IE 582 Statistical Learning for Data Mining
Homework 2, due October 25th, 2018

Instructions: Please solve the following exercises using R (http://www.r-project.org/). Also, you are requested to use certain R packages for this particular homework. You are expected to use GitHub Classroom and present your work as an html file (i.e. web page) on your progress journals (as you did for Homework 0).

Do not share your code (except the one in your progress journals)! As a fundamental principle for any educational institution, academic integrity is highly valued and seriously regarded at Boğaziçi University.

Tasks

Task 1

This task is to understand if we can obtain significant information regarding the game outcomes using the odd data from multiple bookmakers.

a) Select a least 5 bookmakers to check if over/under 2.5 game result can be explained by the odds for different types of bets. This can be achieved by training a classification model but suppose we would like to perform an analysis in an unsupervised way. For each game, you need to end up with a feature vector of the odds (i.e. home odd for bookmaker x, away odd for bookmaker x, tie odd for bookmaker x, over2.5 odd for bookmaker x, under 2.5 off for bookmaker x, both teams to score/YES for bookmaker x and etc.). Suppose we perform a principal component analysis (PCA) on this feature set to end up with a lower dimensional representation (i.e. 2D or 3D). Comment on your PCA results (i.e. eigenvectors, variance covered and etc.). Suppose we plot the new coordinates where the points are color-coded based on their under/over 2.5 status. Is there any interesting information? Comment on your findings.
b) Follow the similar strategy by applying multidimensional scaling (MDS). Since you are given a data matrix. You need to transform this information to distance matrix. Use Manhattan and Euclidean distance to perform the same task in part (a). What are your conclusions? What are the differences between the MDS on Euclidean distance and Manhattan distance?
c) Compare your PCA results with MDS.

Task 2

Repeat Part (a) of Task 1 for match outcomes (Home, Tie, Away).

Task 3

In this task, you are requested to perform certain operations on images. The aim is to compress images using PCA.

Here is a background information about how a grayscale image is represented on our computers. A grayscale image is basically a matrix where each matrix entry shows the intensity (brightness) level. In other words, when you take a picture with a digital camera, the image is represented by a numerical matrix where the matrix size is defined by the resolution setting of your camera. If your resolution setting is 1280x720, then your image is represented by 1280x720= 921600 pixel values (Actually that is why higher resolution provides better quality pictures).

When you have a color image, the image stores the information of multiple channels depending on the image type. The most famous one is RGB type where R, G and B stand for “red”, “green” and “blue” respectively. Hence, you have a matrix as in greyscale images representing the intensity for each channel. Combining these matrices generates the color image.

Below is the steps you need to follow for this exercise:

- Take a picture of yours (or some object of your choice) and save it as **.jpg* or **.jpeg* file.
- Resize the image to size 512x512 px (pixel) using an image editor (i.e. *Paint* in Windows). This image is the one that you will use for this exercise.

1- Read image as a variable in R. You need to install "jpeg" package to read image into a variable.

2- What is the structure of the variable that stores the image? What is the dimension?

a. Display the image. (Hint: google "rasterImage")

b. Display each channel using "image" function on a single plot. (Hint: google "multiple plots in r")

3- In order to create a noisy image, add a random noise from uniform distribution between 0 and 0.1 to each pixel value for each channel of original image.

a. Display the new image.

b. Display each channel separately using "image" function on a single plot.

4- Transform your noisy image to a greyscale one using either R or an image editor (both is fine). In signal processing, it is often desirable to be able to perform some kind of noise reduction on an image or signal (note that image can be considered as a 2D signal). Suppose we aim at reducing the size of the image with minimal loss (this reduction may or may not help in noise reduction). What we can do in order to perform such a reduction is to apply PCA to patches extracted from the images. How this is achieved is to visit each pixel by creating a window of certain size (which includes the neighboring entries). For 2D signals such as images, complex window patterns are possible (such as "box" or "cross" patterns) but think of a patch as a box around a certain pixel value as illustrated in Figure 1.

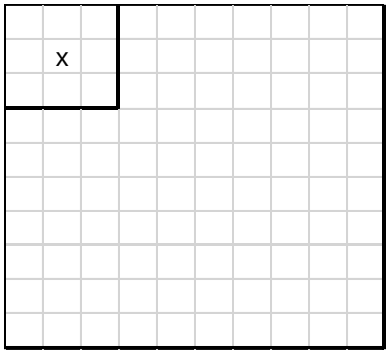

Figure 1. Sample patch from an image

Figure 1 is a sample image represented by 10 by 10 matrix. A 3 by 3 patch extracted for pixel at (2,2) is also shown. Suppose we extract all possible patches by sliding one pixel over the image and represent each patch as a vector of length 9. In other words, each patch becomes an instance. For this particular example you will have 8x8=64 patches.

a) Apply PCA to this data matrix and comment on PCA results.
b) Let's say we decide to use the first component to reconstruct the image (i.e. use the mapping on the first component as the pixel value for the patch). Plot the scores (i.e. mapping) for the first component as an image (recall that each patch is extracted from a certain location so that you can obtain an image from the scores). Do the same for the second and third components.
c) Components (eigenvectors) themselves refers to a patch. Plot the first component as 3 by 3 image. Do the same for the second and third components and comment.