CS 613 - Machine Learning

Assignment 4 - Dimensionality Reduction

Introduction

In this assignment you'll work on reducing data dimensionality for improved generalizability, visualization, and compression.

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
- k-nearest neighbors functions

Unless explicitly told to do so. But you **MAY** use basic statistical functions like:

- std
- mean
- cov
- eig
- svd

Grading

Part 1 (Theory)	25pts
Part 2 (PCA for Visualization	10pts
Part 3 (PCA and KNNs)	35pts
Part 4 (Eigenfaces as Compression)	30pts
TOTAL	100pts

Table 1: Grading Rubric

DataSets

Labeled Faces in the Wild Datasaet This dataset consists of celebrities download from the Internet from the early 2000s. Information about the entire dataset can be found here:

http://vis-www.cs.umass.edu/lfw/

This dataset has been pre-processed significantly. In particular:

- Only people with at least 20 images are included (which is 62 people).
- Only the first 20 images of those people are included (for a total of 1240 images.
- The images have been alligned, cropped, and converted to grayscale, resulting in images that have 87×65 pixels.
- The images have been placed in a *csv* file, **lfw20.csv** such that each row pertains to an image, the first column is the person $ID \in [0, 61]$, follwed by 1240 columns pertaining to the pixels of this image.

Sample images can be seen below:

1 Theory Questions

All of the theory questions will use the following data:

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

1. Zscore the data and create a 2D plot of the datapoints, visualizing class one data as squares, and class two data as circles (5pts).

2. PCA

   (a) Find the principle components of the data. You may use an *eig* function, but show the math leading up to, and after, using that function. Make sure that your principle components are normalized to be unit length. (5pts).

   (b) Project your data down to 1D using the principle component associated with the largest eigenvalue. Plot this data in 1D, visualizing class one data as squares, and class two data as circles (3pts).

   (c) Based on your plot, does the PCA projection provide good class separation? Why or why not (2pts)?

3. LDA

   (a) Using LDA, find the direction of projection (show your work in detail similar to the prior question). Normalize this vector to be unit length (5pts).

   (b) Project your data down to 1D using the the direction of projection found from the LDA process. Plot this data in 1D, visualizing class one data as squares and class two data as circles (3pts).

   (c) Based on your plot, does the LDA projection provide good class separation? Why or why not (2pts)?

2 Dimensionality Reduction for Visualation

First let's visualize our data.

Import the labeled faces in the wild dataset and display an image from this dataset. This will require reshaping a row of the data matrix to form an 87×65 image matrix (this may also requiring doing some tranposes).

Next we're going to visualize our dataset in 2D. To do this we'll use principle component analysis.

Write a script that:

1. Dviides all the pixel values by 255 so that they are now in the range [0,1] (this will provide some numeric stability).
2. Zscores the data.
3. Computes the principle components of the data.
4. Projects the data onto the two most relevant principle components and plots these points in 2D space.
5. *Whitens* the projected data and then once again plots these points in 2D space.

Include in your report:

- The image you displayed.
- Your plot for the (non-whitened) 2D PCA projected data.
- Your plot for the whitened 2D PCA projected data.

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

3 Dimensionality Reduction for KNNs

Next we're going to demonstrate how PCA can be used to reduce overfitting (and therefore help with generalization). To do this you'll also implement a k-nearest neighbors classifier (KNN). It's worth noting that a KNN is likely not the best tool for the job of image classification, but it will still allow us to illustrate our point (while getting experience implementing KNN).

Write a script that:

1. Imports the labeled faces in the wild dataset and scales the pixel values to [0,1] .
2. Separates the data into training and validation samples, and zscores both sets using the training data.
3. Computes the validation accuracy using KNN for the following scenarios:
   (a) $k = 1$ $D = original$
   (b) $k = 1$ $D = 100$ as reduced using PCA.
   (c) $k = 1$ $D = 100$ as reduced using PCA then *whitened*.

Implementation Details:

- Although we only asked for results when $k = 1$, your code should work for any $1 \leq k \leq N$, where N is the number of observations in the training data.
- For your distance measurement, use the squared euclidean distance.
- Once again, although you may not use any package ML functions like *pca*, you **may** use statistical functions like *eig* or *svd*. In addition, you cannot use an KNN related functions/methods.

Include in your report the accuracy for all three scenarios.

4 Eigenfaces as Compression

Finally we're going to see how we can use PCA for compression.

Write a script that:

1. Imports the labeled faces in the wild dataset and scales the pixel values to [0,1] and zscores *all* the data.
2. Performs PCA on all the data.
3. Visualizes the principle component related to the largest eigenvalue by reshaping it to an 87×65 matrix, and displaying it as an image.
4. Projects a person (row of the data) onto the principle component related to the largest eigenvalue, reconstructs the person using that projection and the same principle component, and displays that reconstructed image.
5. Determines how many principle components are necessary to captue 95% of the information, then projects the person onto these components, reconstructs them, and displays the reconstructed image.

Additional Notes:

- As part of the reconstruction you will likely want to "unzscore" your reconstructed data.
- If you use all the components you should be able to perfectly reconstruct someone. You can use this idea to test your code.

Include in your report:

- The primarly component, visualized as an image.
- The image that you are using for compression/reconstruction.
- The reconstructed image using one component.
- The minimum number of components necessary to perform 95% reconstrution.
- The reconstructed image using the components found in the previous part.

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.

The PDF document should contain the following:

1. Part 1:
   (a) Answers to Theory Questions
2. Part 2:
   (a) One of the original images
   (b) Your plot for the non-whitened 2D PCA projected data.
   (c) Your plot for the whitened 2D PCA projected data.
3. Part 3:
   (a) Accuracy for the three requested scenarios.
4. Part 4:
   (a) The primarly component, visualized as an image.
   (b) The image that you are using for compression/reconstruction.
   (c) The reconstructed image using one component.
   (d) The minimum number of components necessary to perform 95% reconstrution.
   (e) The reconstructed image using the components found in the previous part.