CS 383 - Machine Learning

Assignment 1 - Dimensionality Reduction

Introduction

In this assignment, in addition to related theory/math questions, you'll work on visualizing data and reducing its dimensionality.

You may not use any functions from machine learning library in your code, however you may use statistical functions. For example, if available you **MAY NOT** use functions like

- pca
- entropy

however you **MAY** use basic statistical functions like:

- std
- mean
- cov
- eig

Grading

Although all assignments will be weighed equally in computing your homework grade, below is the grading rubric we will use for this assignment:

Part 1 (Theory)	30pts
Part 2 (PCA)	40pts
Part 3 (Eigenfaces)	20pts
Report	10pts
TOTAL	100pts

Table 1: Grading Rubric

DataSets

Yale Faces Datasaet This dataset consists of 154 images (each of which is 243x320 pixels) taken from 14 people at 11 different viewing conditions (for our purposes, the first person was removed from the official dataset so person ID=2 is the first person).

The filename of each images encode class information:

$$\text{subject} < ID > . < condition >$$

Data obtained from: http://cvc.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html

1 Theory Questions

1. (15 points) Consider the following data:

$$\text{Class 1} = \begin{bmatrix} -2 & 1 \\ -5 & -4 \\ -3 & 1 \\ 0 & 3 \\ -8 & 11 \end{bmatrix}, \text{Class 2} = \begin{bmatrix} -2 & 5 \\ 1 & 0 \\ 5 & -1 \\ -1 & -3 \\ 6 & 1 \end{bmatrix}$$

   (a) Compute the information gain for each feature. You could standardize the data overall, although it won't make a difference. (13pts).

   (b) Which feature is more discriminating based on results in Part (a) (2pt)?

2. (15 points) In principle component analysis (PCA) we are trying to maximize the variance of the data after projection while minimizing how far the magnitude of w, $|w|$ is from being unit length. This results in attempting to find the value of w that maximizes the equation

$$w^T \Sigma w - \alpha(w^T w - 1)$$

   where Σ is the covariance matrix of the observable data matrix X.

   One problem with PCA is that it doesn't take class labels into account. Therefore projecting using PCA can result in worse class separation, making the classification problem more difficult, especially for linear classifiers.

   To avoid this, if we have class information, one idea is to separate the data by class and aim to find the projection that maximize the distance between the means of the class data after projection, while minimizing their variance after projection. This is called **linear discriminant analysis** (LDA).

   Let C_i be the set of observations that have class label i, and μ_i, σ_i be the mean and standard deviations, respectively, of those sets. Assuming that we only have two classes, we then want to find the value of w that maximizes the equation:

$$(\mu_1 w - \mu_2 w)^T (\mu_1 w - \mu_2 w) - \lambda((\sigma_1 w)^T (\sigma_1 w) + (\sigma_2 w)^T (\sigma_2 w))$$

   Which is equivalent to

$$w^T (\mu_1 - \mu_2)^T (\mu_1 - \mu_2) w - \lambda(w^T (\sigma_1^T \sigma_1 + \sigma_2^T \sigma_2) w)$$

   Show that to maximize this we must find the eigenvector/eigenvalue pairs, (w, λ) **for the equation:**

$$(\sigma_1^T \sigma_1 + \sigma_2^T \sigma_2)^{-1}((\mu_1 - \mu_2)^T (\mu_1 - \mu_2))w = \lambda w$$

2 (40pts) Dimensionality Reduction via PCA

Download and extract the dataset *yalefaces.zip* from Blackboard. This dataset has 154 images ($N = 154$) each of which is a 243x320 image ($D = 77760$). In order to process this data your script will need to:

1. Read in the list of files
2. Create a 154x1600 data matrix such that for each image file
   (a) Read in the image as a 2D array (234x320 pixels)
   (b) Subsample the image to become a 40x40 pixel image (for processing speed)
   (c) *Flatten* the image to a 1D array (1x1600)
   (d) Concatenate this as a row of your data matrix.

Once you have your data matrix, your script should:

1. Standardizes the data
2. Reduces the data to 2D using PCA
3. Graphs the data for visualization

Recall that although you may not use any package ML functions like *pca*, you **may** use statistical functions like *eig*.

Your graph should end up looking similar to Figure 1 (although it may be rotated differently, depending how you ordered things).

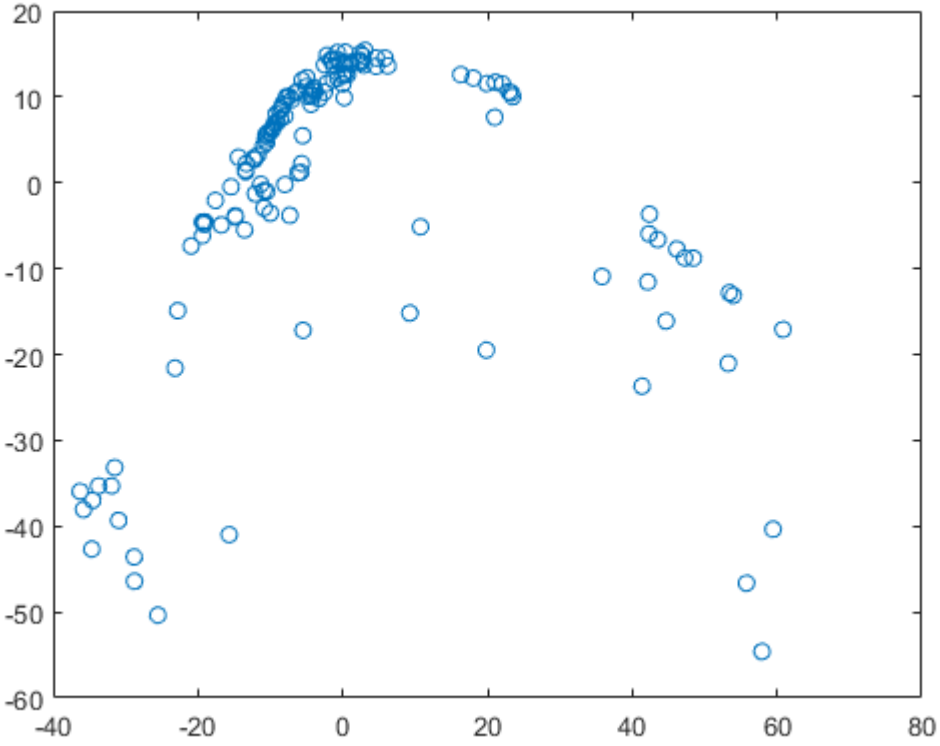

Figure 1: 2D PCA Projection of data

3 (20 points) Eigenfaces

Download and extract the dataset *yalefaces.zip* from Blackboard. This dataset has 154 images ($N = 154$) each of which is a 243x320 image ($D = 77760$). In order to process this data your script will need to:

1. Read in the list of files
2. Create a 154x1600 data matrix such that for each image file
   (a) Read in the image as a 2D array (234x320 pixels)
   (b) Subsample the image to become a 40x40 pixel image (for processing speed)
   (c) *Flatten* the image to a 1D array (1x1600)
   (d) Concatenate this as a row of your data matrix.

Write a script that:

1. Imports the data as mentioned above.
2. Standardizes the data.
3. Performs PCA on the data (again, although you may not use any package ML functions like *pca*, you **may** use statistical functions like *eig*).
4. Determines the number of principle components necessary to encode at least 95% of the information, k.
5. Visualizes the most important principle component as a 40x40 image (see Figure 2).
6. Reconstructs the first person using the primary principle component and then using the k most significant eigen-vectors (see Figure 3). For the fun of it maybe even look to see if you can perfectly reconstruct the face if you use all the eigen-vectors!

Your principle eigenface should end up looking similar to Figure 2.

Figure 2: Primary Principle Component

Your reconstruction should end up looking similar to Figure 3.

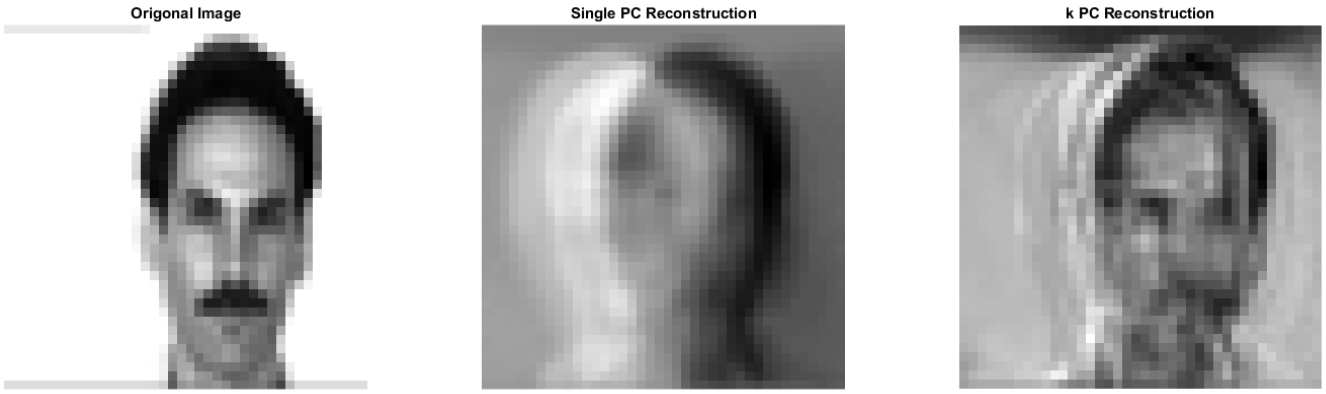

Figure 3: Reconstruction of first person (ID=2)

Submission

For your submission, upload to Blackboard a single zip file containing:

1. PDF Writeup
2. Source Code
3. readme.txt file

The readme.txt file should contain information on how to run your code to reproduce results for each part of the assignment. Do not include spaces or special characters (other than the underscore character) in your file and directory names. Doing so may break our grading scripts.

The PDF document should contain the following:

1. Part 1: Your answers to the theory questions.
2. Part 2: The visualization of the PCA result
3. Part 3:
   (a) Number of principle components needed to represent 95% of information, k.
   (b) Visualization of primary principle component
   (c) Visualization of the reconstruction of the first person using
      i. Original image
      ii. Single principle component
      iii. k principle components.