Class #19: Statistical Analysis, Principal Components Analysis

Purpose: The objective of this experiment is to familiarize yourself with common statistical analysis terms and revisit linear curve fitting with multiple inputs.

Background: Before doing this experiment, students should be able to

- Compute determinants and perform matrix multiplication
- Use Matlab to perform compute eigenvalues and eigenvectors
- Review the background for the previous experiments.

Learning Outcomes: Students will be able to

- Understand the properties of Multi variate Gaussian distributions
- Understand the concept of Eigenvalues and Eigenvectors
- Use matrix mathematics to determine how the Eigenvalues and Eigenvectors relate to the covariance matrix

Equipment Required:

- Matlab

Keywords:

- Multivariate Gaussian
- Eigenvalues
- Eigenvectors
- Mean
- Covariance Matrix
- Principal Components Analysis

Helpful links for this experiment can be found on the course website under Class #19.

Part A – Image Processing in Matlab

Background

In the last experiment, we worked with multivariate Gaussians, and we understood how the eigenvalues and eigenvectors of the covariance matrix are related to the distribution of the data. In this experiment we will take these ideas to process images. Images are not distributed according to Gaussian distributions, but the main ideas still generalize. We will be working with the black and white faces dataset in Figure A-1.

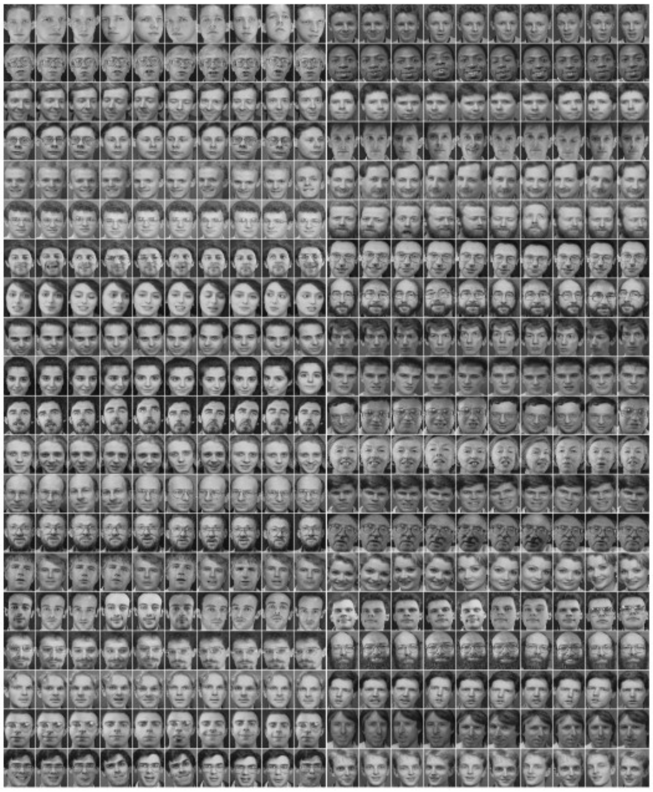

Figure A-1: Faces Dataset

To process these images, we will be using Matlab. The dataset has 10 pictures of each of the 40 subjects. The way that Matlab represents pictures is by using matrices: each picture is matrix whose size is the number of pixels in the image. In this dataset the matrices are 112 x 92. Say that the image is saved as a variable with the name `image.` To print the image the following commands are required

```
>> colormap(gray);
>> imagesc(image);
>> axis square;
```

You can download from the website of the class the dataset and a file called `main.m`. Extract the images and save the main file at the same level as the directory containing all the images. The script `main.m` will take care of loading all the images from the dataset and saving them into a three-dimensional matrix called `dataSet.` You can load image number n (with n between 1 and 400) by running

```
>> image = dataSet(:,:,n);
```

Exercise

1) Download the dataset and the script `main.m.` Run the script, make sure that the dataset is created, and the image is displayed.
2) An important quantity of any distribution is the mean. Since each image is a sample from a “face distribution”, we can try to compute its mean (or the average face). Use the command `mean` to do so. To get more information how it works with three dimensional arrays you may want to type on Matlab

```
>> help mean
```

What is the dimension that the result of this computation should have? Plot the mean face (I know, the average person is not very good looking).

Part B – Principal Components Analysis

Background

When processing information it is fundamental to understand that not all data is equally important. In this dataset images have $112 \times 92 = 10304$ pixels and we have 400 images making for over four million pixels. We will try to reduce this number to the order of 50 thousand pixels. How can one achieve such improvement? By keeping only what is important. But how can we know what information is important? We can look at the eigenvalues of the covariance matrix with larger value. Are these few parameters enough for the human eye to reconstruct the full image? Not really, we still need in the order of two million, however the computers can recognize the person with much less information (more of this in the next class). For a better reconstruction in terms of how the image looks we can use the Fourier Transform (Signals and Systems covers this topic in detail, but it is also related to eigenvalues and eigenvectors of a different matrix).

Images are stored in Matlab as matrices; however, it is more convenient to reshape them into vectors for processing. We can do that with the following command

```
>> vectorized_image = reshape(image,[112*92,1]);
```

If we think of each image as a column vector x with length 10304 (this is the result of multiplying 112 times 92) then the covariance matrix Σ is a square matrix of dimension 10304 times 10304. With Matlab is relatively easy to get the largest eigenvalues and eigenvectors. If the covariance matrix is stored in Matlab as the variable `Sigma`, to get the five largest eigenvalues and eigenvectors it suffices to run

```
>> [V,D] = eigs(Sigma,5);
```

If you want the 10 largest eigenvalues and eigenvectors, you would run

```
>> [V,D] = eigs(Sigma,10);
```

Each of the columns of the matrix `V` is an eigenvector. If we keep K eigenvectors (where K in the previous examples is 5 or 10) the matrix V is of dimensions 10304 x K. What is the Principal Component (PC) decomposition? The result of multiplying the original image by the matrix V transposed. Mathematically this is

$$x_{pc} = V^T(x - \mu)$$

Where μ is the mean face. In Matlab code this looks like

```
>> pc_image = transpose(V)*(vectorized_image-vectorized_mean_face);
```

To reconstruct the image, it suffices to perform the inverse operation

$$x = V\,x_{pc} + \mu$$

In Matlab code this looks like

```
>> reconstructed_image = V*pc_image+vectorized_mean_face;
```

Exercise:

a) We will try to understand the information reduction process. We will start by computing the mean and the covariance matrix of the vectorized image. Fill the code skeleton in the main file to do so. What is the dimension of the vectorized mean face μ? What is the dimension of the covariance matrix Σ?

b) If we decide to keep K eigenvectors (the principal components), what is the dimension of V? Complete the skeleton of the code to get the 5 largest eigenvalues.

c) Take any image of the data set in vector form and compute x_{pc}. Complete the code skeleton. What is the dimension of x_pc?

d) Give an expression of the total number of parameters that one needs to reconstruct all the images in terms of K. What is the total number of parameters that you need if K = 5? What is the ration of information that we are keeping with K=5? This is `dataNeededForReconstruction/originalData`.

e) Add a line that performs the reconstruction of the desired image and display the original image and the reconstructed version.

f) Since we are selecting only 5 coefficients you cannot recognize the person. How many coefficients do you need to start recognizing the subject? How many coefficients do you need to have a decent reconstruction? What is the ratio of information that we are keeping in these cases?