Pattern Recognition and Machine Learning : Assignments

- You are expected to upload a soft copy of the code and a report highlighting your inferences on the moodle page.
- **Plagiarism of any kind will result in a failing grade in the course. This also includes downloading codes that are freely available from the internet and using it**.

Task 1: Character Recognition using Bayesian Classifier

In this assignment, I expect you to build a rudimentary pattern recognizer by exploring out on the Bayesian classifier concepts, that we discussed in the class. To this goal, you are given training images of 3 characters in a folder named **TrainCharacters.rar**. Each of the 200 training images of each class are of size 128×128. You are given 300 test images (100 in each class) of size 128×128 in a separate folder **TestCharacters.rar**.

Assume the samples to be generated from a multi dimensional Gaussian distribution, having class specific mean vectors $\boldsymbol{\mu_i}$. Consider each of the modelling schemes for computing the covariance matrix.

(i) The samples of a given character class are modelled by a separate covariance matrix $\boldsymbol{\Sigma_i}$.

(ii) The samples across all the characters are pooled to generate a common diagonal covariance matrix $\boldsymbol{\Sigma}$. The diagonal entries correspond to the variances of the individual features, that are considered to be independent.

(iii) The covariance matrix of each class is forced to be identity matrix.

For each scenario, build a generative Bayesian classifier using the training images and categorize the characters contained in the test folder. The mean and the covariance matrices are to be estimated from the training data using the Maximum Likelihood techniques, as discussed in class. Report the individual character accuracies as well as the averaged accuracy for each of the models.

Employ the 128×128 pixel intensity values directly as features in this task. If you happen to encounter memory storage issues during simulation, you may consider resizing the images to a more manageable size (say 32×32) for the feature computation. However, note that in order to beat the curse of dimensionality, you have to add a regularization term of the form $\lambda \mathbf{I}$ in the computation of the covariance matrix.

1(b)

Give 4 examples of images from the test set that are misclassified by each of the classifiers designed in Task 1(a). You need to display both the state of nature (true label) and the predicted class for each image.

Task 2: GMM based clustering

Implement a GMM based clustering framework on the image 'building.jpg'. You may choose 3 Gaussian components and use the RGB values as the features. Start the iterations using the means $\boldsymbol{\mu_1} = \begin{bmatrix} 120 \\ 120 \\ 120 \end{bmatrix}$ $\boldsymbol{\mu_2} = \begin{bmatrix} 12 \\ 12 \\ 12 \end{bmatrix}$ and $\boldsymbol{\mu_3} = \begin{bmatrix} 180 \\ 180 \\ 180 \end{bmatrix}$. Also assume an identity covariance matrix for each of the components.

- Dispay the segmented output.
- Display a graph depicting the convergence of the log likelihood values.

Task 3: Face Recognition using PCA

The gallery folder **'Gallery.zip'** contains images from 40 individuals, each of them providing 5 images. The pixel intensities of the 200 face images will be used for computing the KL Transform. By employing the method of efficient computation of the basis vectors for high dimensional data (discussed in class),

(i) Display the Eigenface images corresponding to the top 5 Eigen values of the covariance matrix $\boldsymbol{\Sigma}$.

(ii) Plot a graph depicting the percentage of the total variance of the original data retained in the reduced space versus the number of dimensions. From this graph, find the number of dimensions required for projecting the face vectors so that at least 95% of the total variance of the original data is accounted for in the reduced space.

(iii) Reconstruct the image 'face_input_1.pgm' using the:

(a) Eigenface corresponding to the largest eigenvalue.

(b) Top 15 Eigenfaces

(c) All the Eigenfaces

Display the reconstructed image and the mean squared error in each case.

(iv) Depict graphically the mean squared error obtained for different number of Eigenfaces. Note that the Eigenfaces will vary from 1 to 200 (total number of training samples).

(v) Repeat the parts (iii) and (iv) for the image 'face_input_2.pgm'. Comment on your result.

Task 4 : Support Vector Machines

- Build a SVM Classifier for the classes ω_1 , ω_2 and ω_3 using a Radial Basis Function Kernel. Plot a graph depicting the recognition accuracy on the test data `Test1.mat`, `Test2.mat` and `Test3.mat` for different values of penalty factors C and precisions γ of the Radial Basis Function. The software package that I recommend for this task is `LibSVM`. On their webpage, download the version (either Matlab/Python), appropriate to the platform in which you are working.