Principal Component Analysis

Review of Eigen values and eigen vectors

Given a $d \times d$ matrix $\mathbf{M}$, a very important class of linear equations is of the form

$$\mathbf{M}\mathbf{x} = \lambda \mathbf{x}, \tag{26}$$

which can be rewritten as

$$(\mathbf{M} - \lambda \mathbf{I})\mathbf{x} = 0, \tag{27}$$

Diagonalization property

$$M = VDV^{-1}$$

V Matrix, whose columns correspond to Eigenvectors of M

D Diagonal matrix, corresponding to eigenvalues of M

Orthogonal Matrix

$\{q_1, q_2, \ldots .. q_M\}$ M vectors

$$q_i^{\ T} q_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

$\Rightarrow$ $Q^T Q = I$

columns of Q denote
$\left[\, q_1 \; q_2 \; \ldots .. \; q_M \,\right]$

Q is of size $M \, X M$ and is said to be an 'orthogonal' matrix

Principal Component Analysis decorrelation property

- PCA also called Karnoueve Loeve (KL) Transform (in image processing). The transformed coefficients after projection may be used as features.

$$\mathbf{z} = \mathbf{W}^T (\mathbf{x} - \mathbf{m})$$

- Since the Eigen vectors of Σ are orthogonal, the $\mathbf{W}$ matrix satisfies :

$$\mathbf{W}^T \mathbf{W} = \mathbf{I}$$

- PCA de-correlates the feature vectors → covariance matrix of transformed coefficients becomes diagonal.

$$\mathbf{z} = \mathbf{W}^T (\mathbf{x} - \mathbf{m})$$

$$E(\mathbf{z}\mathbf{z}^T) = \mathbf{W}^T E((\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T)\mathbf{W}$$

$$E(\mathbf{z}\mathbf{z}^T) = \mathbf{W}^T \sum \mathbf{W} = diagonal\ \ matrix$$

- If indeed the original data is Gaussian in nature, de correlation implies independence between features

- Hence PCA are meant to give a set of independent features in a reduced dimension.

PCA as change of basis

- Change of basis.
- Data dependent unlike FFT, DCT.
- Need to compute covariance matrix using training data.

$$\mathbf{x} = \sum_{i=1}^{D} \left(\mathbf{x}^T \mathbf{w}_i \right) \mathbf{w}_i$$

Approximate signal after reduction to lower dimension

When we choose *M* eigenvectors (corresponding to the *M* largest eigen values), we can approximate the signal

$$\tilde{\mathbf{x}} = \sum_{i=1}^{M} (\mathbf{x}^T \mathbf{w}_i)\mathbf{w}_i$$

where

$\mathbf{x}^T \mathbf{w}_i$ is the projection of $\mathbf{x}$ onto the principal direction $\mathbf{w}_i$

PCA can be used for the purpose of compression !!!

Implication of eigen values

- Let the *D* dimensional feature vector be reduced to *M* dimensions.

- Ratio of sum of top *M* Eigen values of Σ to the total sum of all Eigen values (trace) captures a certain percentage of variance (denoted by *V)* .

$$V = \left(\frac{\lambda_1 + \lambda_2 + \ldots + \lambda_M}{\lambda_1 + \lambda_2 + \ldots + \lambda_M + \ldots + \lambda_D}\right) * 100 \quad (1)$$

$$\lambda_1 > \lambda_2 > \ldots \lambda_M$$

- In practice, some eigen values have little contribution to the variance and may be discarded.

- Suppose we want to retain at least x % of the variance, we sort the λ_i s in descending order and accordingly calculate the value of *M* using equation (1)

PCA-high dimensional data

- Consider, case when the number of data points is smaller than the dimensionality of the data space $N < D$

- Idea is to reduce data to M dimensions.
- example:
  - data set: a few hundred images
  - dimensionality: several million corresponding to three color values for each pixel

PCA-high dimensional data

- Standard algorithm is to find eigenvectors for a *DxD* covariance matrix
- If *D* is really high, a direct PCA is computationally infeasible
- For example, consider an image of size 128 X 128. If pixels are used as features, computation of covariance matrix will lead to a 128^2 X 128^2 square matrix (Very large !!!).
- You may have to deal with memory issues in implementation !!!
- So we need to look for a faster implementation scheme.

PCA-high dimensional data

If $N < D$

- a set of N points defines a linear subspace whose dimensionality is at most N
- there is little point to apply PCA for $M > N$

if $M > N$

- at least D-N of the eigenvalues are 0
- eigenvectors has zero variance of the data set

PCA-high dimensional data

- Define **X**: *N x D* dimensional mean-centred data matrix
- n^{th} row: $(\mathbf{x}_n - \overline{\mathbf{x}})^{\mathrm{T}}$
- Covariance matrix

$$\mathbf{\Sigma} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T$$

$$\Downarrow$$

$$\mathbf{\Sigma} = \frac{1}{N}\mathbf{X}^T\mathbf{X}$$

DxD

PCA-high dimensional data

$\mathbf{X}\mathbf{X}^T$ → *N x N* matrix is easier to handle compared to *D X D*

$$\frac{1}{N}\mathbf{X}\mathbf{X}^T\mathbf{w}_i = \lambda_i \mathbf{w}_i$$

Pre-multiplying by $\mathbf{X}^T$

$$(\frac{1}{N}\mathbf{X}^T\mathbf{X})\ (\mathbf{X}^T\mathbf{w}_i) = \lambda_i(\mathbf{X}^T\mathbf{w}_i)$$

PCA-high dimensional data

Eigenvector equation for matrix $\Sigma = \frac{1}{N}\mathbf{X}^T\mathbf{X}$

NxN

- $\frac{1}{N}\mathbf{X}\mathbf{X}^T$ and $\Sigma = N^{-1}\mathbf{X}^T\mathbf{X}$ have the same N eigenvalues

has **D-N** zero eigen values

- Eigenvectors are $\mathbf{X}^T\mathbf{w}_i$

- Eigen values of $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}\mathbf{X}^T$ are same.

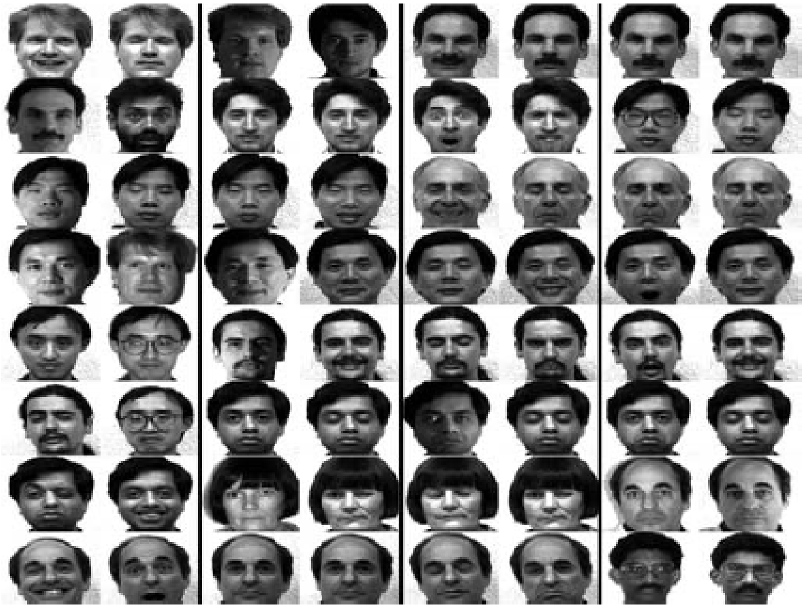
PCA-high dimensional data

- Eigenvectors $\mathbf{X}^T\mathbf{w}_i$ is not normalized !
- So it is required to normalize it to unit norm.
- One application of PCA is in face recognition (use of eigen faces for face reconstruction)

PCA-high dimensional data

Eigenfaces [Turk, Pentland '91]

- Input images:



- Principal components: