Face Recognition using Principal Component Analysis

Ravitha N

Face Recognition

- **Face recognition** is the challenge of classifying whose face is in an input image.
- A naïve way of accomplishing this is to take the new image, flatten it into a vector, and compute the Euclidean distance between it and all of the other flattened images in our database. Obvious Drawback is speed

Say you have a image of size 8*8 ~ 64 *1

If it needs to be compared with 200 images in the database, then the approach is unrealistic

Number of computations = 200 * 64 ~ 12,800 computations

Problems with Distance classifier

- It cannot capture inherent noise in image (translation, rotation, illumination variations) in an image

Can we improve classifier?

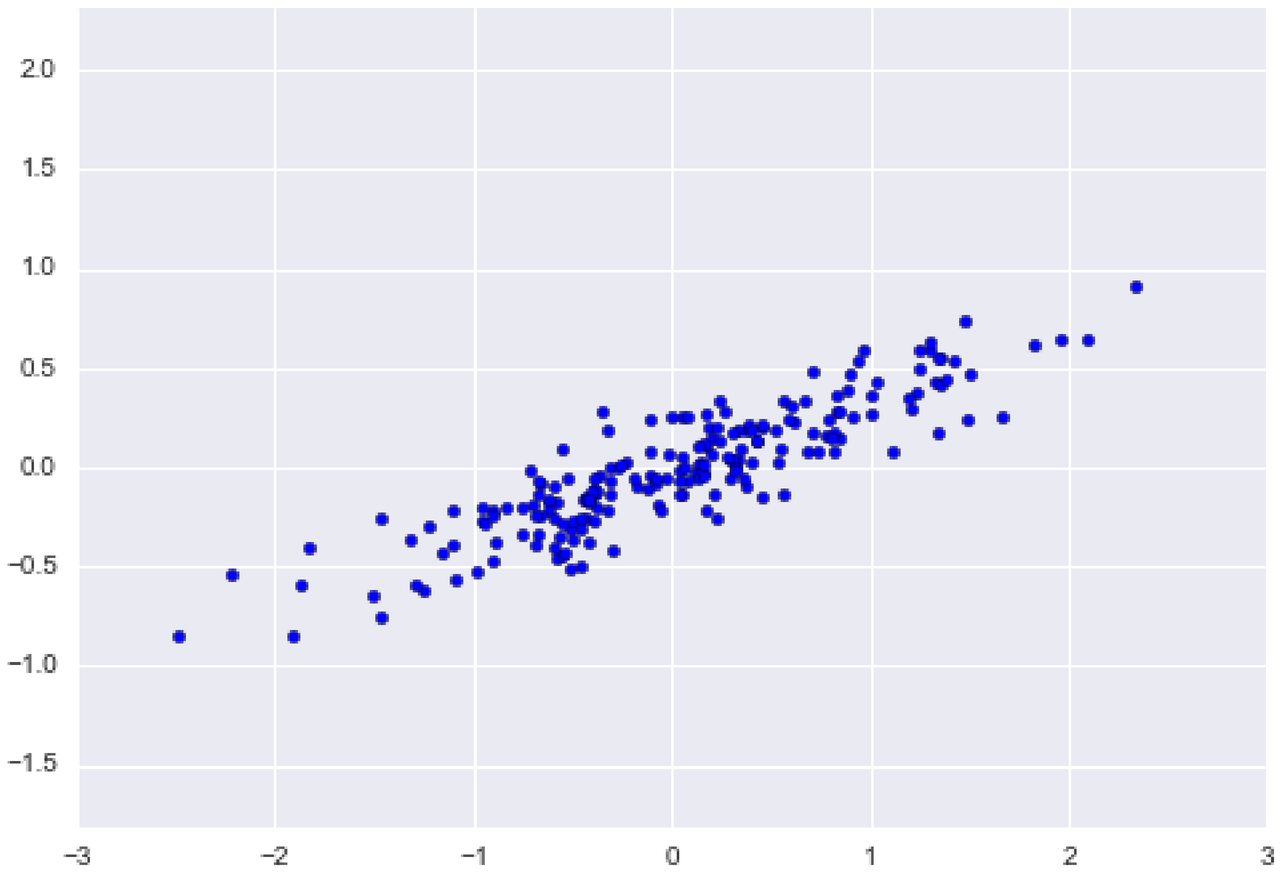
- Advanced Classifiers such as Neural Networks can be used. But still the network would suffer from curse of dimensionality

if the image is flattened into mn * 1 vector; then the network requires more than mn images which is not realistic for images (as it would depend on the storage space, computational complexity)

Principal Component Analysis

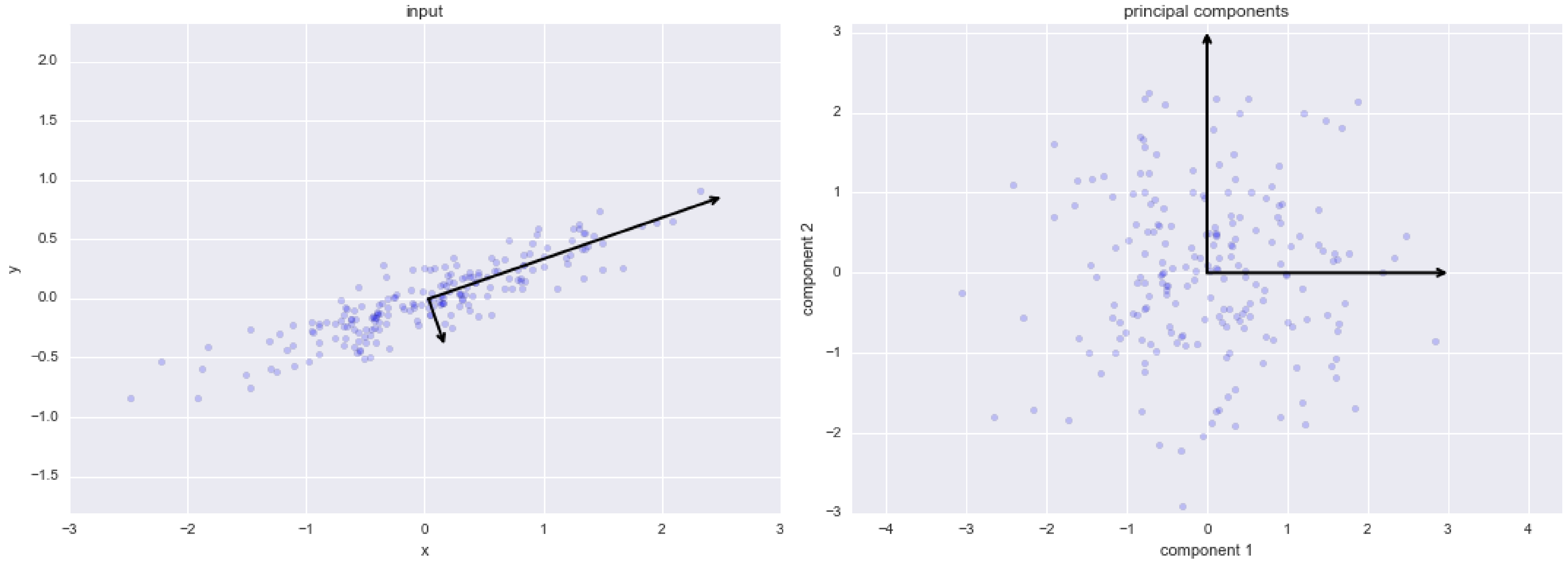
- Reduce the size of the image (not blindly)
- Can we have a technique which can map the face to smaller dimension but retains most of the important parts in it
- Principal Component Analysis is one such technique which helps in mapping the datapoints to lower dimensions.

Sample Data and its Principal Components



Contd..

Observation : In the First Figure, Principal Vector captures most of the variance in the data.



Projection of Data point to a vector

- Data Point [3,2]
- Vector [4 5]

Perpendicular Projection of data point onto the vector

Dot Product between the vectors will help in projection of point onto a vector.

Transformation of data points onto the Principal Vector

Mathematical Approach

- Consider an image of size 4 * 4. They can be represented as row vector

$$x = \{x_1, x_2, x_3, x_4 \cdot \cdots x_{16}\}$$

- Observation matrix is constructed, where images are encoded as columnar vectors

$$\begin{bmatrix} x_{11} & \cdots & x_{1,16} \\ \vdots & \ddots & \vdots \\ x_{1,16} & \cdots & x_{n,16} \end{bmatrix}$$

- Mean subtract the data (across the rows)

Contd..

- Compute the covariance matrix

$$C = \frac{1}{M-1}(x-\bar{x})(x-\bar{x})^T$$

Where M represents the number of samples (images)

$\bar{x}$ indicates the mean vector

- Compute eigen values and eigen vectors for the matrix (Eigen vectors are special vectors which when incident on a matrix , it never modify the orientation)

$Cv = \lambda v$ which implies

$$C - \lambda = 0$$

Contd..

- Solving the equation we will obtain the unknown quantity λ(eigen value) and v (eigen vector)

Find the principal component for the following data points

- $x_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} x_2 = \begin{bmatrix} 3 \\ 4 \end{bmatrix} x_3 = \begin{bmatrix} 5 \\ 0 \end{bmatrix} x_4 = \begin{bmatrix} 7 \\ 6 \end{bmatrix} x_5 = \begin{bmatrix} 9 \\ 2 \end{bmatrix}$

Observation Matrix

$$x = \begin{bmatrix} 2 & 3 & 5 & 7 & 9 \\ 1 & 4 & 0 & 6 & 2 \end{bmatrix}$$

Mean Vector

$$\bar{x} = \begin{bmatrix} (2+3+5+7+9)/5 \\ (1+4+0+6+2)/5 \end{bmatrix} = \begin{bmatrix} 5.2 \\ 2.6 \end{bmatrix}$$

Mean Subtracted data

$$x - \bar{x} = \begin{bmatrix} -3.2 - 2.2 & -0.2 & 1.8 & 3.8 \\ -1.61 \cdot 4 & -2.6 & 3.4 & -0.6 \end{bmatrix}$$

Covariance Matrix

$$C = \frac{1}{4}\begin{bmatrix} 3.2 & -2.2 & -0.2 & 1.8 & 3.8 \\ -1.6 & 1.4 & -2.6 & 3.4 - 0.6 \end{bmatrix}\begin{bmatrix} 3.2 & -1.6 \\ -2.2 & 1.4 \\ -0.2 & -2 \cdot 6 \\ 1.8 & 3 \cdot 9 \\ 3.8 & -0 \cdot 6 \end{bmatrix}$$

$$C = \frac{1}{4}\begin{bmatrix} 32.8 & 6 \cdot 4 \\ 6.4 & 23.2 \end{bmatrix}$$

$$C = \begin{bmatrix} 8.2 & 1.6 \\ 1.6 & 5.8 \end{bmatrix}$$

Finding Eigen Vectors

$$\begin{vmatrix} 8.2 & 1.6 \\ 1.6 & 5.8 \end{vmatrix} - \begin{vmatrix} \lambda & 0 \\ 0 & \lambda \end{vmatrix} = 0$$

$$\begin{vmatrix} 8.2 - \lambda & 1.6 \\ 1.6 & 5.8 - \lambda \end{vmatrix} = 0$$

$$\lambda^2 - 14\lambda + 45 = 0$$

$$(\lambda - 9)(\lambda - 5) = 0$$

$$\lambda = 9, \lambda = 5$$

Eigen vector for lambda 9
Substitute it in

$$\begin{vmatrix} 8.2 - \lambda & 1.6 \\ 1.6 & 5.8 - \lambda \end{vmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

$$\begin{vmatrix} 8.2 - 9 & 1.6 \\ 1.6 & 5.8 - 9 \end{vmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

Assume x2=1

$$-0.8x_1 + 1.6x_2 = 0$$
$$1.6x_1 + 3.2x_2 = 0$$

-0.8x1 = -1.6 => x1=2

$$\begin{bmatrix} x1 \\ x2 \end{bmatrix} = \frac{1}{\sqrt{2^2 + 1^2}} \begin{bmatrix} 2 \\ 1 \end{bmatrix} \simeq \begin{pmatrix} 0.89 \\ 0.45 \end{pmatrix}$$

- Eigen vector for lambda 5
- Substitute it in

$$\begin{vmatrix} 8.2-\lambda & 1.6 \\ 1.6 & 5.8-\lambda \end{vmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

$$\begin{vmatrix} 8.2-5 & 1.6 \\ 1.6 & 5.8-5 \end{vmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

Assume x2=1

$$3.2x_1 + 1.6x_2 = 0$$
$$1.6x_1 + 0.8x_2 = 0$$

3.2x1 = -1.6 => x1=-0.5

$$\begin{bmatrix} x1 \\ x2 \end{bmatrix} = \frac{1}{\sqrt{-0.5^2+1^2}} \begin{bmatrix} -0.5 \\ 1 \end{bmatrix} \simeq \begin{bmatrix} -0.45 \\ 0.89 \end{bmatrix}$$

Projecting Data points onto Principal Components

- Eigen Vectors v1 = $\binom{0.89}{0.45}$

V2 = $\begin{bmatrix} -0.45 \\ 0.89 \end{bmatrix}$

Convert data point (2 1) its mean centered value (-3.2 -1.6)

Project (-3.2 -1.6) $\binom{0.89}{0.45}$ = -3.568

Onto v1

Project (-3.2 -1.6) $\begin{bmatrix} -0.45 \\ 0.89 \end{bmatrix}$ = 0.016

Onto v2

- https://pythonmachinelearning.pro/face-recognition-with-eigenfaces/