8.6 Principal Components Analysis (PCA)

PCA is a common technique for identifying patterns in high-dimensional data. It transforms the original correlated measurements into uncorrelated measurements. One of the main uses of PCA is as a dimension reduction tool, in which only the directions in which the data varies the most are considered. This can lead to enormous simplifications of the data and provide insights for a wide variety of data. PCA is alternatively known as the Karhunen-Loéve transform (KLT), the Hotelling transform or proper orthogonal decomposition (POD)

These new coordinate axes (along which the data varies the most) are are known as *principal components* and are, by construction, orthogonal.

A useful visualisation tool to aid your understanding of PCA is at http://setosa.io/ev/principal-component-analysis/.

Suppose we have a $m \times n$ matrix of measurement data A. For example, n trials where m properties were measured in each trial. Then, if $\mathbf{a}_i$ are the measurements from the ith trial,

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}.$$

For example, $\mathbf{A}$ could be $n = 100$ observations of the position of an object measured in $m = 3$ dimensions.

In the following, assume that the rows of $\mathbf{A}$ have been centred, so that the mean of each row is 0 (each rows of $\mathbf{A}$ corresponds to a dimension in the original data). If this is not already the case, it can be achieved by subtracting the mean of each row of $\mathbf{A}$ from each element of that row. That is, set element

$$a_{ij} = a_{ij} - \sum_{j=1}^{n} a_{ij}/n$$

to centre the rows of $\mathbf{A}$. This is a critical assumption and allows us to concentrate on the variance.

Each observation $\mathbf{A}$ is just the m-vector $\mathbf{a}_i$. The idea of PCA is to chose a new basis $\mathbf{u}_1, \dots, \mathbf{u}_k$ to express the data points (the $\mathbf{a}_i$'s) so that the variance of the measurements is greatest in the direction of $\mathbf{u}_1$, the next greatest variance is in the direction of $\mathbf{u}_2$ and so on, down to $\mathbf{u}_k$. Ideally, $k < m$.

Define the *covariance matrix* of $\mathbf{A}$ by

$$\mathbf{\Sigma} = \frac{1}{n-1}\mathbf{A}\mathbf{A}^T \approx \frac{1}{n}\mathbf{A}\mathbf{A}^{\mathbf{T}}.$$

Then $\mathbf{\Sigma}$ is an $m \times m$ matrix where the diagonal terms of $\mathbf{\Sigma}$ are the variance of the ith dimension of the measurement, while the off-diagonal terms of $\mathbf{\Sigma}$ are the covariances between different measurements.

It turns out that the best basis to choose are the k eigenvectors of $\mathbf{\Sigma}$ corresponding its k largest eigenvalues. These are known as the *principal components* of $\mathbf{A}$. Call them $\mathbf{u}_1, \ldots, \mathbf{u}_k$ and form the matrix

$$\mathbf{U}_k = [\mathbf{u}_1, \ldots, \mathbf{u}_k]$$

From results we have seen earlier about symmetric matrices, this an orthogonal matrix. We include the subscript k as we may decide to truncate this matrix by including only the eigenvectors corresponding to the largest eigenvalues. That is, if $\mathbf{\Sigma}$ has K eigenvectors and associated eigenvalues, and the largest k eigenvalues are substantially larger than the remaining $K - k$, it is reasonable to form $\mathbf{U}_k$ containing only the most significant k eigenvectors.

We can now represent the original measurements, $\mathbf{A}$, in this this new co-ordinate system. The amount of measurement vector $\mathbf{a}_i$ in direction $\mathbf{u}_j$ is given by $\mathbf{u}_j^\top \mathbf{a}_i$: this is the jth coordinate of $\mathbf{a}_i$ in this new coordinate system. So if we consider just the two dimension space defined by the top two principal components, $\mathbf{a}_i$ has coordinates

$$\mathbf{U}_2^\mathsf{T} \mathbf{a}_i = \begin{bmatrix} \mathbf{u}_1^\mathsf{T} \\ \mathbf{u}_2^\mathsf{T} \end{bmatrix} \mathbf{a}_i = \begin{bmatrix} \mathbf{u}_1^\mathsf{T} \mathbf{a}_i \\ \mathbf{u}_2^\mathsf{T} \mathbf{a}_i \end{bmatrix}.$$

We usually consider this space independently of how it relates to the original m-dimensional space but we can consider it as embedded in the original space. To find the coordinates of a point in this embedded space, define the projection matrix, $\mathbf{P}_k$ by

$$\mathbf{P}_k = \mathbf{U}_k \mathbf{U}_k^\mathsf{T} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \ldots & \mathbf{u}_k \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^\mathsf{T} \\ \mathbf{u}_2^\mathsf{T} \\ \vdots \\ \mathbf{u}_k^\mathsf{T} \end{bmatrix}$$

so that each measurement vector $\mathbf{a}_i$ is projected via $\mathbf{P}_k \mathbf{a}_i$.

One interpretation of PCA is that the projection $\mathbf{P}_k$ is chosen to minimise the projection error $\sum_{j=1}^n \|\mathbf{a}_j - \mathbf{P}_k \mathbf{a}_j\|^2$

Example: Find the principal components of the data matrix $\mathbf{A}$ where

$$\mathbf{A} = \begin{bmatrix} -4 & 3 & -5 & 18 & 6 & -5 \\ 2 & 6 & -2 & 10 & 1 & -1 \\ 7 & 11 & 3 & 6 & 9 & 3 \end{bmatrix}.$$

Find the amount of the first principal component in the first measurement vector of $\mathbf{A}$ (that is, the first column), and calculate the projection matrix for projecting $\mathbf{A}$ onto the first two principal components.

Solution: First, centre the rows of $\mathbf{A}$ so that each row has mean zero. Call this centred matrix $\mathbf{B}$.

$$\mathbf{B} = \begin{bmatrix} -6.1667 & 0.8333 & -7.1667 & 15.8333 & 3.8333 & -7.1667 \\ -0.6667 & 3.3333 & -4.6667 & 7.3333 & -1.6667 & -3.6667 \\ 0.5 & 4.5 & -3.5 & -0.5 & 2.5 & -3.5 \end{bmatrix}.$$

Now form the covariance matrix for the centred data matrix, $\mathbf{\Sigma} = \frac{1}{n}\mathbf{B}\mathbf{B}^{\mathsf{T}}$:

$$\mathbf{\Sigma} = \frac{1}{5}\begin{bmatrix} 406.8333 & 176.3333 & 52.5 \\ 176.3333 & 103.3333 & 36.0 \\ 52.5000 & 36.0000 & 51.5 \end{bmatrix}.$$

This matrix has eigenvalues 99.31, 9.46 and 3.561 corresponding to eigenvectors

$$[\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3] = \begin{bmatrix} 0.8987 & 0.2829 & 0.3352 \\ 0.4158 & -0.3062 & -0.8564 \\ 0.1396 & -0.9090 & 0.3928 \end{bmatrix} = \mathbf{U}.$$

These eigenvectors are the principal components of $\mathbf{A}$ (and of $\mathbf{B}$).

The amount of the first principal component in $\mathbf{a}_1$ is

$$\mathbf{u}_1^{\top}\mathbf{a}_1 = [0.8987, 0.4158, 0.1396]\begin{bmatrix} -4 \\ 2 \\ 7 \end{bmatrix} = -1.756.$$

To project $\mathbf{A}$ into the coordinate system defined by the first two principal components, form the projection matrix,

$$\mathbf{P}_2 = \mathbf{U}_2\mathbf{U}_2^{\mathsf{T}} = \begin{bmatrix} 0.8987 & 0.2829 \\ 0.4158 & -0.3062 \\ 0.1396 & -0.9090 \end{bmatrix}\begin{bmatrix} 0.8987 & 0.4158 & 0.1396 \\ 0.2829 & -0.3062 & -0.9090 \end{bmatrix} = \begin{bmatrix} 0.8877 & 0.2870 & -0.1316 \\ 0.2870 & 0.2666 & 0.3364 \\ -0.1316 & 0.3368 & 0.8457 \end{bmatrix}$$

8.7 Examples

See associated slides for population structure in Europe and Eigenfaces.

The “eigenfaces” example in the slides was developed by Matthew Turk and Alex Pentland (Journal of Cognitive Neuroscience, 1991, v3 (1)). The following quote is from their abstract:

> We have developed a near-real-time computer system that can locate and track a subject’s head, and then recognize the person by comparing characteristics of the face to those of known individuals. ... The system functions by projecting face images onto a feature space that spans the significant variations among known face images. The significant features are known as ”eigenfaces,” because they are the eigenvectors (principal components) of the set of faces; they do not necessarily correspond to features such as eyes, ears, and noses. The projection operation characterizes an individual face by a weighted sum of the eigenface features, and so to recognize a particular face it is necessary only to compare these weights to those of known individuals. Some particular advantages of our approach are that it provides for the ability to learn and later recognize new faces in an unsupervised manner, and that it is easy to implement using a neural network architecture.