Face Recognition - Eigenface Method

Vashie Garan

April 2021

1 Dataset

- Each of the image can be represented as an nxn matrix

$$X = \begin{bmatrix} X_{11} & X_{12} & . & . & . & X_{1n} \\ X_{21} & X_{12} & . & . & . & X_{1n} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ X_{n1} & X_{n2} & . & . & . & X_{nn} \end{bmatrix} \tag{1}$$

Matrix X can be represented as n^2- vector as well where each columns are stacked below

$$X = \begin{bmatrix} X_{11} \\ X_{12} \\ . \\ . \\ X_{21} \\ X_{22} \\ . \\ . \\ X_{nn} \end{bmatrix} \tag{2}$$

To make easier for us to understand on further notes we will denote our matrix X as

$$X = \begin{bmatrix} X_1 \\ X_2 \\ . \\ . \\ . \\ . \\ X_{n^2} \end{bmatrix} \tag{3}$$

- To represent m numbers of images, we can represent n^2xm matrix A as:

$$A = \begin{bmatrix} X_1^{(1)} & X_2^{(2)} & . & . & . & X_1^{(m)} \\ X_2^{(1)} & X_2^{(2)} & . & . & . & X_2^{(m)} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ X_{n^2}^{(1)} & X_{n^2}^{(2)} & . & . & . & X_{n^2}^{(m)} \end{bmatrix} \tag{4}$$

2 Problem Statement

Normally we would compare the distance between a given face image, $\bar{x}$ with each images, $x^{(i)}$ in the data set.

$$d_i = ||\bar{x} - x^{(i)}$$

Whenever the value of a d_i is smaller than a chosen threshold, the given image can be identified as the person that is associated with the i-th face image.

However, the vector that represent the images usually have large dimensionality. hence this will consume a amount of time and computationally expensive.

We need to find a lower-dimensional subspace to describe the image space.

3 Eigenfaces

3.1 Principal Component Analysis

Principal Component Analysis(PCA) is an algorithm that reduces the dimensionality of a data. The algorithm projects the data set into a lower dimensional linear subspace with minimum reconstruction error.

3.2 PCA and Eigenfaces

PCA finds the vectors that could capture the best information of face images within the entire image space. In other words, it could capture the variation in a collection of face images.

Hence, we would like to find the eigenvectors of the covariance matrix of the set of face images. This eigenvectors are the vectors that have the largest data distribution, in other words it has the highest information on the data. When the eigenvector is viewed as images, it will look like a ghostly face that we call eigenface.

3.3 Calculating Eigenfaces

A set of face images is defined as $x^{(1)}, x^{(2)}, x^{(3)}, ..., x^{(m)}$. The mean face image μ of the set is define by

$$\mu = \frac{1}{m} \sum_{i=1}^{m} x^{(i)}$$

Each of the face images differ from the mean by the vector $\bar{x}$ define by

$$\bar{x}^{(i)} = x^{(i)} - \mu$$

All of the $\bar{x}$ can be represented in a n^2xm matrix as:

$$\bar{A} = [\bar{x}^{(1)}, \bar{x}^{(2)}, \bar{x}^{(3)}, ., ., ., \bar{x}^{(m)}]$$

The n^2xn^2 covariance matrix of $\bar{A}$, C is define by:

$$C = \frac{1}{m} \bar{A}\bar{A}^T$$

Matrix C is computationally heavy to compute since Matrix C is a huge Matrix which is $n^2 \times n^2$ matrix . Fortunately, the $m \times m$ symmetrix matrix $\frac{1}{m}\bar{A}\bar{A}^T$ has m number of eigenvalues that are also eigenvalues of C.

$$\frac{1}{m}\bar{A}^T\bar{A}\nu_i = \lambda\nu_i$$

$$\bar{A}(\frac{1}{m}A^T A\nu_i) = \bar{A}(\lambda\nu_i)$$

$$\frac{1}{m}\bar{A}\bar{A}^T\bar{A}\nu_i = \bar{A}\lambda\nu_i$$

$$C\bar{A}\nu_i = \bar{A}\lambda\nu_i$$

From above, we can conclude that both $\frac{1}{m}\bar{A}\bar{A}^T$ and $\frac{1}{m}\bar{A}^T\bar{A}$ have the same eigenvalues.

Therefore, we construct the $m \times m$ symmetric matrix L = $\frac{1}{m}\bar{A}^T\bar{A}$ and find its eigenvectors. The associated eigenvalues can be used to rank the eigenvectors in descending order.

3.4 Face Detection and Recognition

The m eigenvectors calculated from L spans a basis set that descibes the face images. The best q eigenvectors such that q¡m is selected. The q eigenvectors is now the new image space.

Our new image space Matrix I is represented by:

$$I = \sum_{i=1}^{q} s_i^T s_i$$

where, $s_i = \frac{\bar{A}\nu_i}{||\bar{A}\nu_i||}$

A new image $\bar{x}(n^2 - ector)$ is first subtracted by mean vector from the set of images $\bar{x}$. The vector $\bar{x} - \mu$ is then projected to the new image space, S.

$$\bar{x}_{proj} = \sum_{i=1}^{q} I(\bar{x} - \mu)$$
$$\bar{x}_{proj} = \sum_{i=1}^{q} s_i^T(\bar{x} - \mu)s_i$$

The vector $\bar{x}_{proj}$ is the new face images represented by the lienar combination of the eigenfaces. The more the eigenfaces used for the reconstruction, the more the reconstructed face image looks like the original new face image.

The vector $\bar{x} - \mu$ can also be represented as a q-vector with respect to the eigenfaces, E.

$$E = \begin{bmatrix} w_1 \\ w_2 \\ . \\ . \\ w_q \end{bmatrix} \tag{5}$$

where ,

$$w_i = s_i^T(\bar{x} - \mu) \text{ for i= } 1, 2, 3, ..., q$$

3.5 Face Detection

To determine whether the given image $\bar{x}$ is a face image, the distance of $\bar{x}$ from the face space can be used. The distance of $\bar{x}$ from the face space B is:

$$B = \sqrt{||\bar{x} - \bar{x}_{proj}||}$$

if B is smaller than chosen threshold, t, then $\bar{x}$ can be identified as a face image

3.6 Face Recognition

To determine which face class provides the best description of the new imaage, we need to find the face class z that minimized the Euclidean distance.

$Class_z = \sqrt{||(E - E^{(z))}||}$ for $z = 1, 2, 3, ..., m$

Where $E^{(z)}$ is the vector of $\bar{x}^{(z)}$ represented with respect to the eigenfaces:

$$E^{(z)} = \begin{bmatrix} w_1^{(z)} \\ w_2^{(z)} \\ . \\ . \\ w_q^{(z)} \end{bmatrix} \tag{6}$$

where,
$w_i^{(z)} = s_i^T(\bar{x}^{(z)} - \mu)$ for i$= 1, 2, 3, ..., q$

IF the smallest $Class_z$ is lower than a chosen threshold, t2, the new face image $\bar{x}$ is classified as face z