Principal Component Analysis

Introduction

The goal of PCA is dimensionality reduction. Dimensionality reduction is achieved through the formation of basis vectors. Using basis vectors any sample from a data set can be recreated using a linear combination of basis vectors.

For example in 3 dimensions the basis vectors are:

$$\begin{bmatrix}1\\0\\0\end{bmatrix},\quad \begin{bmatrix}0\\1\\0\end{bmatrix},\quad \begin{bmatrix}0\\0\\1\end{bmatrix}$$

So if we are given the sample vector: $\begin{bmatrix}2\\3\\5\end{bmatrix}$

We can reconstruct it using a linear combination of the basis vectors: $\begin{bmatrix}2\\3\\5\end{bmatrix} = 2 * \begin{bmatrix}1\\0\\0\end{bmatrix} + 3 * \begin{bmatrix}0\\1\\0\end{bmatrix} + 5 * \begin{bmatrix}0\\0\\1\end{bmatrix}$

The basis vectors must be orthogonal to one another. This means that the dot product of any two basis vectors is 0. For example in the 3 dimensional case:

$$\begin{bmatrix}1\\0\\0\end{bmatrix} \cdot \begin{bmatrix}0\\1\\0\end{bmatrix} = 1 * 0 + 0 * 1 + 0 * 0 = 0$$

Naturally the next question becomes how can we find the basis vectors? This will be explained mathematically in the next section.

Principal Component Analysis Algorithm Steps

1. Find the mean vector.
2. Assemble all the data samples in a mean adjusted matrix.
3. Create the covariance matrix.
4. Compute the Eigen vectors and Eigen values.
5. Compute the basis vectors.
6. Represent each sample as a linear combination of basis vectors.

To begin, assume that we have a set of images we would like to perform Principal Component Analysis (PCA) on. Assume that each image is x pixels by y pixels. We can treat each image as a vector of size x*y without loss of information. In this way, we can consider an image as a point in x*y dimensional space.

$$\begin{bmatrix}1 & 4 & 7\\ 2 & 5 & 8\\ 3 & 6 & 9\end{bmatrix} \dashrightarrow \begin{bmatrix}1\\4\\7\\2\\5\\8\\3\\6\\9\end{bmatrix}$$

3x3 Image

9x1 vector representation of image

Since images are typically hundreds of pixels in x and hundreds of pixel in y dimensions, an image becomes a point in a very high dimensional space. Using PCA we can reduce this dimensionality significantly. One of the reasons for reducing the dimensions is so that we can focus on those dimensions where there is a high difference between images in our dataset i.e., high variance. PCA allows us to mathematically reduce the dimensions so that high variance dimensions (for a given dataset) are selected in the reduced set of dimensions.

To understand this further, assume we have 500 images of faces in our dataset, each of which are 100 pixels by 100 pixels. Essentially, this means each image of a face is represented by 10,000 numbers (dimensions). Using PCA we can have each face be represented by only 10 numbers (or whatever the number of reduced dimensions we want).

The steps in computing the PCA are:

1. Find the mean vector:

Assuming we have N samples we can compute the mean vector as:

$$\bar{S} = (S_1 + S_2 + \cdots + S_N)/N$$

Note in our example $N = 500$ and the dimensions of $\bar{S}$ will be:

$\bar{S}$ Dimensions will be (10,000x1)

2. Assemble the mean adjusted matrix:

For every image vector i the mean adjusted vector can be computed as:

$$\bar{S}_i = (S_i - \bar{S})$$

We put every mean adjusted vector together to form the mean adjusted matrix:

$$S_{mean} = [\bar{S}_1 \quad \dots \quad \bar{S}_N]$$

S_{mean} Dimensions will be(10,000x500)

3. Compute the covariance matrix:

$$C = \begin{bmatrix} Cov_{0,0} & Cov_{0,1} & \dots & Cov_{0,n} \\ \vdots & \dots & \dots & \vdots \\ Cov_{n,0} & \dots & \dots & Cov_{n,n} \end{bmatrix}$$

C Dimensions will be (500x500)

Where:

$Cov_{i,j=}(S_i - \bar{S}) \cdot (S_j - \bar{S})$ ($\cdot$ indicates *matrix dot product*)

$\bar{S} = The\ mean\ vector\ computed\ in\ step\ 1$

$S_i = The\ ith\ image\ vector$

$S_j = The\ jth\ image\ vector$

Note that the covariance matrix represents the covariance measurement of each sample in the data set with every other sample. In general, the covariance value is computed between two random variables and represents the variables relationship with one another. If the covariance is positive, then the two variables increase together and decrease together. If the covariance is negative the variables are inversely proportional. Basically the covariance matrix attempts to display the relationships between each pair of samples in the dataset.

4. Compute the Eigen vectors and Eigen values of the covariance matrix.

The Eigen values λ, of the covariance matrix can be solved for using the following equation:

$$\det(\lambda \mathrm{I} - \mathrm{C}) = 0$$

Where:

$det = The\ determinant\ of\ the\ matrix$

$\lambda = The\ eigen\ values\ associated\ with\ the\ matrix$

$I = The\ identity\ matrix$

$C = The\ covariance\ matrix$

Solving the above equation for a 500x500 covariance matrix will result in 500 possible values for λ (Eigen values), $\lambda_1, \ldots, \lambda_{500}$.

After computing the Eigen values, sort the Eigen values (λ) by magnitude. Keep the highest *n* Eigen values and discard the rest (where *n* is number of desired reduced dimensions). In our example, we want to reduce down to 10 dimensions so after sorting the 500 eigen values we computed, we will only keep the highest 10.

Why does the magnitude of the Eigen values matter? The Eigen values will be used to form the basis vectors. The larger the magnitude of the Eigen value, the more variance its corresponding basis vector will contain. Since we are reducing dimensionality we are inevitably losing some information, however by picking basis vectors based on Eigen values with the highest variance, we are preserving as much of the original information as possible.

From these 10 Eigen values, we can compute an Eigen vector associated with each Eigen value. For a given Eigen value λ_k, the Eigen vector V_k can be computed as follows:

$$(\lambda_k I - C) * V_k = 0$$

Where:

$\lambda_k = One\ of\ the\ highest\ eigen\ values\ kept$

$I = The\ identity\ matrix$

$C = The\ covariance\ matrix$
$V_k = The\ eigen\ vector\ we\ want\ to\ calculate$

Note V_k will have dimensions of (500x1)

5. Compute the basis vectors.

From the previous step we have 10 Eigen vectors $(V_1, \ldots, V_{10})$, each with dimensions (500x1). We assemble these vectors into an Eigen vector matrix:

$$EV = [V_1 \quad \ldots \quad V_{10}]$$

Dimensions of EV will be (500x10)

To compute the basis vectors, we multiply the mean adjusted matrix computed in step 2 by the Eigen vector matrix:

$$S_B = S_{mean} * EV$$

Where:

$S_B = The\ basis\ vectors$ Dimensions(10,000x10)
$S_{mean} = The\ mean\ adjusted\ matrix\ computed\ in\ step\ 2$, dimensions(10,000x500)
$EV = The\ eigen\ vector\ matrix$, dimensions (500x10)

6. Represent each sample i.e., image as a linear combination of basis vectors.

Now each sample can be represented as a linear combination of basis vectors using the following formula:

$$[10\ numbers] = (S_{sample} - \bar{S})^T * S_B$$

$$[10\ numbers] = (S_{sample} - \bar{S})^T * S_B$$

Where:
$S_{sample} = The\ sample\ you\ want\ to\ represent\ using\ basis\ vectors$, dimensions(10,000x1)
$\bar{S} = The\ mean\ adjusted\ vector\ computed\ in\ step\ 2$, dimensions(10,000x1)
$S_B = The\ basis\ vectors$ Dimensions(10,000x10)

Recall that we had images of faces (100 pixels x 100 pixels) that were represented as 10,000 dimensional points in space. Now that we have completed Principal Component Analysis instead of each face being represented by 10,000 numbers, now each face is represented by 10 numbers. For a specific face, these 10 numbers represent the linear combination of basis vectors used to recreate the image of that specific face.