Principal Components Analysis

Ryan Henning
May 1, 2013

Principal Components Analysis (PCA) is a method for analysing high-dimensional data. PCA transforms the data such that it lays on new axes. The first axis is chosen to maximize the preservation of the data's variance. Subsequent axes are chosen such that they preserve as much of the remaining variance as possible and are orthogonal to all preceding axes. Each new axis is called a "principal component."

1 Method

Given: $\{x^{(1)}, x^{(2)}, x^{(3)}, ..., x^{(m)}\}$, $x^{(i)} \in \mathbb{R}^n$

Let $\bar{x}$ be the average value of all $x^{(i)}$:

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

Create a matrix M, the centered design matrix:

$$M_{m\times n} = \begin{bmatrix} ----- & (x^{(1)} - \bar{x}) & ----- \\ ----- & (x^{(2)} - \bar{x}) & ----- \\ ----- & (x^{(3)} - \bar{x}) & ----- \\ & ... & \\ ----- & (x^{(m)} - \bar{x}) & ----- \end{bmatrix}$$

Recall the equation for covariance between two random variables X and Y:

$$cov(X, Y) = \frac{\sum_{i=1}^{m} (X_i - \bar{X})(Y_i - \bar{Y})}{m}$$

We can easily calculate the covariance between all pairs of dimensions in our data. The following gives the covariance matrix Σ of our data:

$$\Sigma_{n\times n} = M^T M$$

The principal components are the eigenvectors of Σ. The order of the principal components is determined by the eigenvalues of Σ; that is, the first principal component (the one which captures the most variance in the data) is the eigenvector with the largest associated eigenvalue. (proof omitted)

Not all eigenvectors must be kept as the principal components. Let k be the number of eigenvectors that are kept as principal components:

$$U_{k\times n} = \begin{bmatrix} ----- & \text{first principal component} & ----- \\ ----- & \text{second principal component} & ----- \\ & ... & \end{bmatrix}$$

The original data can be projected onto the principal components as follows:

$$Y = MU^T$$

where Y is the data represented by the k dimensions defined by the principal components:

$$Y_{m \times k} = \begin{bmatrix} ----- & (y^{(1)} - \bar{y}) & ----- \\ ----- & (y^{(2)} - \bar{y}) & ----- \\ ----- & (y^{(3)} - \bar{y}) & ----- \\ & ... & \\ ----- & (y^{(m)} - \bar{y}) & ----- \end{bmatrix}$$

The original data can be restored by:

$$M' = YU$$

where M' is the restored design matrix for the data, which will have lost information if $k < n$.

2 Using SVD

If n is very large, calculating the covariance matrix is very expensive, and finding the eigenvectors of that covariance matrix is even more expensive.

Example: Each $x^{(i)}$ is an 100x100 pixel image, thus $x^{(i)} \in \mathbb{R}^{10000}$. The covariance matrix Σ will be very large, $\Sigma \in \mathbb{R}^{10000 \times 10000}$.

If $m << n$, we can avoid calculating the covariance matrix altogether using SVD.

Recall SVD: $A = UDV^T$, where

- $A_{m \times n}$ is any matrix
- $U_{m \times m}$ is an orthogonal matrix whose columns are the eigenvectors of AA^T
- $D_{m \times n}$ is a diagonal matrix of singular values
- $V_{n \times n}$ is an orthogonal matrix whose columns are the eigenvectors of $A^T A$

When $m << n$, it is much cheaper to do SVD on M to calculate the eigenvectors of $\Sigma = M^T M$.

3 Applications

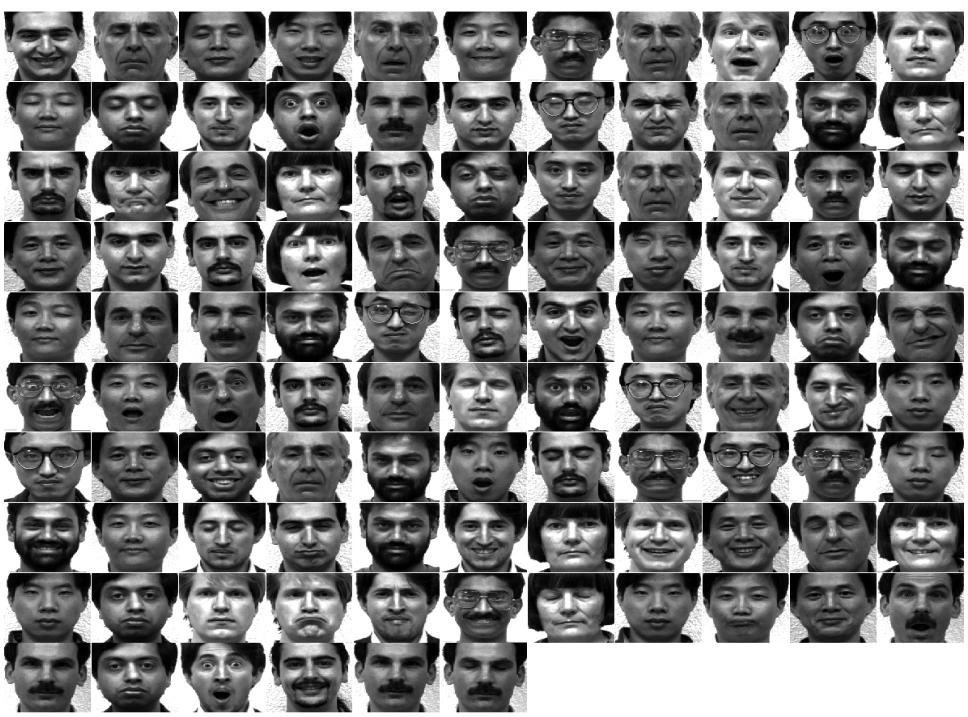
PCA is used for many things, some of which are:

- data visualization (dimensionality reduction)
- noise reduction
- compression
- face recognition (eigenfaces)

4 Eigenfaces Example

4.1 Reducing the dimensionality of images of faces

Input – 106 faces



Output – faces reduced to two dimensions

4.2 Recognition

When given a new image, find its closest match in the lower dimensional image space created from PCA. Predict that the images is of the same face as that closest match.