PCA

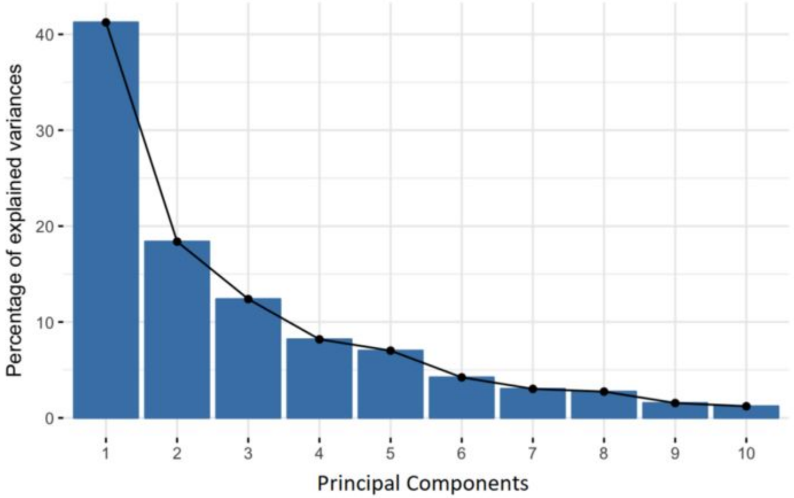
Principal component Analysis

What is PCA

- Is a dimensionality reduction method that is often used to reduce the dimensionality of large data sets
- By transforming a large set of variables into a smaller one that still contains most of the information in the large set.
- Makes analyzing data points much easier and faster for machine learning algorithm
- **PCA reduces the number of variables of a data set, while preserving as much information as possible.**

What is a principal component?

- Principal components are new variables that are constructed as linear combinations of the initial variables.
- These new variables (i.e., principal components) are uncorrelated and most of the information within the initial variables is compressed into the first components.
- So, the idea is 10-dimensional data gives you 10 principal components, but PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on



HOW DO YOU DO A PRINCIPAL COMPONENT ANALYSIS?

1. Standardize the range of continuous initial variables
2. Compute the covariance matrix to identify correlations
3. Compute the eigenvectors and eigenvalues of the covariance matrix to identify the principal components
4. Create a feature vector to decide which principal components to keep
5. Recast (re-position) the data along the principal components axes

STEP 1: STANDARDIZATION

The aim of this step is to standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis.

$$z = \frac{value - mean}{standard\ deviation}$$

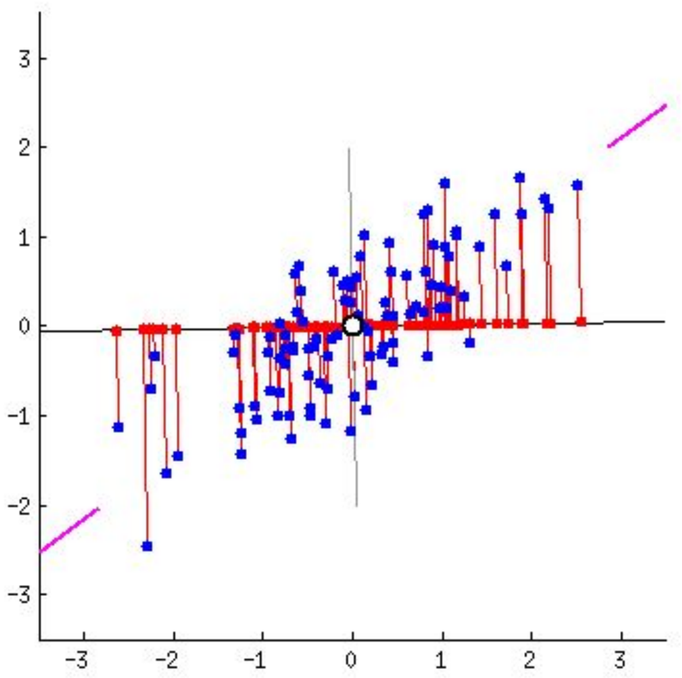
STEP 2: COVARIANCE MATRIX COMPUTATION

- The aim of this step is to understand how the variables of the input data set are varying from the mean with respect to each other (any relationship between them)
- sometimes, variables are highly correlated in such a way that they contain redundant information.
- In order to identify these correlations, we compute the covariance matrix.
- The covariance matrix is a $p \times p$ symmetric matrix, p is the number of features, for a 3-dimensional data set with 3 variables x, y, and z, the covariance matrix is a 3×3 data matrix of this from
- covariance of a variable with itself is its variance, covariance is commutative (Cov(a,b)=Cov(b,a))

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

STEP 3: COMPUTE THE EIGENVECTORS AND EIGENVALUES OF THE COVARIANCE MATRIX TO IDENTIFY THE PRINCIPAL COMPONENTS

- the eigenvectors of the Covariance matrix are actually *the directions of the axes where there is the most variance* (most information) and that we call Principal Components.
- eigenvalues are the coefficients attached to eigenvectors, which give the *amount of **variance** carried in each Principal Component*.
- By ranking your eigenvectors in order of their eigenvalues, highest to lowest, you get the principal components in order of significance.



STEP 4: FEATURE VECTOR

- The feature vector is simply a matrix that has as columns the eigenvectors of the components that we decide to keep.
- This makes it the first step towards dimensionality reduction, because if we choose to keep only ***p*** eigenvectors (components) out of ***n***, the final data set will have only ***p*** dimensions.
- Given this as the resulted eigen vectors and eigenvalues, discard the eigenvector $v2$
-

$$v1 = \begin{bmatrix} 0.6778736 \\ 0.7351785 \end{bmatrix} \qquad \lambda_1 = 1.284028$$

$$v2 = \begin{bmatrix} -0.7351785 \\ 0.6778736 \end{bmatrix} \qquad \lambda_2 = 0.04908323$$

$$\begin{bmatrix} 0.6778736 & -0.7351785 \\ 0.7351785 & 0.6778736 \end{bmatrix}$$

$$\begin{bmatrix} 0.6778736 \\ 0.7351785 \end{bmatrix}$$

STEP 5: RECAST THE DATA ALONG THE PRINCIPAL COMPONENTS AXES

- Using the eigenvectors of the covariance matrix, to reorient the data from the original axes to the ones represented by the principal components
- This can be done by multiplying the transpose of the original data set by the transpose of the feature vector.

$$FinalDataSet = FeatureVector^T * StandardizedOriginalDataSet^T$$

Assignment

- Compress a gray scale image using PCA, then visualize the compresses image
- Decompress the compressed image and visualize it
- Don’t use the PCA built-in function
- Group of 2
- Bonus:
    - Compress a group of images that can be grouped into less number of images containing the common features (hint: search for eigen faces)
    - Compress a colored image instead of a gray image