Lecture 5: Linear Subspace Projections: Principal Component Analysis

Machine Learning, Summer Term 2019

Michael Tangermann Frank Hutter Marius Lindauer

University of Freiburg

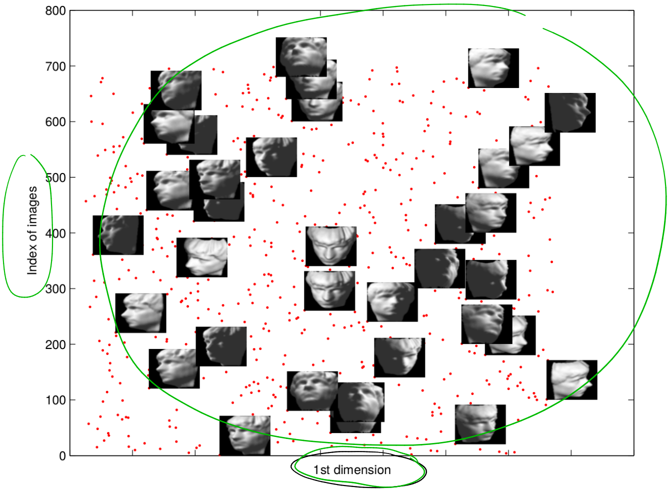
Lecture Overview

1. Motivation
2. The PCA Transformation
3. Wrapup: Related Topics, Summary, Preview

Lecture Overview

Example: Images of Faces

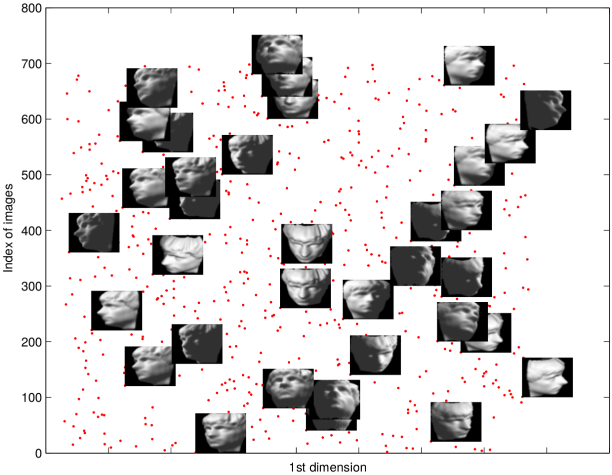
(Plots by Ali Ghodsi, "Dimensionality Reduction, A Short Tutorial", 2006)

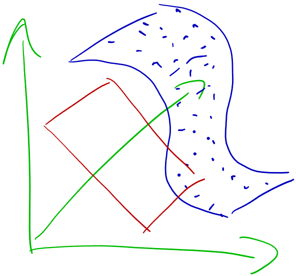


Varying pose and and lighting conditions, 698 images (64x64 pixels) of the same face were generated. Dimensionality $D = 4096$

Example: Images of Faces

(Plots by Ali Ghodsi, "Dimensionality Reduction, A Short Tutorial", 2006)





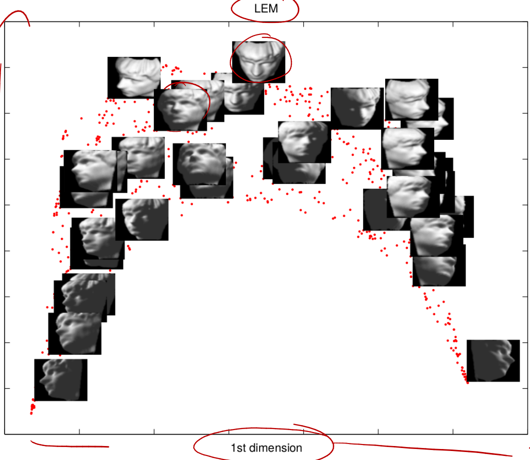
Varying pose and and lighting conditions, 698 images (64x64 pixels) of the same face were generated. Dimensionality $D = 4096$ **really?**

Example: Images of Faces

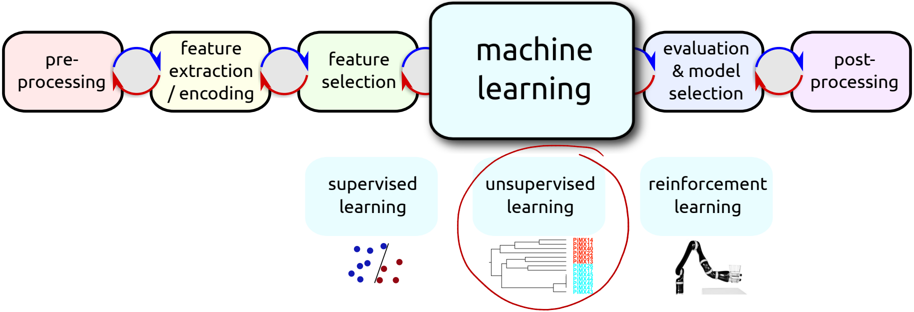
$\rightarrow$ As we know, how this data has been generated, we can expect that all of the images live on an embedded, **smaller-dimensional manifold**, which is expressed by lighting conditions and viewing angle.

Example: Images of Faces

→ As we know, how this data has been generated, we can expect that all of the images live on an embedded, **smaller-dimensional manifold**, which is expressed by lighting conditions and viewing angle.



ML Design Cycle



Today's topic is how to project data to a useful subspace with unsupervised principal component analysis (PCA):

- Labels are not required

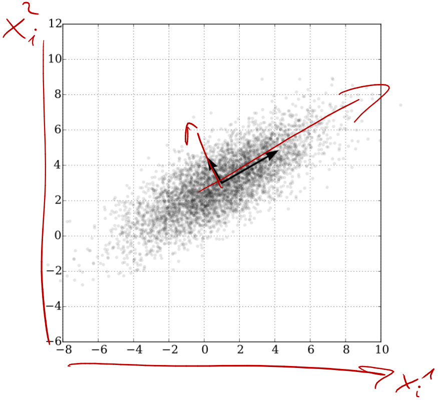
Typical Application of PCA: Dimensionality Reduction

Given:

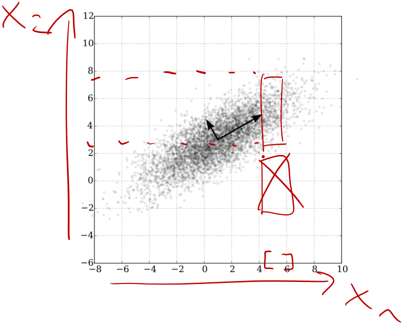
- N high dimensional data points $\mathbf{x}_i \in \mathbb{R}^D$ with $i = 1 \ldots N$.
- Data is collected in matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$

Scatter plot:

$D = 2$



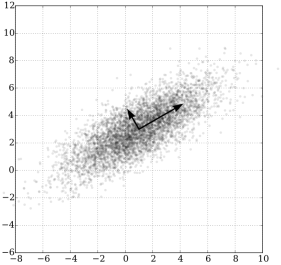
Typical Application of PCA: Dimensionality Reduction



Let's assume, that

- the input dimensions are correlated

Typical Application of PCA: Dimensionality Reduction

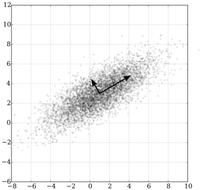


Let's assume, that

- the input dimensions are correlated
- some later method in the pipeline is *extremely* slow on high dimensional data...

What do you propose to do?

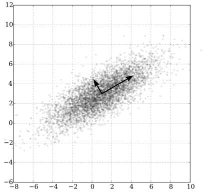
Simple Example: Dimensionality Reduction



Key ideas:

- Let's try to determine a subspace $\mathbb{R}^M$ of $\mathbb{R}^D$, with $M \leqslant D$.
- The subspace should contain the relevant part of our data.

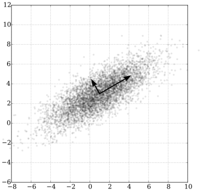
Simple Example: Dimensionality Reduction



Key ideas:

- Let's try to determine a subspace $\mathbb{R}^M$ of $\mathbb{R}^D$, with $M < D$.
- The subspace should contain the relevant part of our data.
- Let's choose the new dimensions of this projected subspace such, that the data is uncorrelated.
- The subspace can be defined by a projection.

Simple Example: Dimensionality Reduction

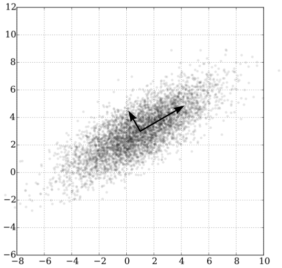


Key ideas:

- Let's try to determine a subspace $\mathbb{R}^M$ of $\mathbb{R}^D$, with $M < D$.
- The subspace should contain the relevant part of our data.
- Let's choose the new dimensions of this projected subspace such, that the data is uncorrelated.
- The subspace can be defined by a projection.

 You have seen projections before - in which context?

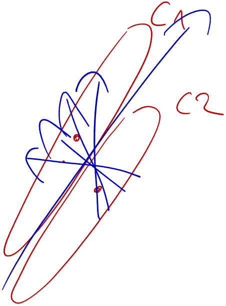
Simple Example: Dimensionality Reduction

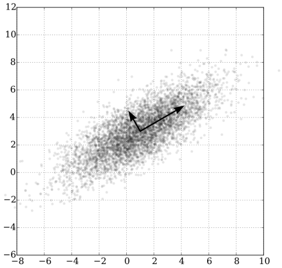


Principal component analysis (PCA) can be applied in such a scenario:

- PCA determines a linear subspace
- PCA makes the (somewhat strong!) assumption, that relevance is expressed by variance!

Simple Example: Dimensionality Reduction

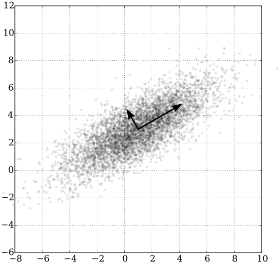




Principal component analysis (PCA) can be applied in such a scenario:

- PCA determines a linear subspace
- PCA makes the (somewhat strong!) assumption, that relevance is expressed by variance! (**Problematic?** ✋🏾✋🏼)

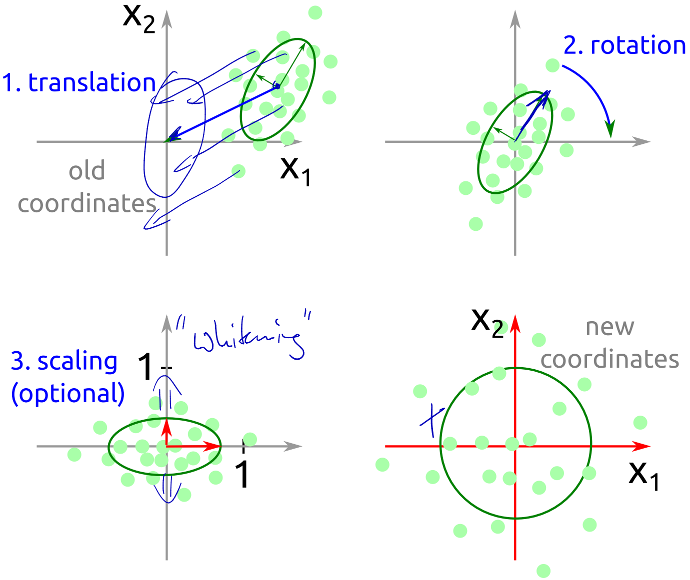
Simple Example: Dimensionality Reduction



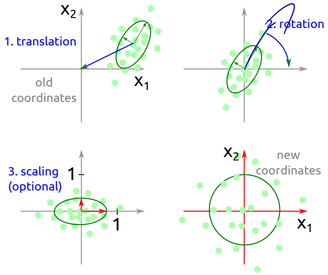
Principal component analysis (PCA) can be applied in such a scenario:

- PCA determines a linear subspace
- PCA makes the (somewhat strong!) assumption, that relevance is expressed by variance! (**Problematic?** 🖐🏾🖐🏼)
- → Can we reduce our data to the subspace with highest variance?

Intuition of PCA



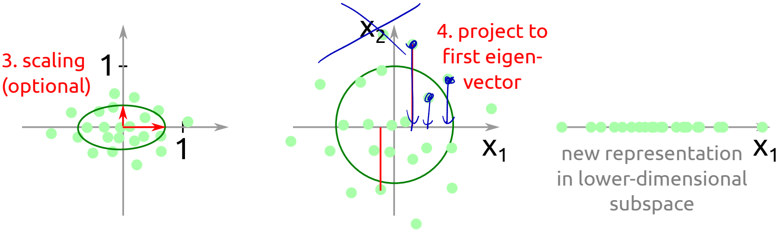
Intuition of PCA

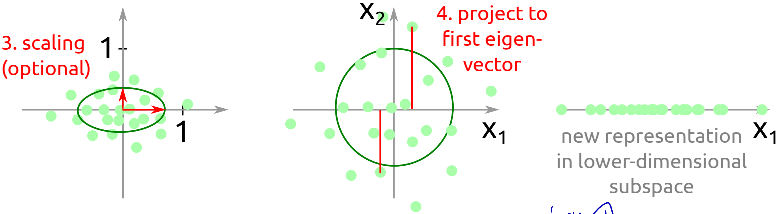


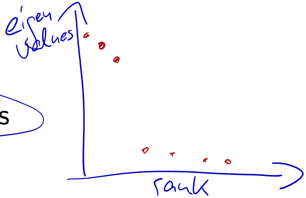
Principal component analysis (PCA) performs the following steps:

- Translation of data $\mathbf{X}$ to the origin
- Rotation, such that eigenvectors of $\mathbf{X}$ form the new axes
- (optional:) Scale the projected data according to the eigenvalues

Intuition of PCA for Dimensionality Reduction







Dimensionality reduction with PCA:

- Project data only to the first M eigenvectors (sorted by strongest eigenvalues)

Assume we have found a lower-dimensional representation with PCA.

Will we be able to save measuring time in the future, as we don't need to measure all variables?

Lecture Overview

Derive Projections by Maximizing the Variance

There are many ways to formulate and solve the PCA problem.
Let's follow the idea to maximize the variance!

Derive Projections by Maximizing the Variance

There are many ways to formulate and solve the PCA problem.
Let's follow the idea to maximize the variance!

Consider a projection to a 1-dimensional subspace $(M = 1)$, defined by the direction of a D-dimensional unit vector $\mathbf{u}_1$.

Derive Projections by Maximizing the Variance

There are many ways to formulate and solve the PCA problem.
Let's follow the idea to maximize the variance!

Consider a projection to a 1-dimensional subspace ($M = 1$), defined by the direction of a D-dimensional unit vector $\mathbf{u}_1$.

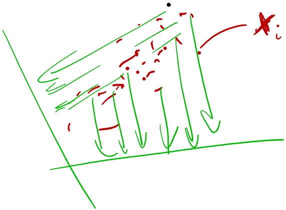
- Unit vector characteristic: $\mathbf{u}_1^T \mathbf{u}_1 = 1$

Derive Projections by Maximizing the Variance

There are many ways to formulate and solve the PCA problem.
Let's follow the idea to maximize the variance!

Consider a projection to a 1-dimensional subspace ($M = 1$), defined by the direction of a D-dimensional unit vector $\mathbf{u}_1$.

- Unit vector characteristic: $\mathbf{u}_1^T \mathbf{u}_1 = 1$
- Then each data point can be projected to a scalar value via $\mathbf{u}_1^T \mathbf{x}$.



Derive Projections by Maximizing the Variance

There are many ways to formulate and solve the PCA problem.
Let's follow the idea to maximize the variance!

Consider a projection to a 1-dimensional subspace ($M = 1$), defined by the direction of a D-dimensional unit vector $\mathbf{u}_1$.

- Unit vector characteristic: $\mathbf{u}_1^T \mathbf{u}_1 = 1$
- Then each data point can be projected to a scalar value via $\mathbf{u}_1^T \mathbf{x}$.

The mean of projected data is thus given by: $\mathbf{u}_1^T \overline{\mathbf{x}}$, with $\overline{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n$.

Derive Projections by Maximizing the Variance

There are many ways to formulate and solve the PCA problem.
Let's follow the idea to maximize the variance!

Consider a projection to a 1-dimensional subspace $(M = 1)$, defined by the direction of a D-dimensional unit vector $\mathbf{u}_1$.

- Unit vector characteristic: $\mathbf{u}_1^T \mathbf{u}_1 = 1$
- Then each data point can be projected to a scalar value via $\mathbf{u}_1^T \mathbf{x}$.

The mean of projected data is thus given by: $\mathbf{u}_1^T \overline{\mathbf{x}}$, with $\overline{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n$.

We want $\mathbf{u}_1$ to maximize the variance of the projected data:

$$\underset{\mathbf{u}_1}{\operatorname{argmax}} \sum_{n=1}^{N} \{\mathbf{u}_1^T \mathbf{x}_n - \mathbf{u}_1^T \overline{\mathbf{x}}\}^2 = \underset{\mathbf{u}_1}{\operatorname{argmax}} \; \mathbf{u}_1^T \mathbf{S} \mathbf{u}_1$$

with $\mathbf{S}$ being the data covariance matrix.

The Principal Component Transformation

Attention: maximizing $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1$ with respect to $\mathbf{u}_1$ requires a constraint, otherwise $\mathbf{u}_1$ would simply grow to infinity...:

$u_1^T S u_1$

$100 \cdot u_1^T S \, 100 \cdot u_1$

The Principal Component Transformation

Attention: maximizing $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1$ with respect to $\mathbf{u}_1$ requires a constraint, otherwise $\mathbf{u}_1$ would simply grow to infinity...:

- Conveniently, we chose the constraint $\mathbf{u}_1^T \mathbf{u}_1 = 1$
- We can enforce it by introducing a so-called Lagrange multiplier λ_1

The Principal Component Transformation

Attention: maximizing $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1$ with respect to $\mathbf{u}_1$ requires a constraint, otherwise $\mathbf{u}_1$ would simply grow to infinity...:

- Conveniently, we chose the constraint $\mathbf{u}_1^T \mathbf{u}_1 = 1$
- We can enforce it by introducing a so-called Lagrange multiplier λ_1

This leads to the following maximization problem:

$$\underset{\mathbf{u}_1, \lambda_1}{\operatorname{argmax}} \quad \underline{\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1} + \lambda_1 (1 - \mathbf{u}_1^T \mathbf{u}_1)$$

The Principal Component Transformation

Attention: maximizing $\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1$ with respect to $\mathbf{u}_1$ requires a constraint, otherwise $\mathbf{u}_1$ would simply grow to infinity...:

- Conveniently, we chose the constraint $\mathbf{u}_1^T\mathbf{u}_1 = 1$
- We can enforce it by introducing a so-called Lagrange multiplier λ_1

This leads to the following maximization problem:

$$\underset{\mathbf{u}_1,\lambda_1}{\operatorname{argmax}} \quad \cancel{\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1} + \lambda_1(1 - \mathbf{u}_1^T\mathbf{u}_1)$$

Setting the derivative with respect to λ_1 to zero, we obtain

$$1 = \mathbf{u}_1^T\mathbf{u}_1$$

The Principal Component Transformation

Attention: maximizing $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1$ with respect to $\mathbf{u}_1$ requires a constraint, otherwise $\mathbf{u}_1$ would simply grow to infinity...:

- Conveniently, we chose the constraint $\mathbf{u}_1^T \mathbf{u}_1 = 1$
- We can enforce it by introducing a so-called Lagrange multiplier λ_1

This leads to the following maximization problem:

$$\underset{\mathbf{u}_1, \lambda_1}{\operatorname{argmax}} \; \cancel{\mathbf{u}_1^T} \mathbf{S} \mathbf{u}_1 + \lambda_1 (1 - \cancel{\mathbf{u}_1^T} \mathbf{u}_1)$$

Setting the derivative with respect to λ_1 to zero, we obtain

$$1 = \mathbf{u}_1^T \mathbf{u}_1$$

Setting the derivative with respect to $\mathbf{u}_1$ to zero, we obtain

$$0 = \underline{\mathbf{S} \mathbf{u}_1} - \underline{\lambda_1 \mathbf{u}_1}$$

The Principal Component Transformation

linear transform

Let's re-write this to

$$\mathbf{S}\mathbf{u}_1 = \lambda_1 \mathbf{u}_1$$

The Principal Component Transformation

Let's re-write this to

$$\mathbf{S}\mathbf{u}_1 = \lambda_1 \mathbf{u}_1$$

This means, that the variance is maximal, if $\mathbf{u}_1$ is an eigenvector of the covariance matrix $\mathbf{S}$.

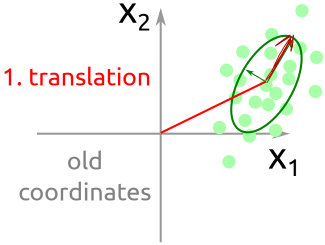
The Principal Component Transformation

Let's re-write this to

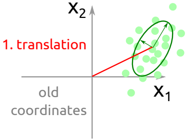
$$\mathbf{S}\mathbf{u}_1 = \lambda_1 \mathbf{u}_1$$

This means, that the variance is maximal, if $\mathbf{u}_1$ is an eigenvector of the covariance matrix $\mathbf{S}$.

This meets our intuition, compare

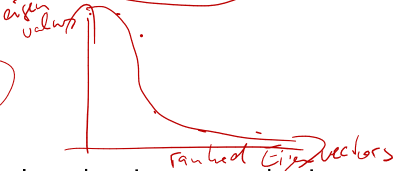


The Principal Component Transformation



$$\mathbf{S}\mathbf{u}_1 = \lambda_1 \mathbf{u}_1$$

Multiplying with $\mathbf{u}_1^T$ from the left and making use of $\mathbf{u}_1^T\mathbf{u}_1 = 1$, variance is given by:



$$\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1 = \lambda_1$$

Observe:

- Variance is maximized, if we set $\mathbf{u}_1$ equal to the eigenvector having the largest eigenvalue λ_1.

Obtaining More Than One Projection Direction

We have seen how to obtain the first projection direction via $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 = \lambda_1$.

Any idea how to find the next one(s)?

Obtaining More Than One Projection Direction

We have seen how to obtain the first projection direction via $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 = \lambda_1$.

Any idea how to find the next one(s)?

Further projection directions can be obtained by iterating the procedure, making sure that the following eigenvector is orthogonal to the ones obtained so far.

- Delivers a set of M eigen~~values~~ vectors $\mathbf{u}_1, \ldots, \mathbf{u}_M$
- Eigenvalues can be sorted according to the eigenvalues $\lambda_1, \ldots, \lambda_m$

Comment:
consider the spectrum of eigenvalues
to determine a suitable value of M

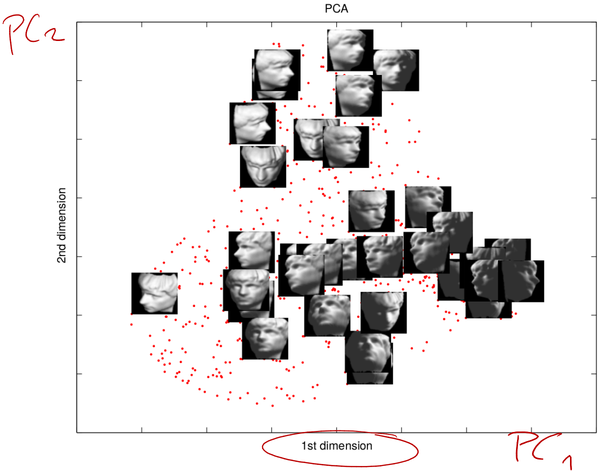
Lecture Overview

Typical Use of Principal Component Analysis

- Data compression / dimensionality reduction (if you think, that variance matters!)

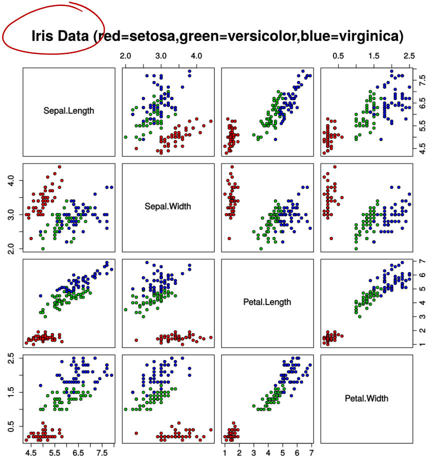
Typical Use of Principal Component Analysis

- Data compression / dimensionality reduction (if you think, that variance matters!)



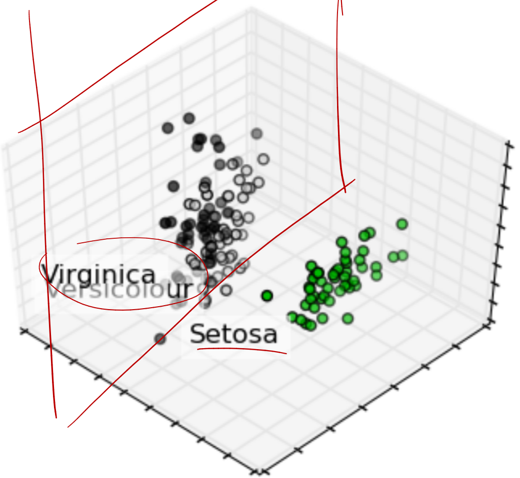
Typical Use of Principal Component Analysis

- Data compression / dimensionality reduction
- Data visualization using a projection onto $M = 2$ or $M = 3$



Typical Use of Principal Component Analysis

- Data compression / dimensionality reduction
- Data visualization using a projection onto $M = 2$ or $M = 3$



Alternative Names and Formulations for PCA

Depending on the context, principal component analysis is also referred to as

- Linear algebra: singular value decomposition SVD
- Linear algebra: eigenvalue decomposition EVD
- Image processing, control theory: Hotelling transform
- Karhunen-Loève transform
- ...

Alternative Names and Formulations for PCA

Depending on the context, principal component analysis is also referred to as

- Linear algebra: singular value decomposition SVD
- Linear algebra: eigenvalue decomposition EVD
- Image processing, control theory: Hotelling transform
- Karhunen-Loève transform
- ...

Comments:

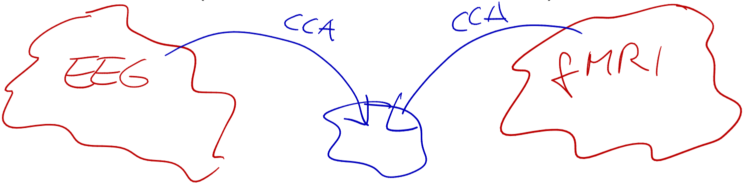
- There are many algorithmic approaches to derive projection directions (Raleigh coefficient, SVD, Bayesian PCA, iterative vs. analytical, ...)
- Using the covariance matrix has disadvantages, if dimensionality D is large.

Further Reading for PCA

- Section 12.1 of Bishop's book was mostly used for these slides
- Wikipedia.org on PCA for a great top-down overview!

Related Subspace Methods

- Whitening / sphereing: transform data to zero mean and unit covariance as a common preprocessing step
- Factor analysis FA (incorporate domain-specific assumptions)
- Canonical correlation analysis CCA (relate two data sources to a common subspace which maximizes cross-covariance)
- Kernel-PCA (non-linear extension of PCA)



Summary by learning goals

Having heard this lecture, you can now . . .

- explain, what PCA is doing
- explain, how the novel basis vectors are obtained
- program an iterative version of the algorithm
- formulate assumptions made by PCA (e.g. what happens, if we forget the translation to the origin?)
- name typical use cases of PCA
- (assignments) explain the role and benefits of whitening
- (assignments) explain typical pitfalls related to the use of PCA