Introduction

Principal Components Analysis

Principal Components Analysis (PCA) is a dimensionality reduction technique that reduces the dimensionality of the data while preserving the variation present in the dataset as much as possible.

The principal components (directions or axes) we want to find are the eigenvectors of the data. Given an eigenvector, if we apply any transformation to it, then the resulting will always be the eigenvector multiplied by a scalar which is known as eigenvalue.

Principal Component Analysis is an unsupervised learning algorithm.

We will find the set of necessary feature dimensions for describing the data
$f : R^D \dashrightarrow R^d$ where $d << D$ while minimizing the information loss by applying PCA.

Dimensionality Reduction using PCA

In high-dimensional problems, data sometimes lies near a linear subspace
We want to maintain the variance in our data so we choose the principal components with largest eigenvalue. When eigenvalues are small, this means the information along the component is basically noise. Since we select the top-k principal components, we might lose some information but if eigenvectors whose eigenvalues are small enough, then we won't lose too much information. Thus, with almost no information we can reduce data size and with less complexity we can compress and decompress data again.

Thus, PCA plays important role in reducing size without much information loss.

Eigenfaces

Eigenfaces are obtained using PCA algorithm

For, dataset X,

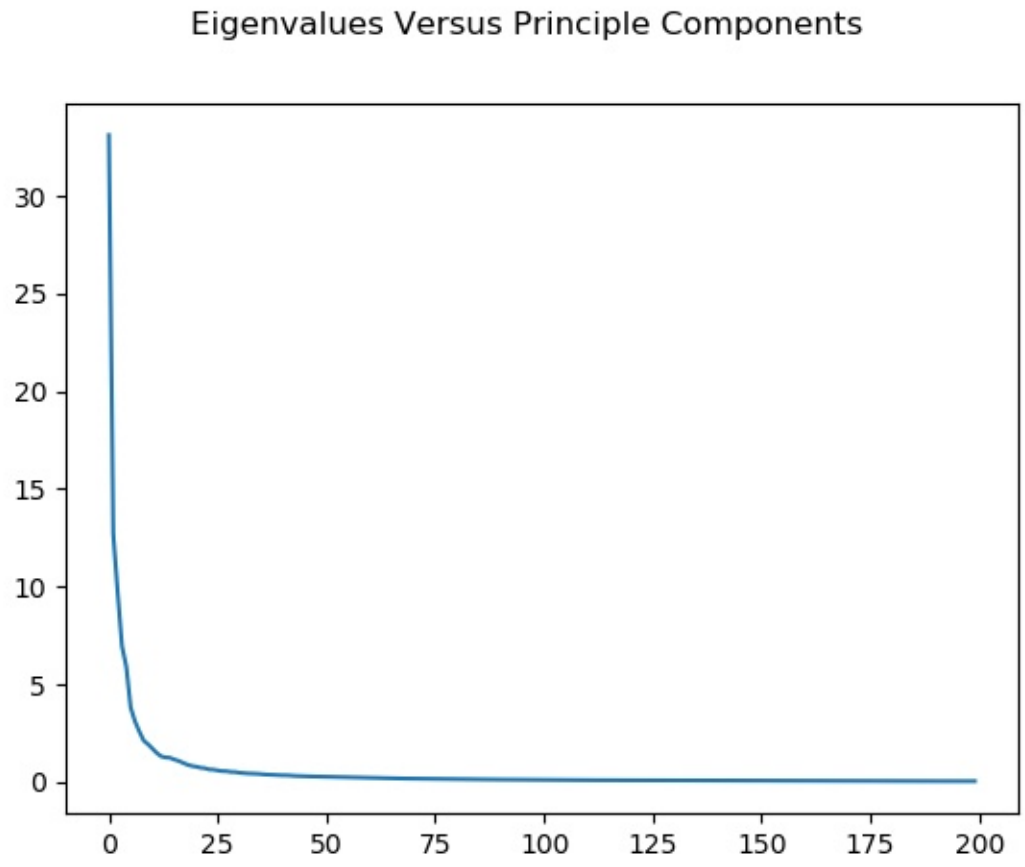
1) Subtract the mean B = X - μ where μ = $\frac{1}{N}\sum_{n}^{N} X^{n}$

2) Compute the covariance matrix $C = \frac{1}{N-1} B^{T} B$

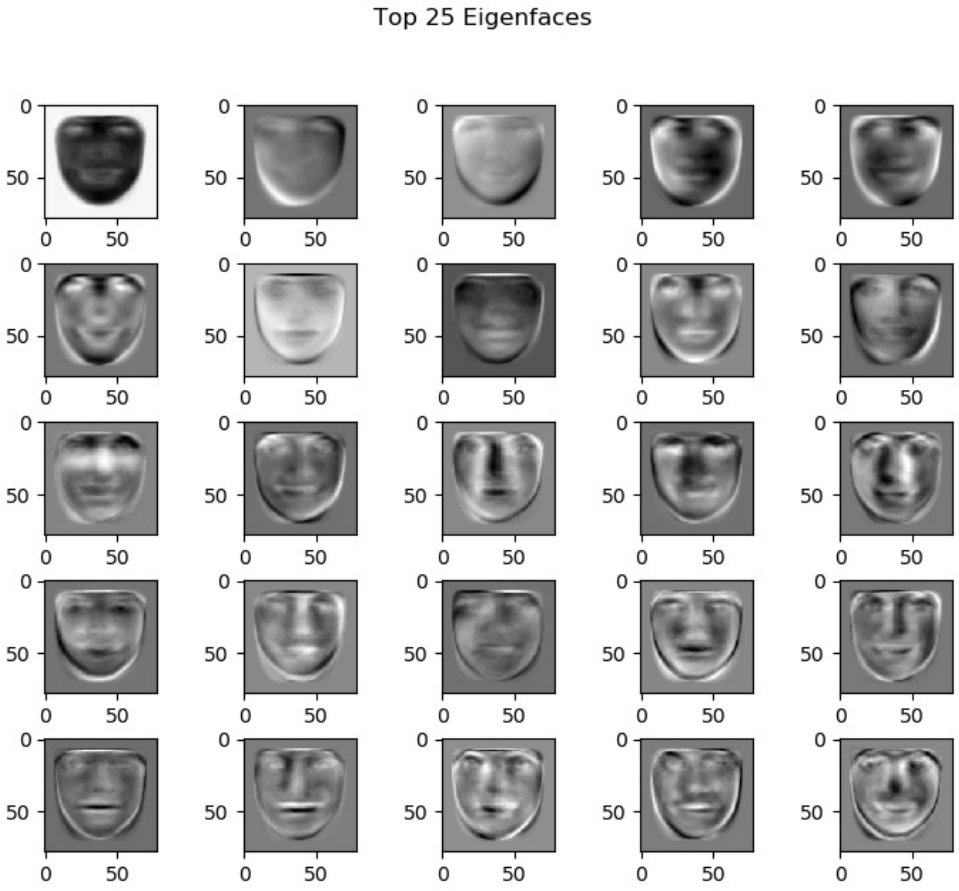
3) Compute the eigenvectors $V^{-1}CV = \sum$ where V is the eigenvectors and the $\sum$ is the eigenvalues.

4) Sort V based on $\sum$ from largest to smallest, and choose the top k and form W.

Below is graph of obtained eigenvalues vs. principle components.



Top 25 Eigenfaces obtained via PCA on given dataset



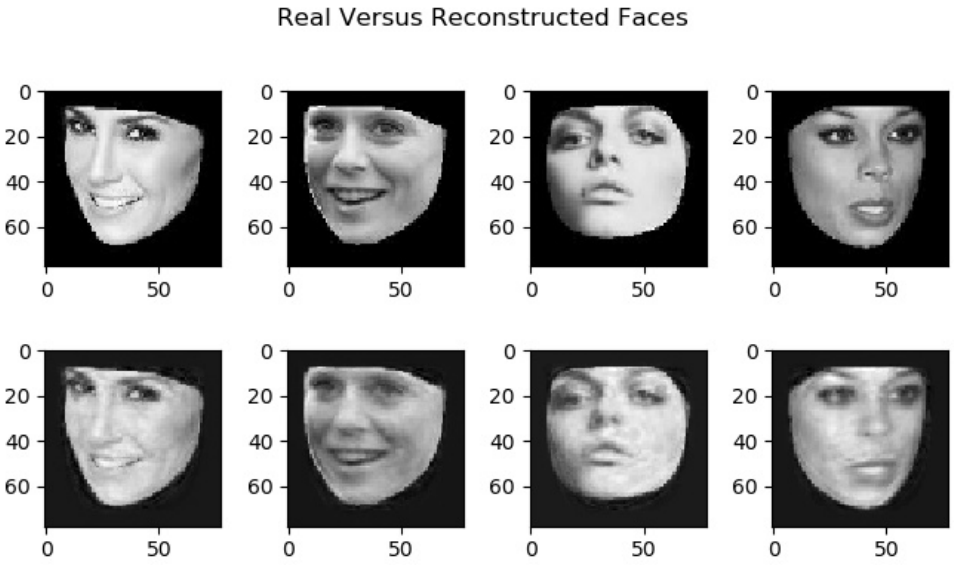
Reconstructions

First,
Project X to the new feature space (a subspace) $Z = BW$ (latent vector).

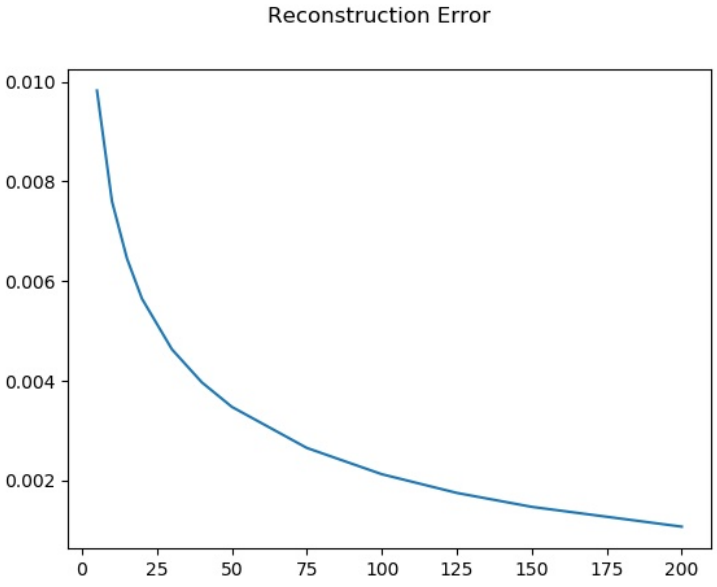
Now to get data back we will reconstruct it from Z by,

$$\hat{X} = ZW^T + \mu$$

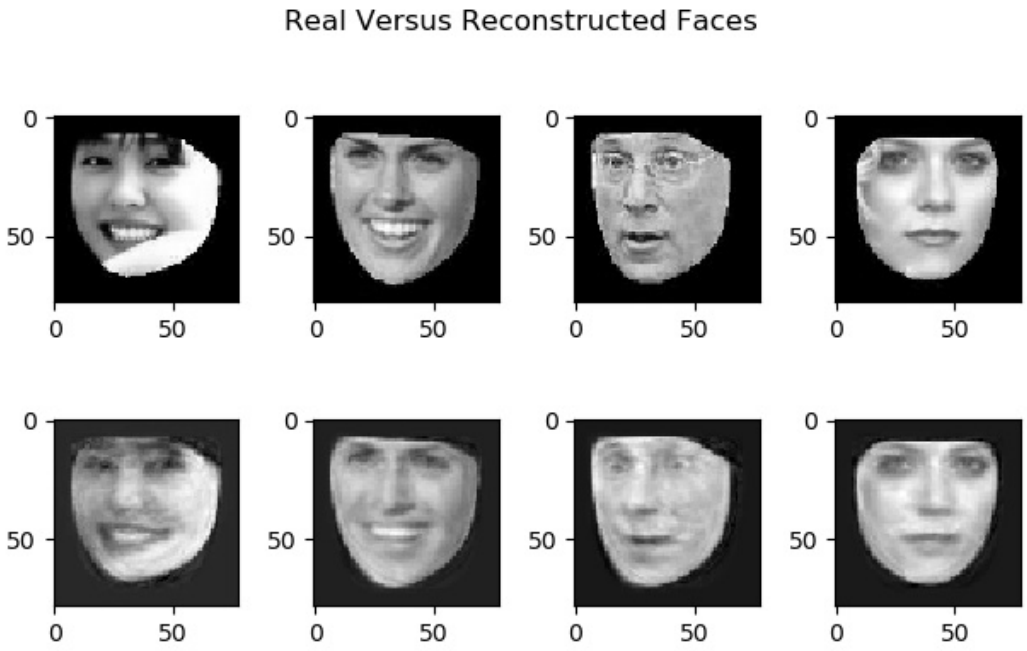
These were the back projected faces of the training dataset using k = 200,



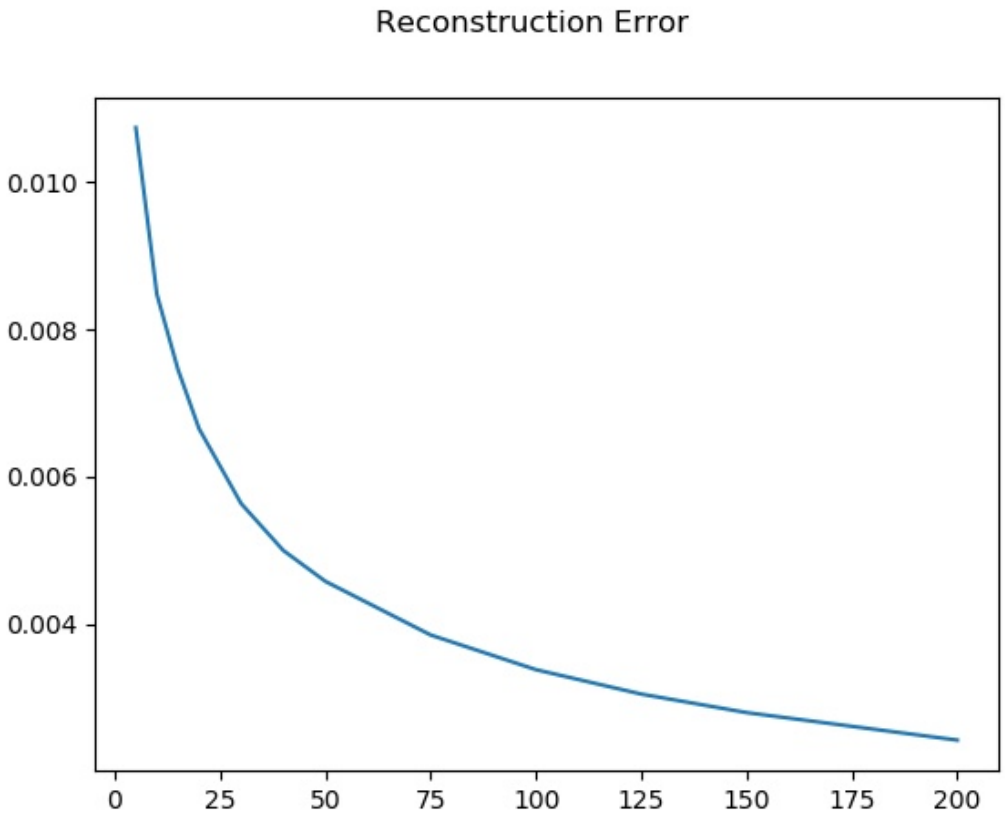
To check the reconstruction error, mean square error (MSE) is used,
K vs. MSE graph obtained of training set with different values of K,



Using same top 200 eigenfaces and checking them on testing data following faces were obtained,



K vs. MSE graph of testing data set with different values of K,



Synthetic Faces

Earlier we used Z = BW as latent vector, instead we will use Gaussian Distribution to generate new faces from eigenvalues $Z \sim \mathcal{N}(0, \sigma^2)$.
Using $\sum$ (eigenvalues) as variance in distribution.
So, we will reconstruct data by, $\hat{X} = ZW^T + \mu$

Some of synthetic faces looks like using K = 35,