://en.wikipedia.org/wiki/Receiver_operating_characteristic)*

Model 1. Learn single Gaussian model using training images and report your results as stated above.

Model 2. Learn Mixture of Gaussian model using training images and report your results as stated above. You can tune the number of components (e.g., based on cross validation strategy).

Model 3. Learn t-distribution model using training images and report your results as stated above.

Model 4. Learn Mixture of t-distribution model using training images and report your results as stated above. You can tune the number of components (e.g., based on cross validation strategy).

Model 5. Learn factor analyzer using training images and report your results as stated above. You can tune the dimensionality of the subspace.

Model 6. Learn Mixture of t factor analyzer model using training images and report your results as stated above. You can tune the dimensionality of the subspace and the number of components (e.g., based on cross validation strategy).

Hint: Think about how to modularize your codes in a nice way. For example, you will only need one ROCplot functions, and you can write a common EM algorithm. It is important to learn how to connect understanding of mathematical derivation to actual efficient implementation.