Instituto Superior Técnico

Departamento de Engenharia Electrotécnica e de Computadores

Machine Learning

6^{th} Lab Assignment

Shift_____ Group number_____

Number_____ Name_____

Number_____ Name_____

Principal Components Analysis (PCA)

1 Introduction

Principal Components Analysis is a classical linear method often used in multivariate analysis and data compression.

This assignment addresses two applications of PCA, one for representing a data set in a reduced dimensionality space, and the other one illustrating how PCA can be used in image compression.

1.1(T) Assume that a certain autocorrelation matrix of size $n \times n$ has n eigenvalues that are different from one another. Give a mathematical proof that there is an orthonormal basis of $\mathbb{R}^n$ which is formed by eigenvectors of that matrix. If you use any properties of autocorrelation matrices, of eigenvectors, or of the eigendecomposition of matrices, you should prove that those properties are true.

R.

2 PCA for multivariate analysis

A useful application of PCA is the reduction of noise present in high-dimensional data, when the relevant data span an affine space of lower dimensionality. The file `pts.txt` contains a set of 354 points in a 20-dimensional space. This file has one column for each data point. Load the file using the command `x=load('points.txt');`. Check the size of matrix `x` using the function `size`.

2.1(E) Try to find any easily understandable information in `x` by making 2-D plots of the 354 points, using various pairs of coordinates (e.g. x_1 versus x_2, x_5 versus x_9). Use dots to represent the points, without any interconnecting lines. Describe what you observe (you don't need to sketch the plots).

R.

2.2(T) Indicate the Matlab commands required to find the principal directions of these data, taking into account the way in which the data are arranged in matrix `x`. You should not use any ready-made Matlab functions or scripts for performing PCA. You may find the functions `eig` and `sort` useful.

R.

2.3(E) Perform PCA using the above commands. Indicate the minimum number of components that represent at least 95% of the energy of the data. Indicate the computations that you performed to find that number.

R.

2.4(E) Consider now the two first principal components. Indicate the energy associated with these components. Then reconstruct the points in the plane that contains the reconstructions made with the two first principal components. Indicate the Matlab commands that you used to accomplish this. Plot the resulting points. Sketch the result, with scales in the axes.

R.

3 PCA for image compression

The file `faces.bmp` contains 24 grayscale images of faces, vertically stacked. You can view these images by directly viewing the file in Windows.

The data set considered in this section is formed by these images. Each image is of size 60 by 50 pixels, thus forming a vector of dimension 3000.

The program `loadimg.m` reads these data and generates a matrix `y` where each column represents an image. Since the data set contains 24 images, this matrix is 3000×24 in size.

The function `showimg` takes a matrix with 3000 rows, and displays the images corresponding to each column, side by side, as images of size 60×50 pixels. For example, `showimg(y(:,1))` shows the first image of the data set, and `showimg(y(:,1:3))` shows the first three images.[1]

3.1(E) Find the principal directions of this dataset (the computations may take some time because you are performing PCA in a 3000-dimensional space). Write down the five highest and the five lowest eigenvalues that you obtained.[2] If you find something strange in some of the values that you obtained, try to explain what happened.

R.

3.2(E) Use `showimg` to display the mean and the eigenvectors corresponding to the five largest eigenvalues. Comment on the appearance of those images.

R.

[1] Enter the command `colormap(gray)` to display the images in grayscale, instead of the default false-color palette.

[2] Use the `diag` function to ease the task of examining a very large diagonal matrix.

3.3(E) Make a small script to reconstruct the images using a number of principal components that is given as a parameter. Then reconstruct the images, progressively increasing the number of principal components, starting from 0. Comment on what you observe. Approximately how many components are needed for the photographed persons to be recognizable? And for the facial expressions (e.g. serious, smiling,...) to be recognizable?

R.

3.4(E) How many principal components would be needed to reconstruct the images without error, if there were no computational errors? Give a rigorous justification of your answer.

R.