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
- svd
- 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)	15pts
Part 2 (PCA)	40pts
Part 3 (Eigenfaces)	30pts
Report	15pts
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 (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/resize the image to become a 40x40 pixel image (for processing speed). I suggest you use your image processing library to do this for you.
   (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. Plot the data as points in 2D space for visualization

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

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

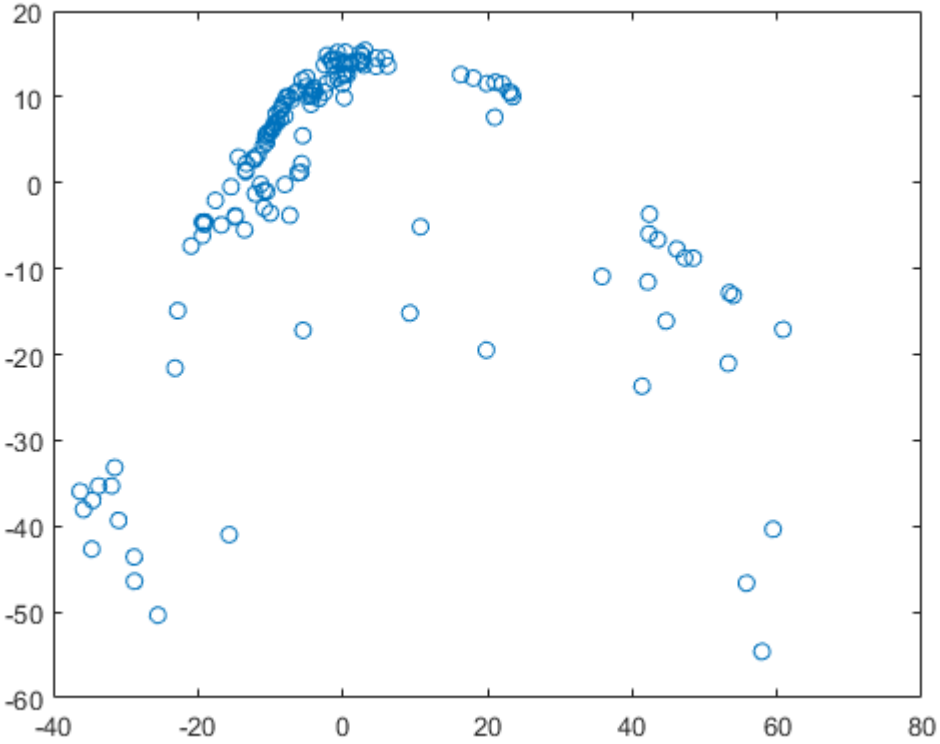

Figure 1: 2D PCA Projection of data

NOTE: Depending on your linear algebra package, the eigenvectors may have the opposite direction. This is fine since technically an eigenvector multiplied by any scalar are equivalent.

3 (30 points) Eigenfaces

Take your original $154x1600$ standardized data matrix from the previous problem and

Write a script that:

1. Performs PCA on the data (again, although you may not use any package ML functions like *pca*, you **may** use statistical functions like *svd, eig*).

2. Takes the most important principle component, reshapes it to be a 40x40 matrix, then visualizes this as an image (see Figure 2). This is often referred to in literature as an *eigenface*.

3. Projects *subject02.centerlight* using the primary principle component, then *un-projects (reconstructs)* this person to return to the original feature space. To best see the full re-construction, "unstandardize" the reconstruction by multiplying it by the original standard deviation and adding back in the original mean.

4. Determines the minimal number of principle components necessary to capture 95% of eigenvalues, k.

5. Uses the k most significant eigen-vectors to project, then reconstructs *subject02.centerlight* (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! Again, to best see the full re-construction, "unstandardize" the reconstruction by multiplying it by the original standard deviation and adding back in the original mean.

NOTE: In order to view your matrix as an image you may need to adjust the values to fit in the range that your image viewing function expects. Read its documentation.

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

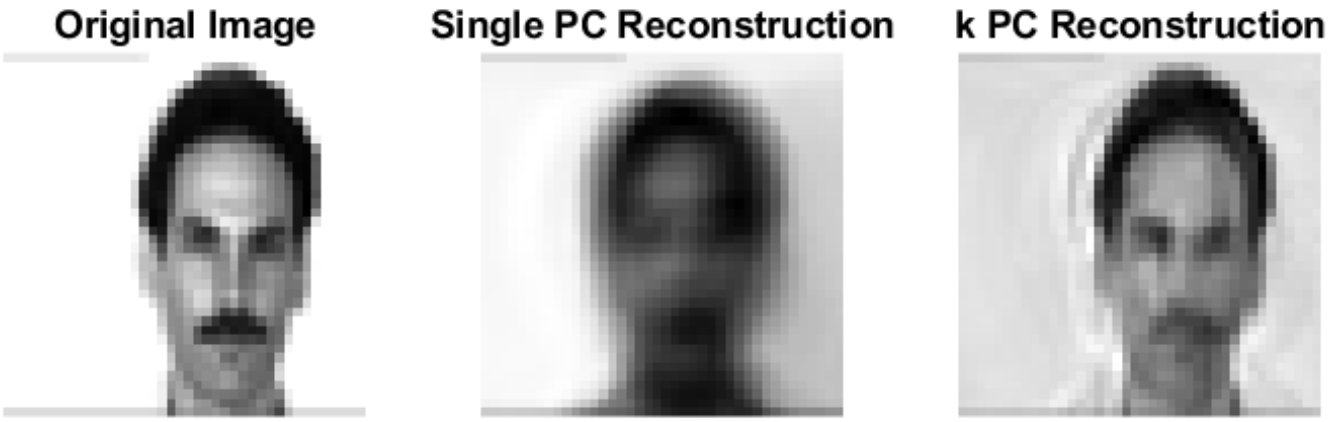

Figure 3: Reconstruction of first person, post-unstandardized (ID=2)

Submission

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

1. PDF Writeup
2. Source Code
3. readme.txt file
4. You do not need to include the images.
5. You **no not** need to include the dataset. HOWEVER, it should be clear in your script (and/or readme) where your code expects the dataset to reside.

The readme.txt file should contain information on how to run your code to reproduce results for each part of the assignment. In particular for this assignment, it should also indicate where the *yalefaces* directory should be in order to run. 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.