Birla Institute of Technology and Science
Machine Learning
Lab Sheet #5

Learning Outcome:

a) Principal Component Analysis in R
b) Create visualizations to display data

Principal Component Analysis (PCA) is a useful technique for exploratory data analysis, allowing you to better visualize the variation present in a dataset with many variables. It is particularly helpful in the case of "wide" datasets, where you have many variables for each sample.

Introduction to PCA

PCA is particularly handy when you're working with "wide" data sets. In such cases, where many variables are present, you cannot easily plot the data in its raw format, making it difficult to get a sense of the trends present within. PCA allows you to see the overall "shape" of the data, identifying which samples are similar to one another and which are very different. This can enable us to identify groups of samples that are similar and work out which variables make one group different from another. The basics of PCA are as follows: you take a dataset with many variables, and you simplify that dataset by turning your original variables into a smaller number of "Principal Components". Principal Components are the underlying structure in the data. They are the directions where there is the most variance, the directions where the data is most spread out. This means that we try to find the straight line that best spreads the data out when it is projected along it. This is the first principal component, the straight line that shows the most substantial variance in the data. PCA is a type of linear transformation on a given data set that has values for a certain number of variables (coordinates) for a certain amount of spaces. This linear transformation fits this dataset to a new coordinate system in such a way that the most significant variance is found on the first coordinate, and each subsequent coordinate is orthogonal to the last and has a lesser variance. In this way, you transform a set of x correlated variables over y samples to a set of p uncorrelated principal components over the same samples. Where many variables correlate with one another, they will all contribute strongly to the same principal component. Each principal component sums up a certain percentage of the total variation in the dataset. Where your initial variables are strongly correlated with one another, you will be able to approximate most of the complexity in your dataset with just a few principal components. As you add more principal components, you summarize more and more of the original dataset. Adding additional components makes your estimate of the total dataset more accurate, but also more unwieldy.

Eigenvalues and Eigenvectors

Just like many things in life, eigenvectors, and eigenvalues come in pairs: every eigenvector has a corresponding eigenvalue. Simply put, an eigenvector is a direction, such as "vertical" or "45 degrees", while an eigenvalue is a number telling you how much variance there is in the data in that direction. The eigenvector with the highest eigenvalue is, therefore, the first principal component. The number of eigenvalues and eigenvectors that exits is equal to the number of dimensions the data set has. We can reframe a dataset in terms of these eigenvectors and eigenvalues without changing the underlying

information. Note that reframing a dataset regarding a set of eigenvalues and eigenvectors does not entail changing the data itself, you're just looking at it from a different angle, which should represent the data better.

A Simple PCA

Today you will try a PCA using a simple and easy to understand dataset. You will use the mtcars dataset, which is built into R. This dataset consists of data on 32 models of car, taken from an American motoring magazine (1974 Motor Trend magazine). For each car, you have 11 features, expressed in varying units. Familiarize yourself with the mtcars dataset using summary, head, and str functions.

Compute the Principal Components

Because PCA works best with numerical data, you'll exclude the two categorical variables from mtcars (vs and am). You are left with a matrix of 9 columns and 32 rows, which you pass to the prcomp() function, assigning your output to mtcars.pca. You will also set two arguments, center and scale, to be TRUE. Then you can have a peek at your PCA object with summary(). Type the following in R:

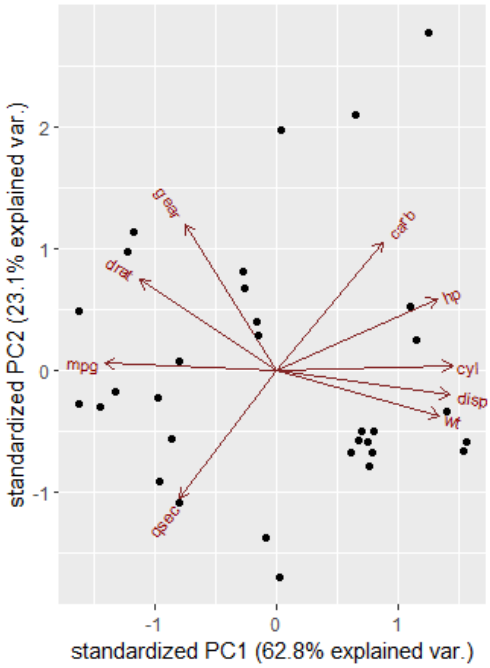
> mtcars.pca <- prcomp(mtcars[,c(1:7,10,11)], center = TRUE,scale. = TRUE)

> summary (mtcars.pca)

Summary command will give you Standard deviation, Proportion of Variance, and Cumulative Proportion. Observe the output to observe the principal components. Observe that by knowing the position of a sample in relation to just PC1 and PC2, you can get a very accurate view on where it stands in relation to other samples, as just PC1 and PC2 can explain 86% of the variance.

Now call str() function on mtcars.pca and observe the result.

You should see the following type of graph:



Exercise:

1. Primarily, in R there are two general methods to perform PCA without any missing values: (1) **spectral decomposition** (R-mode [also known as eigendecomposition]) and (2) **singular value decomposition** (Q-mode; R Development Core Team 2011). Both of these methods can be performed longhand using the functions eigen (R-mode) and svd (Q-mode), respectively, or can be performed using the many PCA functions found in the stats package and other additional available packages. The spectral decomposition method of analysis examines the covariances and correlations between variables, whereas the singular value decomposition method looks at the covariances and correlations among the samples. While both methods can easily be performed within R, the singular value decomposition method (i.e., Q-mode) is the preferred analysis for numerical accuracy (R Development Core Team 2011). Full article about the same can be downloaded from following URL: https://www.ime.usp.br/~pavan/pdf/MAE0330-PCA-R-2013
2. Read the above article and perform PCM using princomp on mtcars dataset. Compare the two.