Project 3 - Dimensionality Reduction and Performance Evaluation (Due 10/15)

Objective:

The objective of this project is, first of all, to practice the usage of dimensionality reduction as one of the preprocessing steps and how to draw an ROC curve and confusion matrix as part of the post-processing (performance evaluation) steps. You need to use both supervised dimensionality reduction method (i.e., FLD) and unsupervised method (i.e., PCA) for that purpose. We will continue using pima as the dataset but in a more comprehensive way. The second objective is to extend the horizon of machine learning applications and solve a seemingly quite unrelated problem -- image compression. Think hard if image compression should be solved using supervised learning or unsupervised learning, and what are the features in this application.

Data Sets:

Two datasets will be used. The first is the Pima Indians dataset with 7 features. The second is a beautiful color image of flowers.

- The Pima dataset: pima.tr (the training set) and pima.te (the test set). Preprocessing you need to do: (Please note the change of terminology used!)
  - 1) Remove the first row with any text editor; Change the labels from "Yes" and "No" to "0" and "1" respectively indicating 'with disease' and 'without disease' - you can do this using the same text editor; (Note that intuitively "Yes" should be 1 and "No" be 0. So both labeling schemes would be fine.
  - 2) Standardize ~~Normalize~~ the data set to make the features comparable (or with the same scale). Suppose x is a data sample, m_i is the mean of each feature i, sigma_i is the standard deviation of each feature i, then standardization ~~normalization~~ is conducted by (x-m_i)/sigma_i. Keep in mind that you also need to standardize ~~normalize~~ the samples in the test set. Be careful which mean and standard deviation you should use. (For each sample in the test set, use the same m_i and sigma_i you derived from the training set.)
- The flower image.

  The image given is a 120x120 full-color image. It can also be treated as a 120 x 120 x 3 matrix, representing a 3-dimensional feature space of 120 x 120 samples. A sample code will be posted on how to convert a color image to/from a 3-dimensional dataset.

Performance Metrics:

Besides the three metrics used in Projects 1&2, i.e., 1) overall classification accuracy, 2) classwise classification accuracy, and 3) run time, we'll introduce a fourth metric that measures the quality of the compressed image as compared to the original full-color image - 4) root mean squared error (RMSE). We'll also add confusion matrices and ROC curves to performance evaluation.

Tasks:

- Task 1: Implement both supervised and unsupervised dimensionality approaches (FLD and PCA). Only the pima dataset is used.
  - (20 pts) Denote the standardized ~~normalized~~ dataset as nX. You need to prepare three projected datasets from nX for later classification purpose.
    - fX: the projected data from FLD. You should realize since pima only has c=2 classes, the number of dimensions it will reduce to should be m = c - 1 = 1 dimension.
    - pX1: the projected data from PCA (only keep the major axis, i.e., the eigenvector that corresponds to the largest eigenvalue). Calculate the error rate introduced by pX1.

        - pX: the projected data from PCA assuming the error rate you can tolerate is no greater than 15%. How many dimensions need to be kept?
        - Plot a 1 x 2 figure showing the histogram of fX and pX1. Do they look like Gaussian to you? Comment on what distribution you think would be appropriate to describe these 1-D datasets? This should give you some hint on if non-parametric learning might be a better option or not.
    - (35 pts) Apply both Case 3 and kNN on nX, fX, px1, and pX. Use the "k" where you obtain the best performance on nX in Project 2.
        - Draw two tables of 4 rows and 4 columns that measures overall accuracy, classwise accuracy, and runtime. Assuming 1:3 prior probability.
        - Generate 8 confusion matrices, i.e., the two classifiers on the four datasets. (need to generate them yourself)
        - Plot ROC curves (in the same figure) using only pX. (Note that the different points on the curves should be generated by changing the prior probabilities)
- Task 2: Use kmeans and wta you implemented in Project 2 to solve the image compression problem. Each pixel of this color image has three components: red, green, and blue. Each component is an 8-bit unsigned char. That is, each pixel is represented using 24 bits, a total of 2^24 possible colors. You are asked to use less number of bits to represent each pixel. For example, if you want to only use 256 colors to represent the original full-color image, then you are basically only using 8 bits to represent each pixel, instead of 24 bits. We refer to the color image not showing its full-color potential as pseudo-color image.
    - (30 pts) Draw a table with 8 rows and 2 columns showing the generated pseudocolor images with k = 256, 128, 64, and 32 different colors using kmeans and wta. Underneath each image, display the reconstruction error measured in terms of RMSE.
    - (10 pts) Comment on the results both visually and through quantitative measurement (i.e., RMSE).
- Final discussion (5 pts).