FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
NATURWISSENSCHAFTLICHE FAKULTÄT

Principal Component Analysis

Lecture “Mathematics of Learning”

Andreas Bärmann
Friedrich-Alexander-Universität Erlangen-Nürnberg

WIRTSCHAFTS MATHEMATIK

Principal Component Analysis PCA

Principal Component Analysis (PCA):

- first idea by Karl Pearson in 1906
- improvements by Harold Hotelling in the 1930s
- widespread use since raise of **computers**

Applications:

- Multivariate statistics
- Cluster analysis
- Data reduction
- Feature extraction
- Image processing
- ...

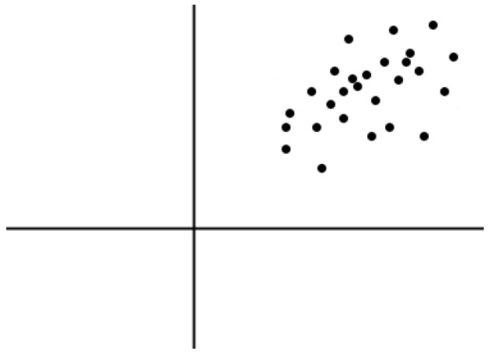
Prelimininaries

- general approach from multivariate statistics
- structures large data sets through eigenvalues and covariances
- represents data through principal components, i.e. linear combinations of statistical variables

What is given?

- input data set with $N \in \mathbb{N}$ points $x^{(1)}, \dots, x^{(N)} \in \mathbb{R}^M$
- no a-priori knowledge about data needed (e.g. cluster label)
- statistically interpretable as N observations of M random variables (e.g. we have measured M so-called *features* for N people / objects.)

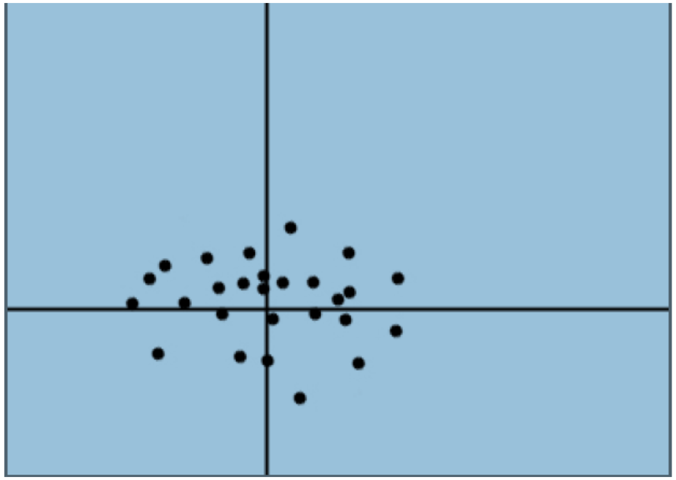
Objectives of PCA



What is the goal?

- Structure identification in the data
- Extraction of meaningful features
- Data reduction to most expressive information,
  i.e. project data points in k-dimensional space, with $k < M$,
  such that no or not much information is lost.

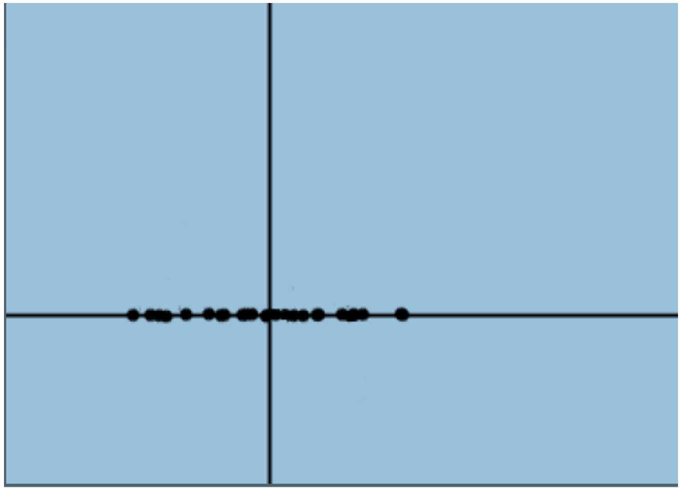
Objectives of PCA



What is the goal?

- Structure identification in the data
- Extraction of meaningful features
- Data reduction to most expressive information,
  i.e. project data points in k-dimensional space, with $k < M$,
  such that no or not much information is lost.

Objectives of PCA



What is the goal?

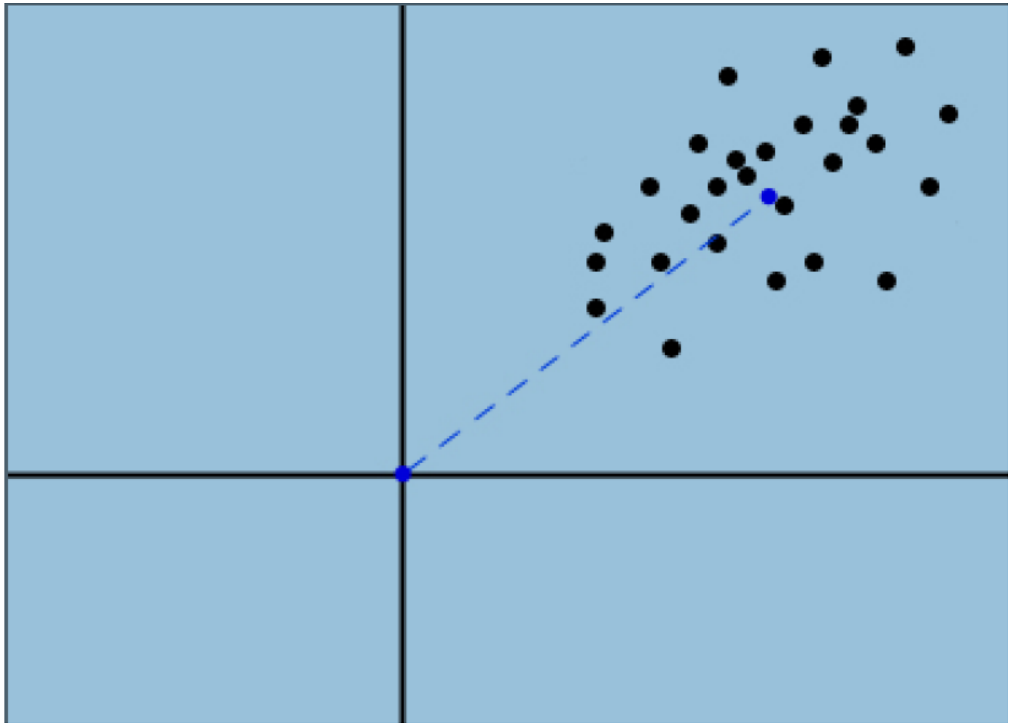
- Structure identification in the data
- Extraction of meaningful features
- Data reduction to most expressive information,
  i.e. project data points in k-dimensional space, with $k < M$,
  such that no or not much information is lost.

Computing PCA: Data centering

Centering the data in the origin

$\rightarrow$ Computation of mean value $\overline{X} = \frac{1}{N}\sum_{i=1}^{N} x^{(i)}$

In the following: $y^{(i)} = x^{(i)} - \overline{X}, i = 1, \ldots, N$ centred data



Computing PCA: Covariance matrix

Computation of covariance matrix $C \in \mathbb{R}^{M \times M}$:

$$\boxed{C := \frac{1}{N} \sum_{i=1}^{N} y^{(i)} {y^{(i)}}^T}$$

$$\begin{aligned} C_{k,l} &= \frac{1}{N} \sum_{i=1}^{N} {y_k}^{(i)} {y_l}^{(i)} = \frac{1}{N} \sum_{i=1}^{N} ({y_k}^{(i)} - 0)({y_l}^{(i)} - 0) \\ &= \frac{1}{N} \sum_{i=1}^{N} ({y_k}^{(i)} - \overline{Y_k})({y_l}^{(i)} - \overline{Y_l}) =: Cov(y_k, y_l) \end{aligned}$$

Recall Linear Algebra Lectures: Diagonalisation of C

Aim: Alternative data representation: $y^{(i)} \in \mathbb{R}^M \to z^{(i)} \in \mathbb{R}^k$,

- based on orthogonal vectors ('principal components')
- vectors should be aligned with directions of highest variance
- data representation should be **uncorrelated**
  $\to Cov(z_j, z_l) = 0$ for $j \neq l$
  $\to$ diagonalisation of matrix C

(Finite-dimensional) spectral theorem from Linear Algebra:

Let $C \in \mathbb{R}^{M \times M}$ be a real, symmetric matrix. Then there exists an orthogonal matrix S such that:

$$S^T C S = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_M \end{pmatrix},$$

for which $\lambda_1, \dots, \lambda_M \in \mathbb{R}$ are the eigenvalues of C.
The columns of S are orthonormal eigenvectors of C.

Computing PCA: Solving the eigenvalue problem

We get the wanted alternative data representation by computing **eigenvalues** and respective **eigenvectors** of C.
Thus, we need to (numerically) solve the eigenvalue problem:

$$\lambda v = Cv$$

Recall: A solution can be found by various methods :

- roots of characteristic polynomial
- QR algorithm
- Jacobi eigenvalue algorithm
- singular value decomposition
- etc.

Observations:

- C positive semi-definite $\Rightarrow$ only non-negative eigenvalues
- $\lambda_j \equiv$ data variance along direction of eigenvector $v^{(j)}$
- eigenvectors form a new local coordinate system

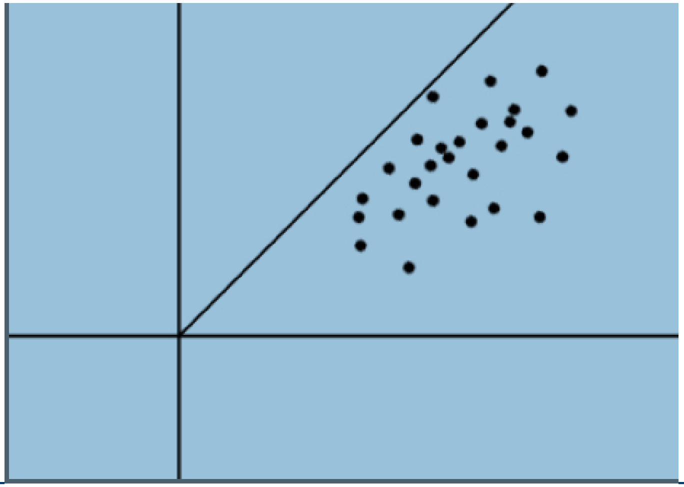
Computing PCA: Solving the eigenvalue problem

A simple example:

- Data is distributed within hyperplane parallel to plane $span(e_1, e_2)$
  $\rightarrow$ no variance in direction e_3 (no depth)
- easy to recognize from eigenvalue λ_3, because $S^T C S$ leads to

$$D = \begin{pmatrix} \lambda_1 > 0 & 0 & 0 \\ 0 & \lambda_2 > 0 & 0 \\ 0 & 0 & \lambda_3 = 0 \end{pmatrix}$$

- Selection of eigenvalues $\lambda_1, \lambda_2 > 0$ leads to dimension reduction.



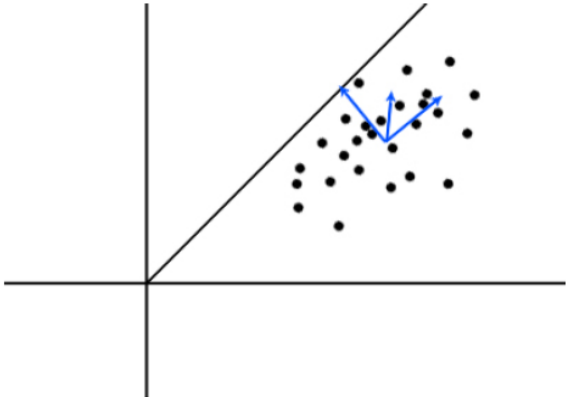
Computing PCA: Solving the eigenvalue problem

A simple example:

- Data is distributed within hyperplane parallel to plane $span(e_1, e_2)$
  $\rightarrow$ no variance in direction e_3 (no depth)
- easy to recognize from eigenvalue λ_3, because $S^T C S$ leads to

$$D = \begin{pmatrix} \lambda_1 > 0 & 0 & 0 \\ 0 & \lambda_2 > 0 & 0 \\ 0 & 0 & \lambda_3 = 0 \end{pmatrix}$$

- Selection of eigenvalues $\lambda_1, \lambda_2 > 0$ leads to dimension reduction.



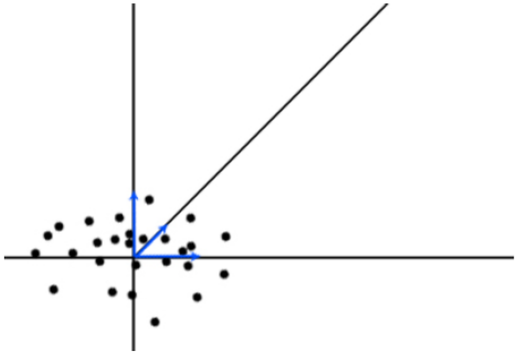
Computing PCA: Solving the eigenvalue problem

A simple example:

- Data is distributed within hyperplane parallel to plane $span(e_1, e_2)$
  $\rightarrow$ no variance in direction e_3 (no depth)
- easy to recognize from eigenvalue λ_3, because $S^T CS$ leads to

$$D = \begin{pmatrix} \lambda_1 > 0 & 0 & 0 \\ 0 & \lambda_2 > 0 & 0 \\ 0 & 0 & \lambda_3 = 0 \end{pmatrix}$$

- Selection of eigenvalues $\lambda_1, \lambda_2 > 0$ leads to dimension reduction.



The actual PCA

Define transformation matrix:

$$T := (v^{(1)}, \dots, v^{(k)}) \in \mathbb{R}^{M \times k},$$

for which $v^{(1)}, \dots, v^{(k)}$ are the respective eigenvectors of the $1 \leq k \leq M$ largest eigenvalues.

Principal component analysis:

- transform the data: $z^{(i)} := T^T y^{(i)} = T^T(x^{(i)} - \overline{X})$ for $i = 1, \dots, N$
- $z^{(i)} \in \mathbb{R}^k$ contains the most relevant information (features) of the input data
- The components $z_j^{(i)}, j = 1, \dots, k$ are called **principal components**
- If T is quadratic ($k = M$) $\Rightarrow$ PCA is simply a rotation in $\mathbb{R}^M$

The principal components of the input data are typically used as (cluster) representatives in **clustering tasks**.

Summary of PCA

For given input data $x^{(1)}, \dots, x^{(N)} \in \mathbb{R}^M$ the PCA can be computed as

The (linear) PCA algorithm

1. Compute mean value of data $\overline{X} = \frac{1}{N}\sum_{i=1}^{N} x^{(i)}$
2. Center data via $y^{(i)} = x^{(i)} - \overline{X}$
3. Compute covariance matrix $C = \frac{1}{N}\sum_{i=1}^{N} y^{(i)} {y^{(i)}}^T$
4. Determine the M eigenvalues and eigenvectors of C numerically
5. Select $1 \leq k \leq M$ respective eigenvectors $v^{(1)}, \dots, v^{(k)}$ of the k largest non-vanishing eigenvalues
6. Assemble selected eigenvectors $v^{(1)}, \dots, v^{(k)}$ columnwise to matrix $T \in \mathbb{R}^{M \times k}$
7. Compute principal components for each centred input point $y^{(i)} \in \mathbb{R}^M$ via:
$$T^T y^{(i)} = z^{(i)} \in \mathbb{R}^k$$

Properties of the PCA

Data reconstruction

Reconstructing the centred input data from its principal components is (partially) possible via:

$$Tz^{(i)} = \tilde{y}^{(i)}, \text{ for } i = 1, \ldots, N$$

It is clear that $\tilde{y}^{(i)} = y^{(i)}$ iff $\lambda_j = 0$ for $k < j < M$

Otherwise: Loss of information

Additional problems:

- computational complexity is: $\mathcal{O}(M^3)$ for eigenvalue decomposition + $\mathcal{O}(NM^2)$ for calculation of covariance matrix
  $\rightarrow$ numerically expensive for large M (dimension of data space)
- number of principal components (and hence possible features) is bounded by M
  example: $x \in \mathbb{R}^2 \Rightarrow$ max. two principal components
- (linear) PCA does not allow for extraction of non-linear features

Conclusions and Outlook

- PCA is a good tool for dimension reduction and feature extraction
- can be used for clustering
- can be interpreted as **linear transformation** of input data to a feature space
- computational complexity mainly depends on dimension M of data space
- PCA is restricted to linear features

Thank you for your attention!