Mathematics for Machine Learning

Prof Willie Brink

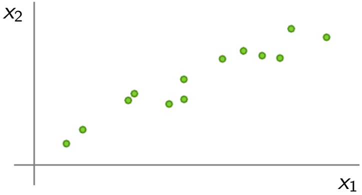
Applied Mathematics, Stellenbosch University

Lecture 9: Dimensionality Reduction with PCA

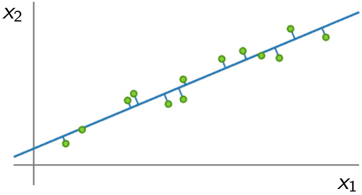
Contents of the module

Introduction

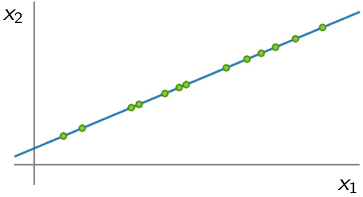
1.



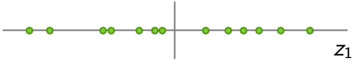
2.



3.



4.



10.1 Problem setting

High-dimensional data is often overcomplete (many redundant dimensions), and may occupy a much lower-dimensional subspace.

Consider data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ in $\mathbb{R}^D$, with a mean of $\mathbf{0}$.

PCA: find projections $\tilde{\boldsymbol{x}}_n$ of data points $\boldsymbol{x}_n$, that are similar to original data but have a significantly lower intrinsic dimensionality.

Let $\boldsymbol{B} = [\,\boldsymbol{b}_1,\,\ldots,\,\boldsymbol{b}_M\,] \in \mathbb{R}^{D\times M}$ be a projection matrix with orthonormal columns, where $M \ll D$. Our task will be to find this matrix $\boldsymbol{B}$ for a given dataset.

Encoding $\boldsymbol{x}_n$ to a low-dimensional representation: $\boldsymbol{z}_n = \boldsymbol{B}^\top \boldsymbol{x}_n \in \mathbb{R}^M$

Decoding $\boldsymbol{z}_n$ in order to reconstruct $\boldsymbol{x}_n$: $\tilde{\boldsymbol{x}}_n = \boldsymbol{B}\boldsymbol{z}_n = \boldsymbol{B}\boldsymbol{B}^\top \boldsymbol{x}_n \in \mathbb{R}^D$

10.2 Maximum variance perspective

Find a $\boldsymbol{B}$ that retains as much information as possible, i.e. captures the most variance.

Let's start by finding a single vector $\boldsymbol{b}_1 \in \mathbb{R}^D$ that maximises the variance of the first coordinate z_1 of the encodings (the first principal component).

$$V_1 = \mathbb{V}[z_1] = \frac{1}{N}\sum_{n=1}^{N} z_{1,n}^2 = \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{b}_1^\top \boldsymbol{x}_n\right)^2 = \boldsymbol{b}_1^\top \left(\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n\boldsymbol{x}_n^\top\right)\boldsymbol{b}_1 = \boldsymbol{b}_1^\top \boldsymbol{S}\boldsymbol{b}_1$$

Solve the constrained optimisation problem: $\max_{\boldsymbol{b}_1} \boldsymbol{b}_1^\top \boldsymbol{S}\boldsymbol{b}_1$ subject to $\|\boldsymbol{b}_1\|^2 = 1$

Introducing a Lagrange multiplier λ_1 and setting derivatives w.r.t. $\boldsymbol{b}_1$ and λ_1 to 0, give

$\boldsymbol{S}\boldsymbol{b}_1 = \lambda_1 \boldsymbol{b}_1, \;\; \boldsymbol{b}_1^\top \boldsymbol{b}_1 = 1$ and note then that $V_1 = \lambda_1 \boldsymbol{b}_1^\top \boldsymbol{b}_1 = \lambda_1$

Therefore we choose $\boldsymbol{b}_1$ as the eigenvector of $\boldsymbol{S}$ associated with its largest eigenvalue.

$\boldsymbol{b}_1$ is the eigenvector of data covariance matrix $\boldsymbol{S}$ associated with its largest eigenvalue.

The second principal component, $\boldsymbol{b}_2$, will be the eigenvector of $\boldsymbol{S}$ associated with the second largest eigenvalue, and so on.

The first M principal components form an ONB for an M-dimensional subspace of $\mathbb{R}^D$.

The maximum amount of variance that PCA can capture is $V_M = \sum_{m=1}^{M} \lambda_m$

The variance lost by PCA's compression is $J_M = \sum_{j=M+1}^{D} \lambda_j = V_D - V_M$

Note: if the data is not centered at $\mathbf{0}$, we would first subtract the data mean $\boldsymbol{\mu}$ from each $\boldsymbol{x}_n$ before forming $\boldsymbol{S}$ and finding $\boldsymbol{B}$.

The encoding would then be $\boldsymbol{z}_n = \boldsymbol{B}^\mathsf{T}(\boldsymbol{x}_n - \boldsymbol{\mu})$, and the decoding would be $\tilde{\boldsymbol{x}}_n = \boldsymbol{B}\boldsymbol{z}_n + \boldsymbol{\mu}$.

10.3 Projection perspective

PCA can also be derived from the perspective of a linear encoder-decoder that minimises the average reconstruction error.

The aim is to find vectors $\tilde{\boldsymbol{x}}_n \in \mathbb{R}^D$ that lie in an M-dimensional subspace spanned by an unknown ONB $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_M)$, that is as close as possible to the original data $\boldsymbol{x}_n$, i.e. that minimise the average reconstruction error:

$$\frac{1}{N} \sum_{n=1}^{N} \|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2$$

It turns out that for a given ONB an orthogonal projection gives the optimal encoding.

It also turns out that minimising the reconstruction error is equivalent to minimising the variance we ignore when projecting to the subspace, leading to the same solution as before (eigenvectors of the data covariance matrix $\boldsymbol{S}$).

10.4 Eigenvector computation

Let $\boldsymbol{X} = [\,\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\,] \in \mathbb{R}^{D \times N}$. We obtain the principal components as eigenvectors of the data covariance matrix $\boldsymbol{S}$, where

$$\boldsymbol{S} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^\top = \frac{1}{N} \boldsymbol{X}\boldsymbol{X}^\top$$

Recall that the first M cols of $\boldsymbol{U}$ in the SVD of $\boldsymbol{X}$ give us exactly those eigenvectors!

The eigenvalues λ_m of $\boldsymbol{S}$ are related to the singular values σ_m of $\boldsymbol{X}$ via: $\lambda_m = \sigma_m^2 / N$.

We would normally use the SVD of $\boldsymbol{X}$ to perform PCA, for its numerical stability and computational efficiency (compared to an eigendecomposition of $\boldsymbol{S}$).

10.6 Key steps of PCA in practice

Given a dataset of points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$ in $\mathbb{R}^D$.

1. Mean subtraction

Center the data at $\mathbf{0}$ by subtracting the mean $\boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n$ from each $\boldsymbol{x}_n$.

2. Standardisation

Divide each data point by the standard deviation σ_d for every dimension $d = 1, \dots, D$.
Now the data has variance 1 along each axis.

3. Determining the principal components

Concatenate the centered, standardised data vectors as columns of $\boldsymbol{X}$, and let $\boldsymbol{B}$ be the first M columns of $\boldsymbol{U}$ in the SVD of $\boldsymbol{X}$.

4. Projection

Any point $\boldsymbol{x} \in \mathbb{R}^D$ (from the same data generating process as the given dataset) can be encoded as a lower-dimensional vector: $\boldsymbol{z} = \boldsymbol{B}^\mathsf{T}\boldsymbol{x}_*$, where $\boldsymbol{x}_*$ has components

$$x_*^{(d)} = \frac{x^{(d)} - \mu_d}{\sigma_d}$$

The vector $\boldsymbol{z}$ is an M-dimensional representation of the D-dimensional vector $\boldsymbol{x}$.

5. Reconstruction

A representation $\boldsymbol{z}$ is transformed back to D-dimensional space by $\tilde{\boldsymbol{x}}_* = \boldsymbol{B}\boldsymbol{z}$, and then de-standardised:

$$\tilde{x}^{(d)} = \tilde{x}_*^{(d)}\sigma_d + \mu_d$$

The vector $\tilde{\boldsymbol{x}}$ might be an approximation of the original $\boldsymbol{x}$ from step 4.

PCA on an image dataset

Dataset:  $\cdots$

 $\approx$  $+ z_1$  $+ z_2$  $+ z_3$  $+ z_4$ 

x μ b_1 b_2 b_3 b_4

$$z = [\, z_1 \;\; z_2 \;\; z_3 \;\; z_4 \,]^T$$