Principal component analysis

Principal component analysis (PCA) is a powerful tool based on linear algebra that can help in applications such as data analysis, computer graphics or data compression. PCA can be used to extract essential features or structures of a data set that can not readily be observed. For example, sensors are used in experiments to record data. To capture the relevant information to understand the underlying physics, sensors must be placed at correct locations which are generally unknown. PCA can then be used to express the data in a new basis that reflects the most significant components of the underlying data which would have been recorded if the optimal sensor locations would have been known beforehand.

1 Introduction

A data set $\mathcal{X} \in \mathbb{R}^l$ is said to have intrinsic dimensionality $m \leq l$ if $\mathcal{X}$ can be (approximately) described in terms of m free parameters.

Example: Assume vectors in $\mathcal{X}$ are generated as functions in terms of m random variables.

$$\mathbf{x} = g(u_1, \ldots, u_m), \qquad u_i \in \mathbb{R}, \quad i = 1, \ldots, m. \tag{1}$$

The respective observation vectors will lie along a manifold whose form depends on the vector valued function $g\colon \mathbb{R}^m \to \mathbb{R}^l$. Assume the following form of function g:

$$\mathbf{x} = [r\cos\theta, r\sin\theta]^T, \qquad r = \text{const}, \ \theta \in [0, 2\pi] \tag{2}$$

In this case, one parameter suffices to describe the data. If a small noise is added then data will cluster around the circumference (see Fig. 1). Statistically this implies that the data will be correlated.

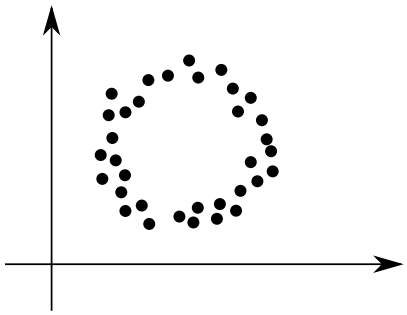

Figure 1: A sample of points that can be well described using a single parameter θ (see Eq. 2).

2 PCA

Assume the observed data are generated by a system or a process that is driven by a relatively small number of *latent* (not directly observable) variables. The

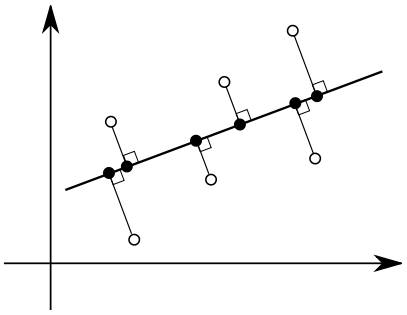

Figure 2: PCA maximises the variance of points •. If you are to select a single principal component you want it to account for most variability possible so the compact representation collects the most "uniqueness" from the data set.

goal is to learn this *latent structure*. Given a vector with l elements (*data*, variables)

$$\mathbf{x}_n \in \mathbb{R}^l, \quad n = 1, 2, \ldots, N \tag{3}$$

which is *assumed to be of zero mean* (else the mean is subtracted) and n denotes the n-th observation out of N samples. The PCA determines the subspace of dimension $m \leq l$ such that after the projection to this subspace, the statistical variation of the data is optimally retained.

This subspace has m mutually orthogonal axes. They are computed so that the variance of the data after projection on the subspace is maximised (see Fig. 2). PCA does not increase the variance, it rotates the data in such a way that the directions with the most spread (variance) are aligned with the principal components.

2.1 First principal component

Assume $m = 1$ and we want to find a single direction in $\mathbb{R}^l$ so that the variance of the corresponding projected points is maximised. Let $\mathbf{u}_1 \in \mathbb{R}^l$ denote the principal axis, then the functional $I\colon \mathbb{R}^l \to \mathbb{R}$ expresses the variance of the projection $\mathbf{u}_1^T\mathbf{x}_n$, $n = 1, 2, \ldots, N$, along the axis $\mathbf{u}_1$

$$\begin{aligned} I(\mathbf{u}_1) &= \frac{1}{N}\sum_{n=1}^{N}(\mathbf{u}_1^T\mathbf{x}_n)^2 = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{u}_1^T\mathbf{x}_n)(\mathbf{x}_n^T\mathbf{u}_1) \\ &= \frac{1}{N}\sum_{n=1}^{N}\mathbf{u}_1^T\mathbf{x}_n\mathbf{x}_n^T\mathbf{u}_1 = \mathbf{u}_1^T C\mathbf{u}_1, \end{aligned} \tag{4}$$

where $C \in \mathbb{R}^{l\times l}$ is the sample covariance matrix of the data

$$C = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^T. \tag{5}$$

2.1.1 Sample mean

Recall that the PCA is sensitive to data that is not in mean deviation form, that is, data with zero mean. We have assumed previously that $\mathbf{x}_n$ is a vector

of l variables for the n-th observation with *zero mean*, where the observations range from $n = 1, 2, \ldots, N$. In practice, we are often faced with data $\hat{\mathbf{x}}_n$ for which we do not know the mean a priori. We can simply transform this data into mean deviation form by subtracting the sample mean vector $\bar{\mathbf{x}}$. The sample mean vector is trivially computed from the observations $\hat{\mathbf{x}}_n$ by

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^{N} \hat{\mathbf{x}}_n, \tag{6}$$

where the mean deviation form is then obtained by subtracting the sample mean $\hat{\mathbf{x}}_n - \bar{\mathbf{x}}$.

2.1.2 Sample covariance

In the case where we start working with data $\hat{\mathbf{x}}_n$, we must compute the sample covariance matrix with data in mean deviation form, as we have discussed in the previous section. In such case, the sample covariance is computed by the relation

$$C = \frac{1}{N-1} \sum_{n=1}^{N} (\hat{\mathbf{x}}_n - \bar{\mathbf{x}})(\hat{\mathbf{x}}_n - \bar{\mathbf{x}})^T, \tag{7}$$

which is not the same as Eq. (5) for which the mean is known a priori.

2.2 How to maximise the functional I with respect to u?

This must be a constrained minimisation otherwise $\|\mathbf{u}_1\| \to \infty$ will be a valid solution. The appropriate constraint is $\mathbf{u}_1^T \mathbf{u}_1 = 1$. We now use Lagrange multipliers to solve the problem

$$\mathbf{u}_1 = \arg\max_{\mathbf{u}} \mathbf{u}^T C \mathbf{u} \tag{8}$$

$$\text{s.t. } \mathbf{u}^T \mathbf{u} = 1 \tag{9}$$

The constrained optimization problem can be formulated with a Lagrangian given by

$$\mathcal{L}(\mathbf{u}, \lambda) = \mathbf{u}^T C \mathbf{u} - \lambda(\mathbf{u}^T \mathbf{u} - 1), \tag{10}$$

where

$$\frac{\partial \mathcal{L}}{\partial \mathbf{u}} = 0 \quad \Longrightarrow \quad C\mathbf{u} = \lambda \mathbf{u}. \tag{11}$$

Therefore, we find that $\mathbf{u}$ are the eigenvectors of C and further

$$I(\mathbf{u}) = \mathbf{u}^T C \mathbf{u} = \mathbf{u}^T \lambda \mathbf{u} = \lambda, \tag{12}$$

where λ are the eigenvalues. Hence, $I(\mathbf{u})$ is *maximised* if $\mathbf{u}_1$ is the eigenvector that corresponds to the maximum eigenvalue λ_1. C is symmetric and positive semi-definite, therefore all eigenvalues are non-negative. The second principal component is obtained such that $\mathbf{u}_2 \perp \mathbf{u}_1$ and maximises the variance along this new axis. It can be shown that the second principle axis is the eigenvector corresponding to the second largest eigenvalue $\lambda_2 < \lambda_1$.

To summarize:

1. Transform the general data $\hat{\mathbf{x}}_n$ into mean deviation form using the sample mean vector $\bar{\mathbf{x}}$, Eq. (6).

2. Compute the sample covariance matrix given by Eq. (7).

3. Find the m eigenvectors that correspond to the m largest eigenvalues. For an $l \times l$ matrix the associated cost for this operation is $\mathcal{O}(l^3)$. If only the first principal component is of interest, a famous method to compute the largest eigenvalue and the associated eigenvector is called the *power method*. However, its convergence is only linear and depends on the factor $|\lambda_2/\lambda_1|$, that is, the convergence is very slow for cases where the largest eigenvalue is close to the second largest eigenvalue. If the complete solution to the eigenvalue problem is required, eigenvalue-revealing factorizations are usually applied, such as Schur factorization.

3 Minimum error formulation

Introduce a complete orthonormal set of D-dimensional basis vectors u_i where $i = 1, 2, \ldots, D$

$$\mathbf{u}_i^T \mathbf{u}_j = \delta_{ij} \tag{13}$$

Basis is complete so that

$$\mathbf{x_n} = \sum_{i=1}^{D} \alpha_{ni} \mathbf{u}_i \tag{14}$$

note that this corresponds to a change of the coordinate system.
Inner product with u_j:

$$\mathbf{x}_n^T = \sum_{i=1}^{D} \alpha_i \mathbf{u}_i^T \quad \text{and} \quad \mathbf{x}_n^T \mathbf{u}_j = \sum_{i=1}^{D} \alpha_{ni} \mathbf{u}_i^T \mathbf{u}_j \tag{15}$$

$$\alpha_{nj} = \mathbf{x}_n^T \mathbf{u}_j \tag{16}$$

So

$$\mathbf{x}_n = \sum_{i=1}^{D} (\mathbf{x}_n^T \mathbf{u}_i) \mathbf{u}_i \tag{17}$$

But we do not wish to perform a rotation, we wish to perform a *restricted representation using $M < D$ vectors*.
The M-dimensional landscape can be represented without loss of generality by the M vectors

$$\tilde{\mathbf{x}_n} = \sum_{i=1}^{M} z_{ni} \mathbf{u}_i + \sum_{i=M+1}^{D} b_i \mathbf{u}_i \tag{18}$$

z_{ni} depend on the data points and b_i are constants. Choose $\mathbf{u}_i$, z_{ni} and b_i to minimise *distortion*.
Define

$$J = \frac{1}{N} \sum_{n=1}^{N} \|\mathbf{x_n} - \tilde{\mathbf{x_n}}\|^2 \tag{19}$$

Substitute for $\tilde{\mathbf{x}_n}$, take derivative $\frac{\partial J}{\partial z_{ni}} = 0$ and using orthogonality we obtain

$$z_{nj} = \mathbf{x}_n^T \mathbf{u}_j, \text{ for } j = 1...M \tag{20}$$

What if we had chosen another cost function?

Setting $\frac{\partial J}{\partial b_i} = 0$ and using again orthogonality, we get

$$b_j = \overline{\mathbf{x}}^T \mathbf{u}_j, \ j = M+1 ... D \tag{21}$$

where $\overline{\mathbf{x}}$ is the sample mean.

$$\mathbf{x}_n - \tilde{\mathbf{x}_n} = \sum_{i=M+1}^{D} (\mathbf{x}_n - \overline{\mathbf{x}})^T \mathbf{u}_i \mathbf{u}_i, \text{ where } \mathbf{x}_n - \tilde{\mathbf{x}_n} \text{ is the distortion} \tag{22}$$

so orthogonal to the principal subspace.
We minimise the distortion measure

$$J = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=M+1}^{D} \left(\mathbf{x}_n^T \mathbf{u}_i - \overline{\mathbf{x}}^T \mathbf{u}_i \right)^2 = \sum_{M+1}^{D} \mathbf{u}_i^T C \mathbf{u}_i \tag{23}$$

3.1 Minimization of J with respect to $\mathbf{u}_i$ constraint else $\mathbf{u}_i = 0$

Before a general solution, we try the intuition for $D = 2$ and $M = 1$. Therefore

$$J = \mathbf{u}_2^T C \mathbf{u}_2 \quad \text{and} \quad \mathbf{u}_2^T \mathbf{u}_2 = 1 \tag{24}$$

$$\tilde{J} = \mathbf{u}_2^T C \mathbf{u}_2 + \lambda_2 (1 - \mathbf{u}_2^T \mathbf{u}_2) \tag{25}$$

Set $\frac{\partial \tilde{J}}{\partial u_2} = 0$ and obtain

$$C \mathbf{u}_2 = \lambda_2 \mathbf{u}_2 \tag{26}$$

which is an eigenvalue problem. Note that $J = \lambda_2$.

So we minimise the distortion by choosing the u_2 that corresponds to the smaller from the two eigenvalues. Thus the principal space is along the eigenvector having the largest eigenvalue.

Maximise variance along a direction that passes through the mean by solving:

$$C \mathbf{u}_i = \lambda_i \mathbf{u}_i \tag{27}$$

and distortion is

$$J = \sum_{i=M+1}^{D} \lambda_i \tag{28}$$

4 PCA for high dimensional data

E.g. apply PCA to $O(100)$ images each of which corresponds to a vector in a space of potentially several million dimensions (corresponding to three color values for each of the pixels in the image).

In a D-dimensional space, a set of N points, where $N < D$, defines a linear subspace whose dimensionality is at most $N - 1$ and so there is little point in applying PCA for values of $M > N - 1$.

If we perform PCA we will find that at least $D - (N - 1)$ of the eigenvalues are zero.

Also note that with a cost of $O(D^3)$, most PCA on images will be very expensive.

HOW TO RESOLVE THIS?

X is $N \times D$, dimensional centered matrix whose n-th row is $(X_n - \bar{X})^T$. The $D \times D$ covariance matrix is

$$C = \frac{1}{N} X^T X \tag{29}$$

which leads to the eigenvalue problem:

$$\frac{1}{N} X^T X \mathbf{u_i} = \lambda_i \mathbf{u_i} \tag{30}$$

Pre-multiply with X and get

$$\frac{1}{N} X X^T (X \mathbf{u_i}) = \lambda_i (X \mathbf{u_i}) \tag{31}$$

Here we can define $\mathbf{v_i} = \mathbf{X u_i}$ and get:

$$\frac{1}{N} X X^T \mathbf{v_i} = \lambda_i \mathbf{v_i}, \tag{32}$$

which is the eigenvalue problem in the $N \times N$ space. Note that the $N - 1$ eigenvalues λ_i are equal to the first $N-1$ eigenvalues of the matrix $X^T X$ (which has $D - (N - 1)$ zero eigenvalues). The computational cost is decreased from $O(D^3)$ to $O(N^3)$, and we can derive the eigenvectors of $X^T X$ by pre-multiplying eq 32 by X^T:

$$\frac{1}{N} (X^T X)(X^T v_i) = \lambda_i X^T \mathbf{v}_i \tag{33}$$

This is again the original eigenvalue problem of the matrix $C = X^T X$ where we already have computed the eigenvectors $X^T \mathbf{v}_i$ and eigenvalue λ_i. Therefore the eigenvectors of $X X^T$ can be written as:

$$\mathbf{u}_i = \frac{1}{\sqrt{N \lambda_i}} X^T \mathbf{v}_i \tag{34}$$

Therefore the PCA of a dataset X where $N < D$ is performed by solving the eigenvalue problem for $X X^T$, which yields eigenvectors that lie in a N-dimensional space, and then computing the principal components with equation 34.

5 Kernel PCA

Given a dataset of d dimensional vectors N_n with $n \in \{1, N\}$, it is not in general possible to linearly separate them along $M < d$ principal components. Kernel PCA methods introduce a non-linear kernel ϕ that maps the data onto an $M > d$ dimensional space, where it is more likely to find linear relations that describe the features of the data. Therefore each point $\mathbf{x}_n$ has a corresponding point n feature space $\phi(\mathbf{x}_n)$. Assuming that the data is centered in feature space ($\sum_{n=1}^{N} \phi(\mathbf{x}_n) = \mathbf{0}$), the covariance matrix of the dataset projected in feature space is given by:

$$C = \frac{1}{N} \sum_{n=1}^{N} \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T \tag{35}$$

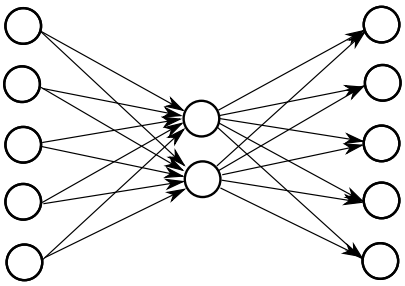

Figure 3: Example of an autoassociative network. From left to right: Input layer $\mathbf{x}_i$ (circles), matrix ϕ_E (arrows), hidden layer $\mathbf{z}_i$, matrix ϕ_D, output layer $\tilde{\mathbf{x}}_i$ (see Eq. 37).

The PCA can be written in feature space as the eigenvalue problem:

$$C\mathbf{v}_i = \lambda_i \mathbf{v}_i \tag{36}$$

Kernel PCA methods solve this eigenvalue problem without having to work in the potentially high-dimensional feature space.

6 Auto-associative NN

Consider a Neural Network (NN) mapping the input $\mathbf{x}_i \in \mathbb{R}^d$ onto an output $\tilde{\mathbf{x}}_i \in \mathbb{R}^d$ through an intermediate feature space $\mathbf{z}_i \in \mathbb{R}^M$ (see Fig. 3):

$$\mathbf{z}_i = \phi_E(\mathbf{x}_i), \qquad \tilde{\mathbf{x}}_i = \phi_D(\mathbf{z}_i) \tag{37}$$

Here ϕ_E and ϕ_D denote the mapping to and from feature space defined by the weights w of the NN. One example of such mapping are the linear relations $\mathbf{z}_i = \Phi_E \mathbf{x}_i$ and $\tilde{\mathbf{x}}_i = \Phi_D \mathbf{z}_i$, where Φ_E and Φ_D are matrices whose elements are defined by the weights w. Auto-associative mapping consists in learning the weights w such that the output $\tilde{\mathbf{x}}_i$ replicates the input $\mathbf{x}_i$. This is achieved by minimizing the cost function:

$$\mathcal{L}(w) = \frac{1}{2} \sum_{n=1}^{N} \|\tilde{\mathbf{x}}_i - \mathbf{x}_i\|^2 \tag{38}$$

If both ϕ_E and ϕ_D are linear relations (i.e. if the activations of all hidden units of the network are linear functions) then it can be shown that $\mathcal{L}(w)$ has an unique global minimum and the network performs a projection onto the M-dimensional subspace which is spanned by the first M principal components of the data.