PATTERN RECOGNITION -- Spring 2019

Assignment 2: Dimensionality Reduction

DUE: Before 12midnight on 06 Apr 2019 (Saturday)

INSTRUCTIONS:

i. Please do the assignment in Python.

ii. You need to submit pdf files to the TAs. One file should contain your answers, results and analysis. A separate file should contain code you have written and its sample output.

iii. At the top-right of the first page of your submission, include the assignment number, your name and roll number.

iv. *IMPORTANT:* Make sure that the assignment that you submit is your own work. *Do not directly copy any part from any source* including your friends, seniors or the internet.

v. Your grade will depend on the correctness of answers and output. In addition, due consideration will be given to the clarity and details of your answers and the legibility and structure of your code.

Preamble:

The aim of this assignment is to experiment with *dimensionality reduction* techniques we learned in the class on real world problems.

(1) Implement dimensionality reduction techniques such as Principal Component Analysis (PCA) and Fisher's Linear Discriminant Analysis (LDA) using your own code.

(2) Apply PCA and LDA on IRIS dataset. Show projection of the original data in PCA space (PC1 versus PC2; PC1 versus PC3 and PC2 versus PC3) and 1-dimensional LDA space. Please label each point with their class labels for better visualization. Now, comment on the similarities and differences of results obtained in these two approaches.

(3) Apply PCA to a high dimensional dataset such as UCI Arcene cancer classification data. Generate the Scree Plot. Comment on how many components to choose for explaining 85%, 90%, 95% and 99% of the variance of the data. Project the data into first two PCs as well as 1-dimensional LDA space (To avoid clutter, plot only 10% of the data but equal number from each class. Use class labels on the data points to visualize better) and comment on the "representation" of the data in the lower dimensional subspace.

(4) Use kernel PCA (kPCA) and Local Linear Embedding (LLE) from any python package (for example, *scikit-learn*)on synthetic datasets (such as 3D *swissroll* dataset). Compare the results from your PCA with those of kPCA and LLE. You can also experiment by introducing punctures in the *swiss roll* manifold to see what ordinary PCA versus kPCA and LLE do to this data in 2D projection.

As part of the submission include the code for each of the algorithms along with a small report that explains the algorithms, implementation details, the results and their analysis.