Advanced Machine Learning for KCS
Lecture 3.2: PCA

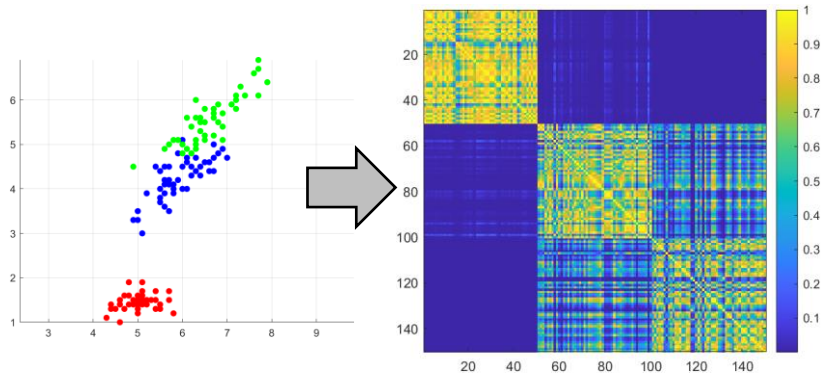
10.09.2023

Stella

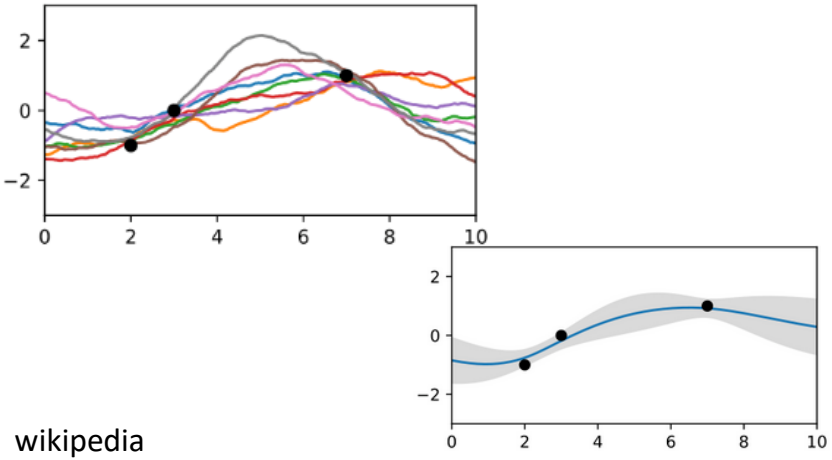
Department of Computer Science

IT UNIVERSITY OF COPENHAGEN

Recap - Kernel Methods



- **kernel** $k(\mathbf{x}_n, \mathbf{x}_m)$ represents similarity between samples
- no need for function $\phi(\mathbf{x})$
- different modalities, data representation possible



wikipedia

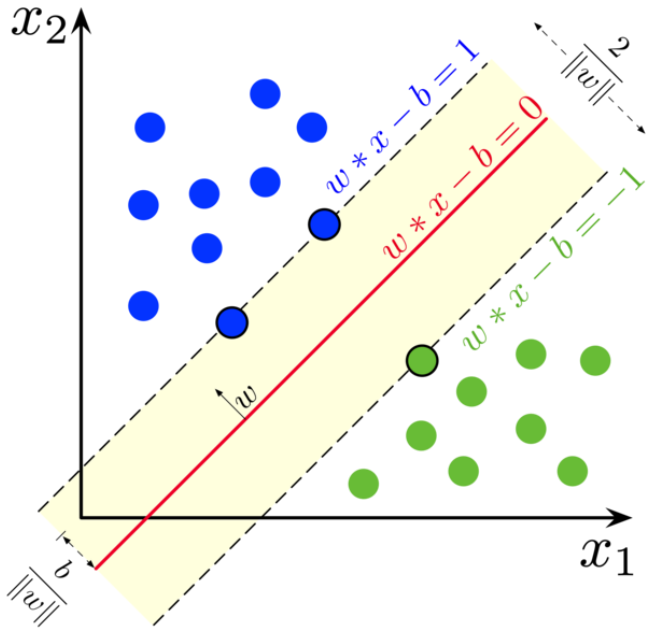
Gaussian Processes:

- model distribution of target variable directly
- uncertainty

$$p(\boldsymbol{t}) = \mathcal{N}(\boldsymbol{t}|\mathbf{0}, \boldsymbol{C})$$

$$p(t_{N+1}|\boldsymbol{t}) = \mathcal{N}(m, \sigma^2)$$

Recap - SVMs



$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}) + w_0$$

Pros:

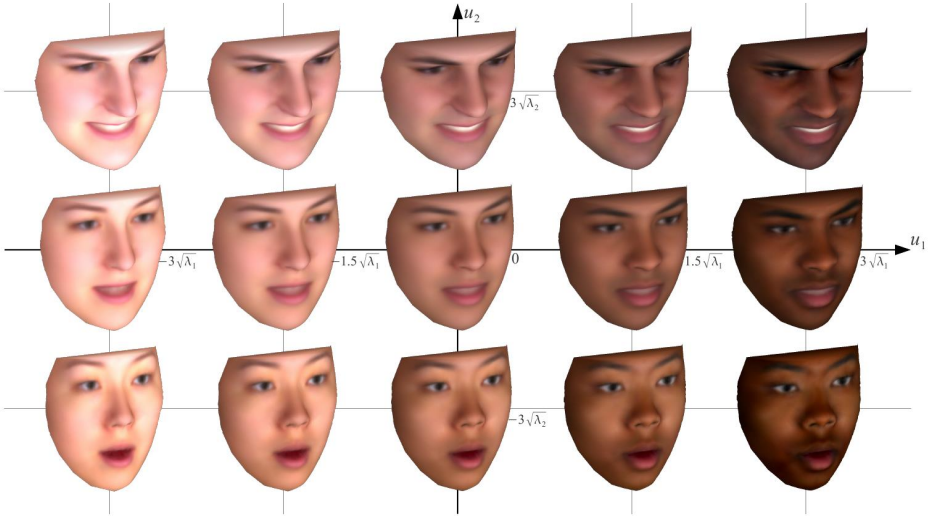
- light-weight model
- less likely to overfit
- Kernel-trick
- works well in higher dimensions

Cons:

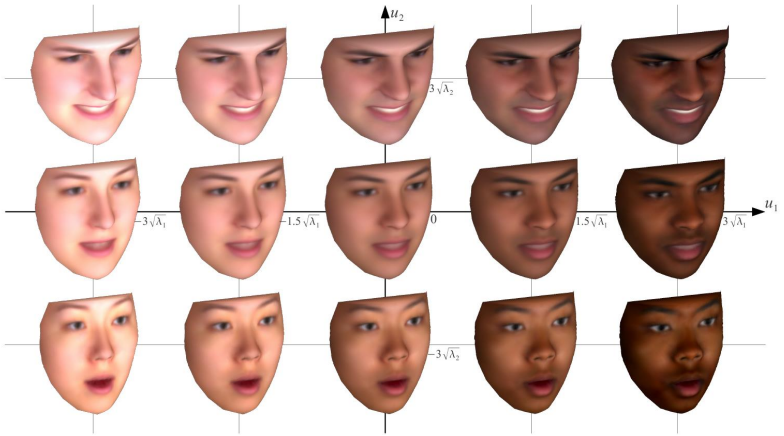
- Not suitable on large datasets
- Kernel is difficult to choose
- Only class decision, i.e. discriminative
  - No class probabilities
  - Not generative

ILOs - today

- Definition of Correlation
- Define PCA
- Apply PCA for:
  - Dimensionality reduction
  - Generating samples
  - regression



Outline

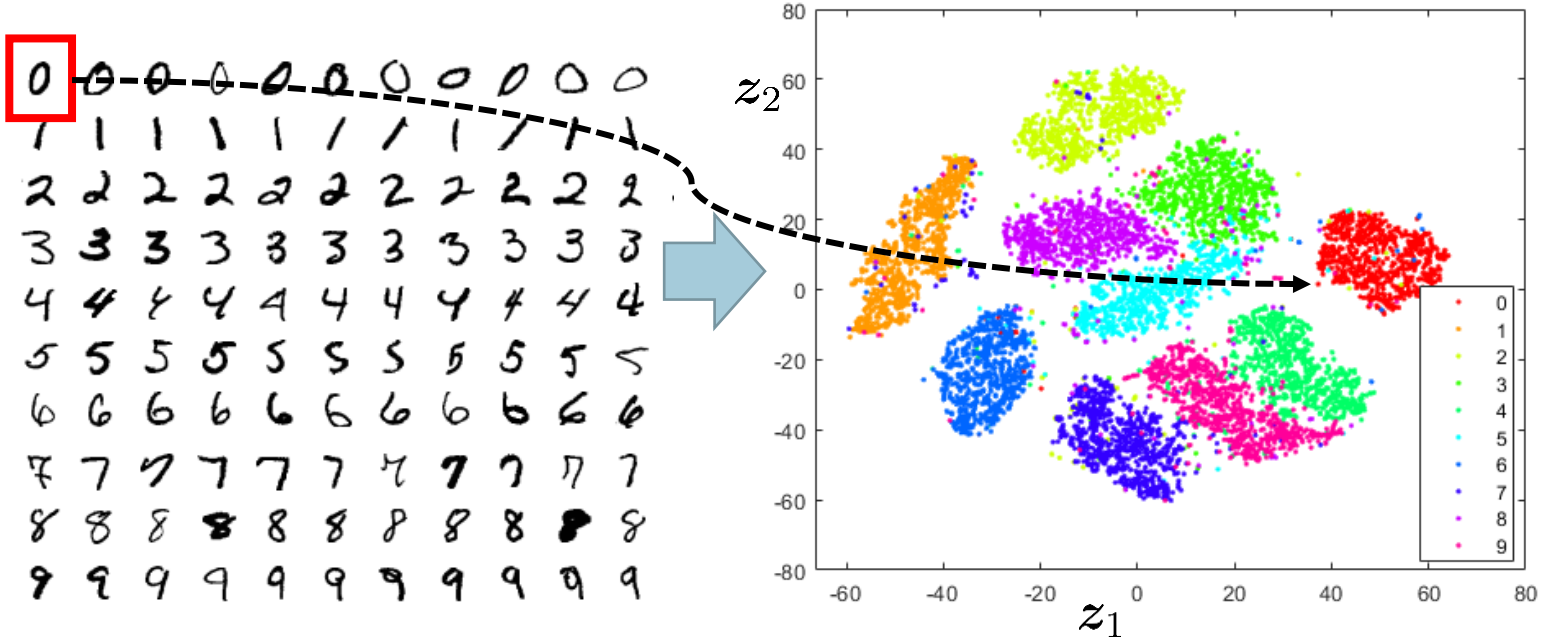


- Recap:
  - Correlation: good vs. bad
  - Eigenvectors
- Principal Component Analysis
- Applications
- Extensions
- Summary

Feature Extraction – Dimensionality Reduction

For high-dimensional data, we want: $x_n \in \mathbb{R}^{28 \cdot 28} \mapsto z_n \in \mathbb{R}^2$

- Compact representation of data, e.g. image $x_n \in \mathbb{R}^{28 \cdot 28}$
- Extract **most relevant** information $z_n \in \mathbb{R}^2$



Sources:

[1] MNIST, wikpedia.org, Josef Steppan [CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0)]

[2] MathWorks, https://se.mathworks.com/help/stats/visualize-high-dimensional-data-using-t-sne.html

Correlation

- How much two variables are “linearly” related
- Neither good or bad, i.e. depends on application

$\hat{\sigma}$ estimator for σ

$\mathbb{E}(\hat{\sigma}) = \sigma$

(x_i, y_i) samples

mean of all y_i

$$\rho = \frac{\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y})}{(N-1)s_x s_y}$$

biased

deviation of y

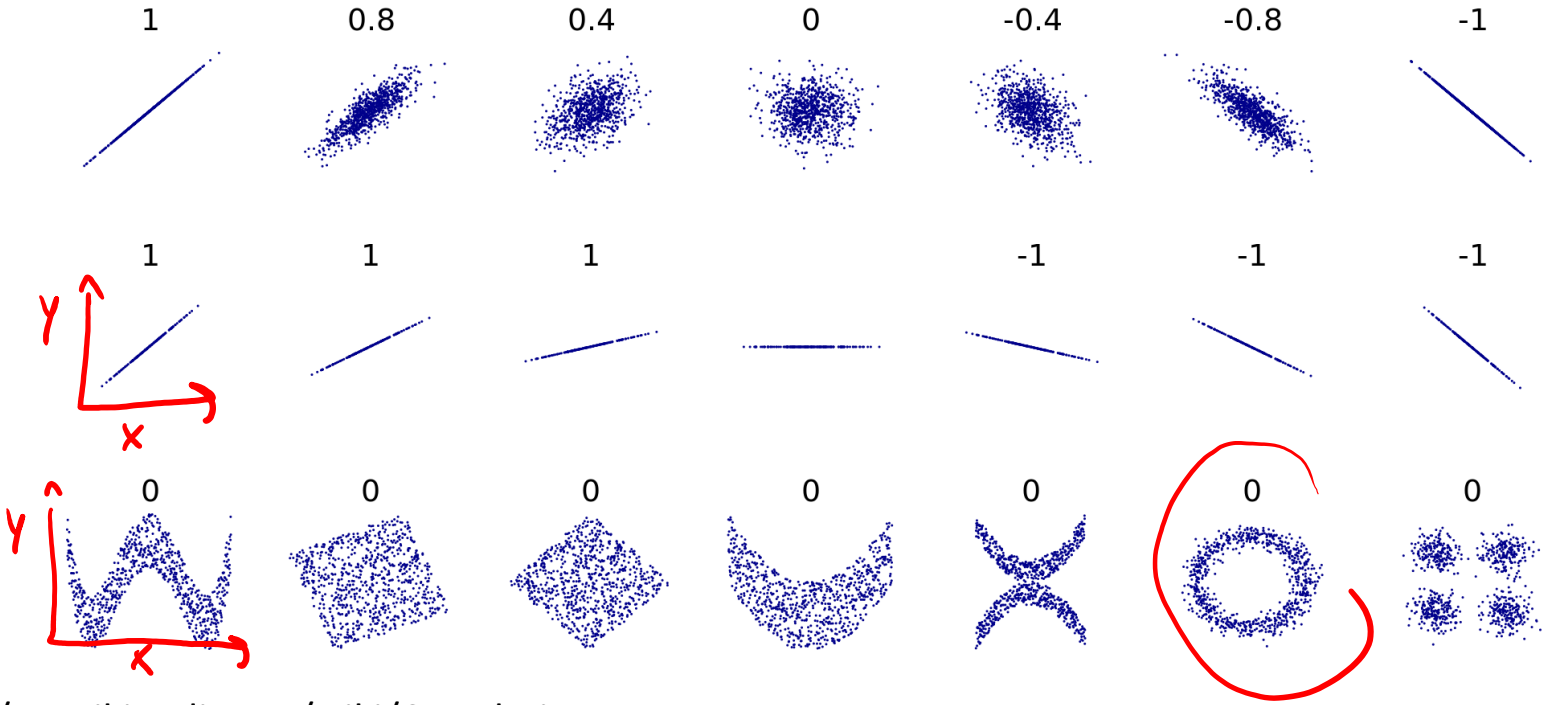
— “ — x

$\frac{1}{N}$ vs. $\frac{1}{N-1}$

difference variance implementations

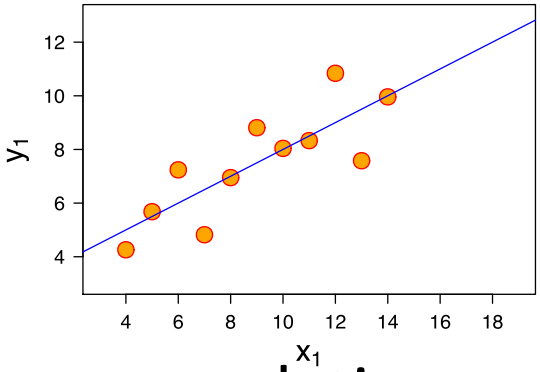
Correlation

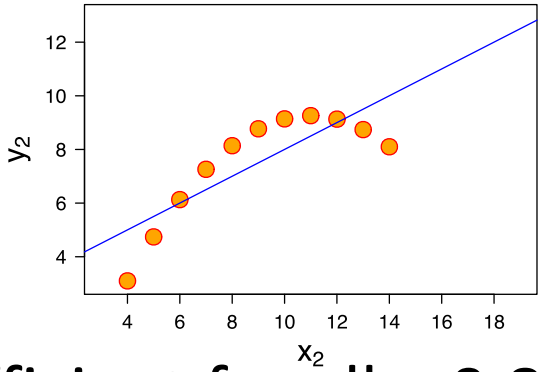
- How much two variables are “linearly” related
- Neither good or bad, i.e. depends on application



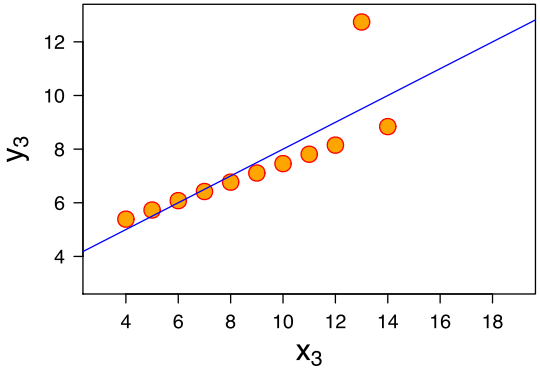
https://en.wikipedia.org/wiki/Correlation

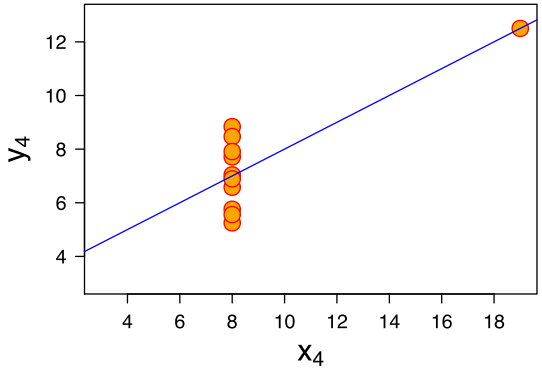
Which examples has the highest correlation?





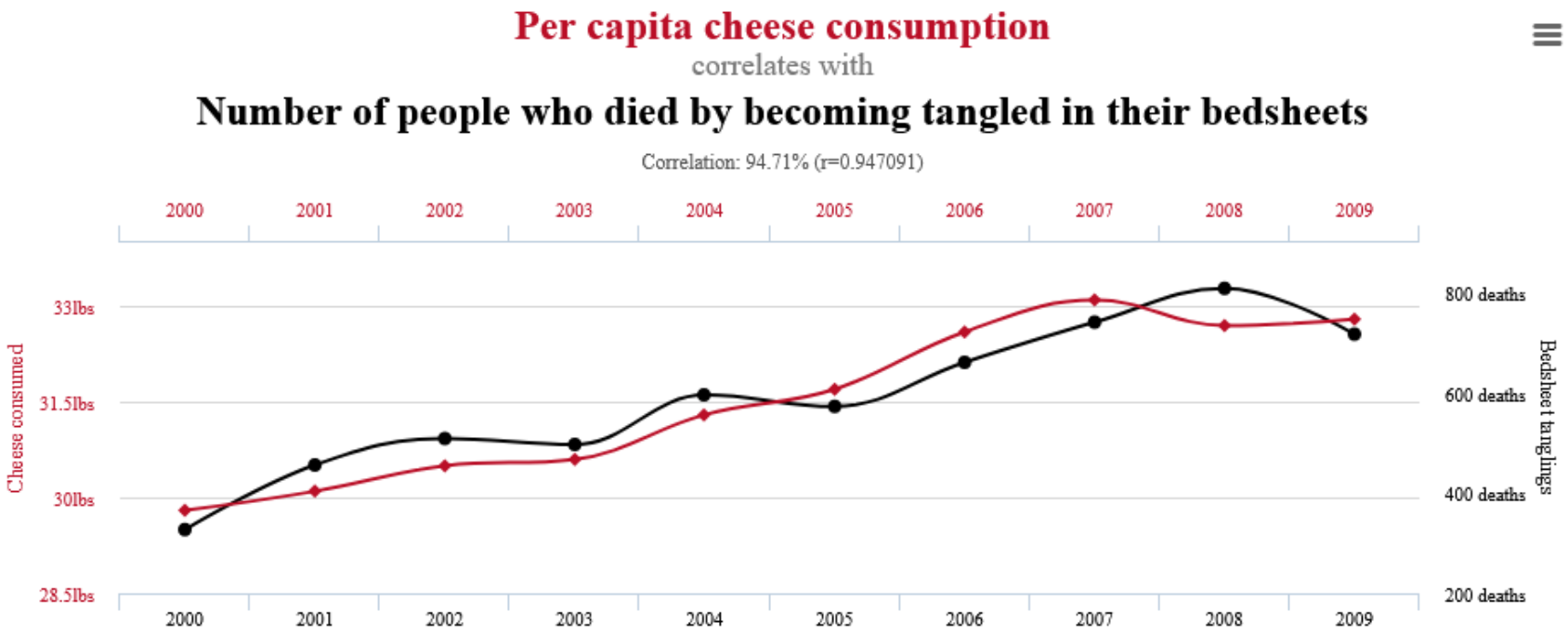
same correlation coefficient for all = 0.816





https://en.wikipedia.org/wiki/Correlation

Spurious Correlations



https://www.tylervigen.com/spurious-correlations

Correlation

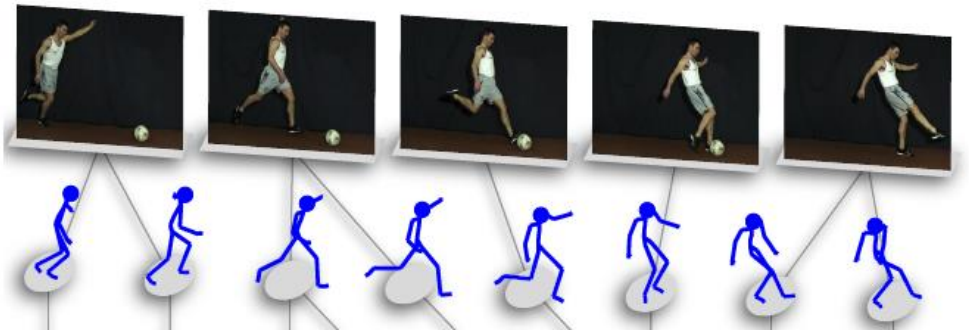
High correlation

- Good - if you want redundancy or find something common
- Bad - if you want disentangled data

Feature Extraction vs Feature Selection

Good correlation: Canonical Correlation Analysis (CCA)

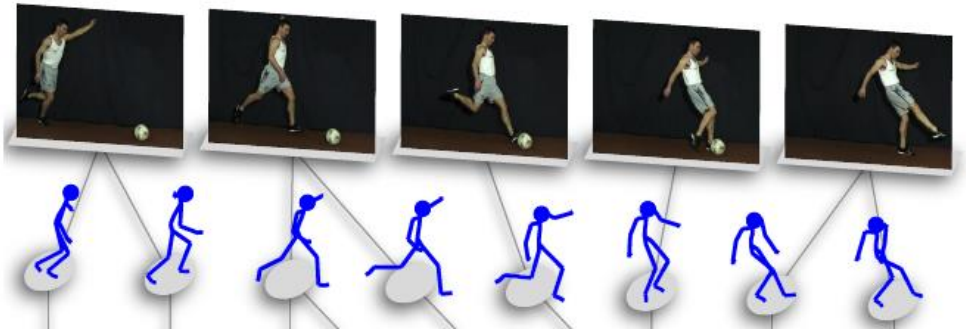
Idea: find maximum correlation between two signals



- Data has T frames
  - Video (image sequence)
    D = Tx100x100x3 [width x height x channel]
  - 3D points
    E = Tx15x3
- **Problem:** dimensions differ
- **Goal:** Reduce the data to one **joint dimension**

„Generalized Canonical Time Warping“, Zhou et al., 2013

Good correlation: Canonical Correlation Analysis (CCA)



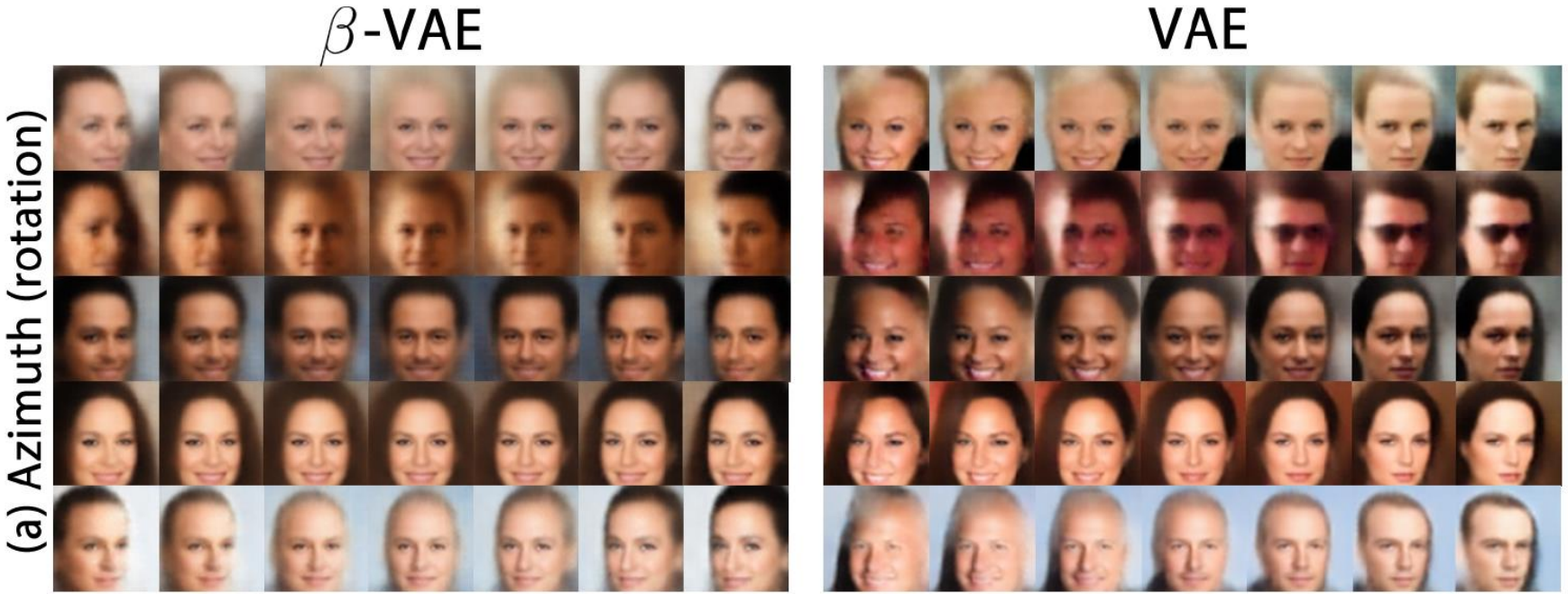
Given data: Find two projections
maximize correlation between projections

$$\rho = \mathrm{Corr}\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}, \boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}\right) = \frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_{xy}\boldsymbol{v}}{\sqrt{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_{xx}\boldsymbol{w}}\sqrt{\boldsymbol{v}^{\mathrm{T}}\boldsymbol{S}_{yy}\boldsymbol{v}}} \qquad \begin{matrix}\boldsymbol{w}, \boldsymbol{x} \in \mathbb{R}^{D} \\ \boldsymbol{v}, \boldsymbol{y} \in \mathbb{R}^{E}\end{matrix}$$

„Generalized Canonical Time Warping“, Zhou et al., 2013

Bad correlation:

Directions in VAE are entangled
beta VAE is better

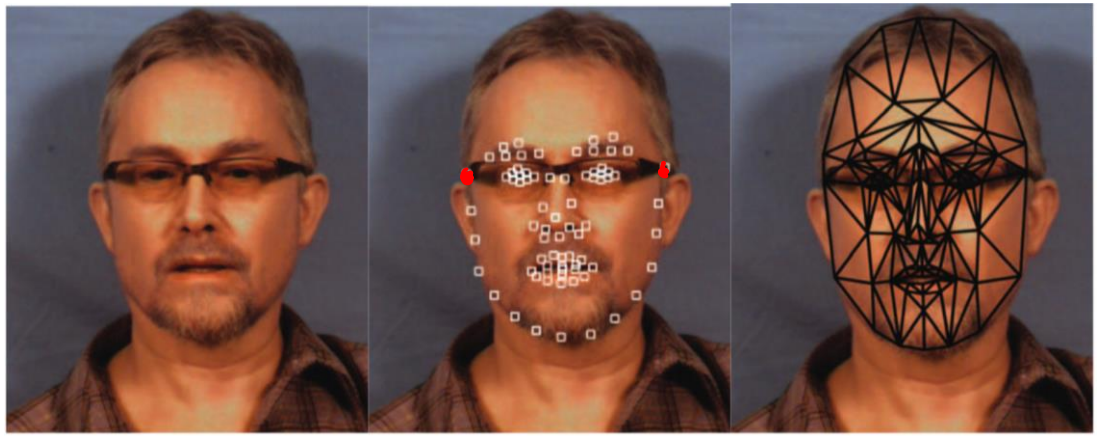


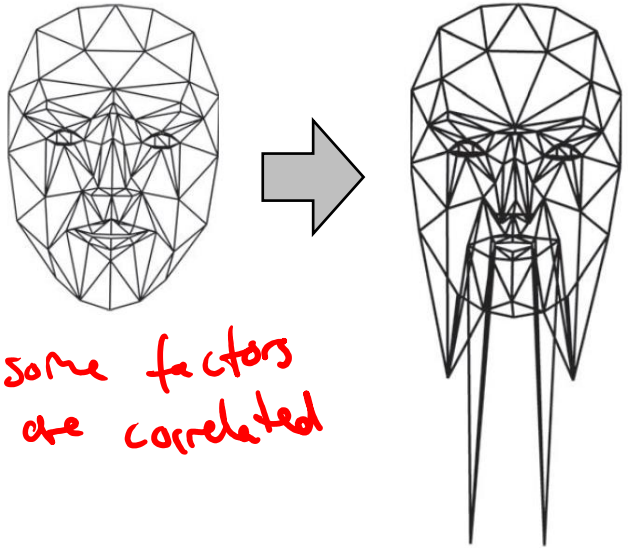
Estimate correlated parameters

How to use a 3D face model for 3D reconstruction from 2D landmarks

$\mathbb{R}^{3N} \ni \mathbf{f} = \bar{\mathbf{f}} + \mathbf{M}\mathbf{p}$

$\begin{bmatrix} x \\ y \\ z \end{bmatrix}$ ← delete z ⇒ $f_{2D} = \bar{f}_{2D} + \tilde{M}_p$





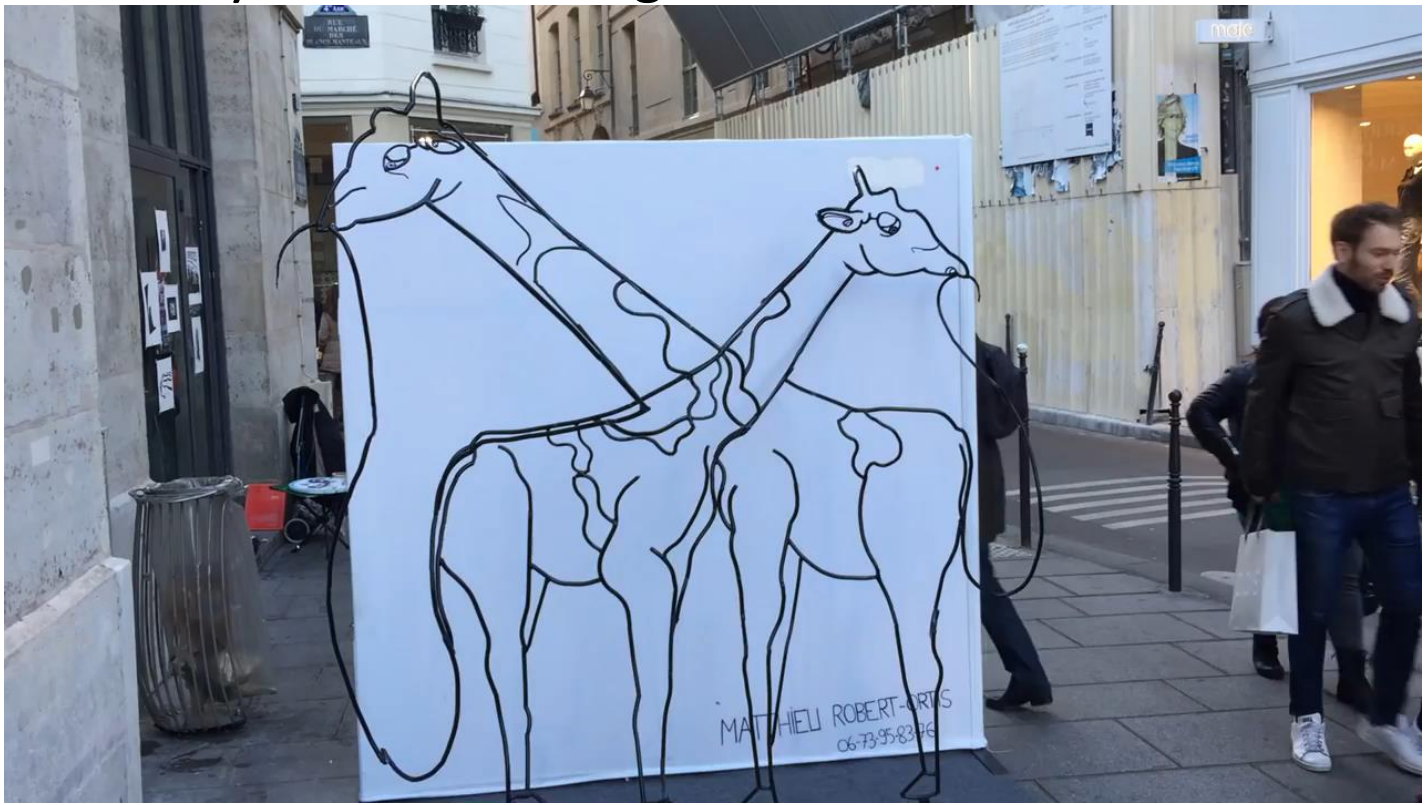
some factors are correlated

=> How to de-correlate?

Grasshof, et al. 'Estimation of Face Parameters Using Correlation Analysis and a Topology Preserving Prior'. 2015

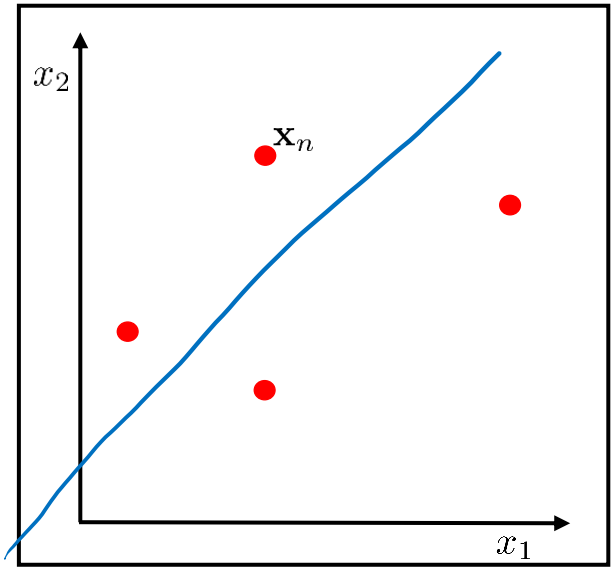
Motivation

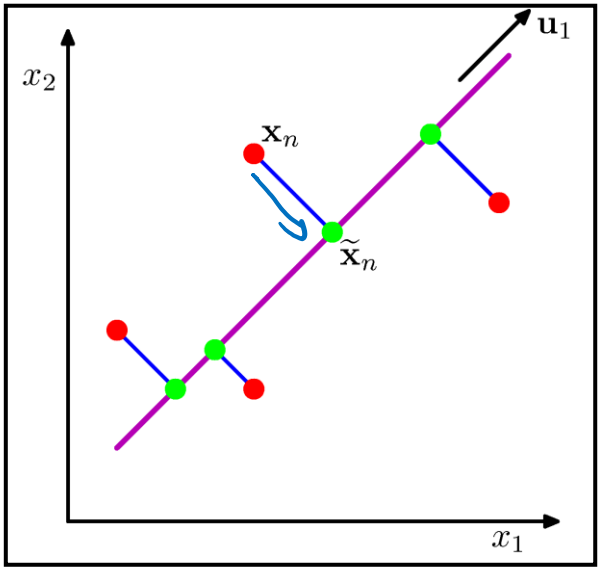
Why should rotating the data tell me more?



https://www.youtube.com/watch?v=PiYMol0VjWo

Principal Component Analysis (PCA)



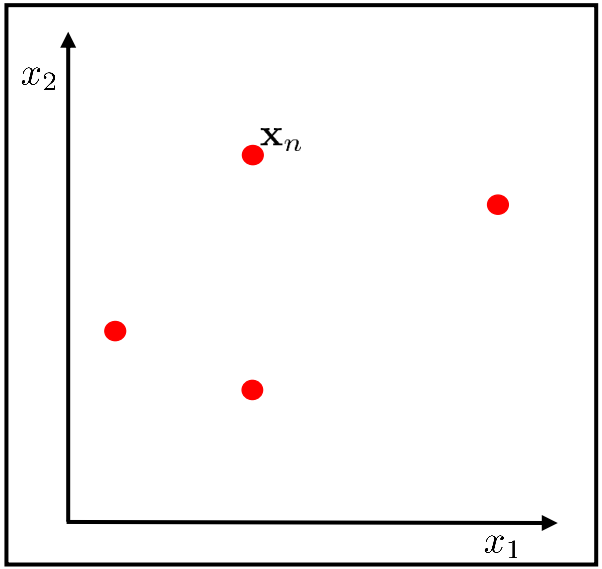


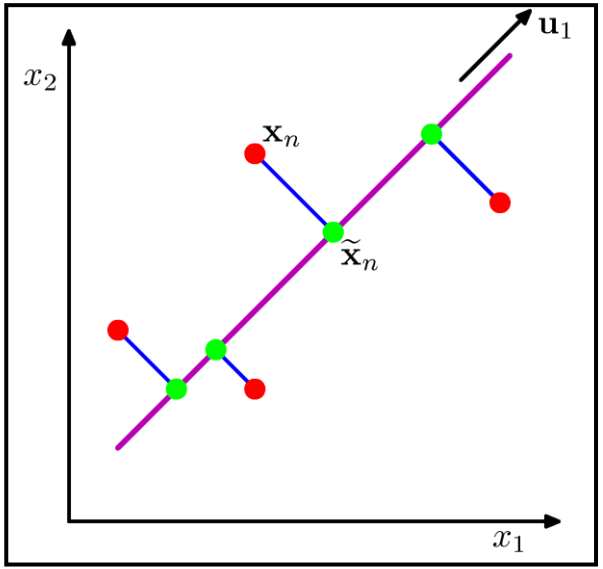
Find projection which

- **Maximizes** variance
- **Minimizes** reprojection error

- Find a low-dimensional space such that:
  when ***x*** is projected there, "information loss" is minimized
- new directions must be **uncorrelated**, i.e.
  covariance matrix is diagonal

Principal Component Analysis (PCA)





Find projection which

- Maximizes variance
- Minimizes reprojection error

PCA – how to compute the principal components

data $\vec{x} \in \mathbb{R}^2$, principal component $\vec{u}_1 \in \mathbb{R}^2$, $z \in \mathbb{R}$,

$z = \vec{x}^T \vec{u}_1$ ↓ deterministic

$\mathbb{V}(z) = \mathbb{V}(\vec{x}^T \vec{u}_1) = \vec{u}_1^T \mathbb{V}(x) \vec{u}_1$ ← covariance matrix

$\mathbb{V}(x) = S = S^T$

[1×D] [D×D] [D×1]

max variance under conditions:

(1) $\|\vec{u}_1\|_2 = \vec{u}_1^T \vec{u}_1 = 1$

(2) $\vec{u}_1^T \vec{u}_2 = 0$

$\Rightarrow \max_{\vec{u}_1} \mathbb{V}(z) + \lambda (1 - \vec{u}_1^T \vec{u}_1)$

$f(\vec{u}_1) = \vec{u}_1^T S \vec{u}_1 + \lambda - \lambda \vec{u}_1^T \vec{u}_1$

← derivative

$\nabla f(\vec{u}_1) = 2 S \vec{u}_1 - 2\lambda \vec{u}_1 \stackrel{!}{=} 0$

$S \vec{u}_1 = \lambda \vec{u}_1$

↑ first principal component

$\mathbb{V}(aX) = a^2 \mathbb{V}(X)$

↑ random variable

↳ scalar, deterministic

estimate for cov. matrix

$S \approx \frac{1}{N-1} X^T X$

↑ data matrix

Eigenvectors and Eigenvalues

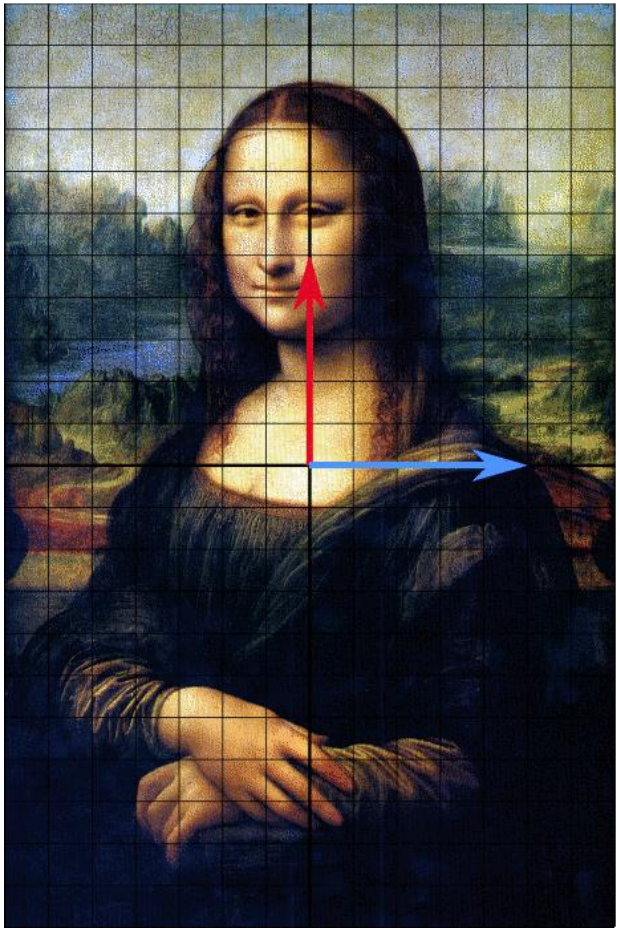
Definition of an eigenvector and eigenvalue of matrix A:

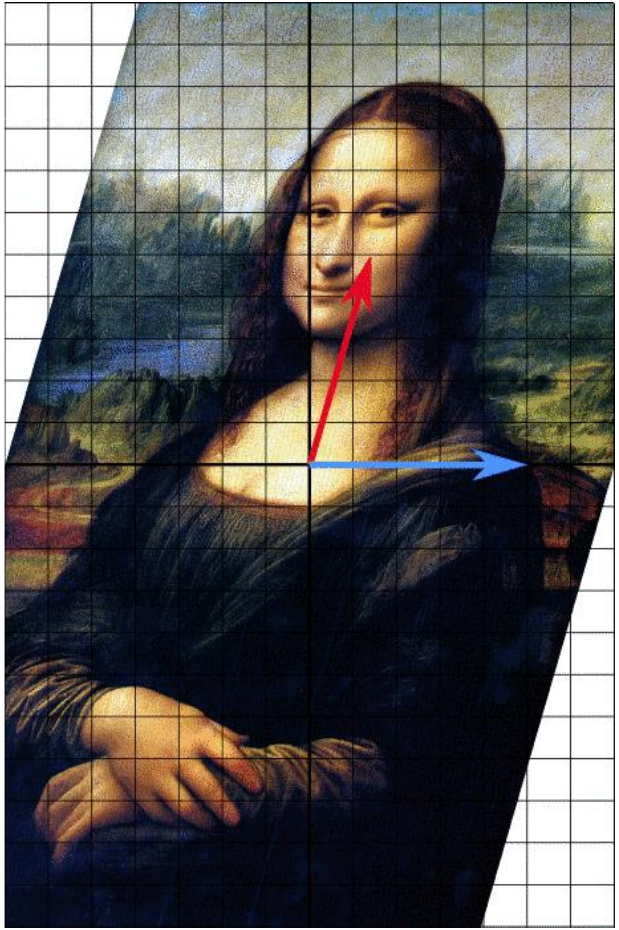
$$\boldsymbol{A}\mathbf{v} = \lambda\mathbf{v}$$

How to read equation:

An eigenvector is

- “a vector that changes at most by a scalar factor when a linear transformation is applied to it”
- “a vector whose direction remains unchanged when a linear transformation is applied to it.”





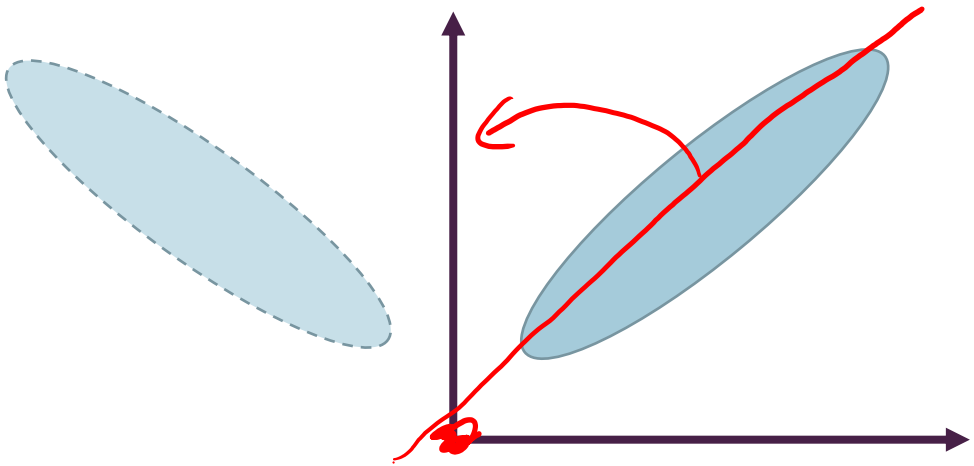
- red arrow changes direction
- blue arrow does not change direction

=> is an eigenvector of this shear mapping
=> length is unchanged
=> eigenvalue=1

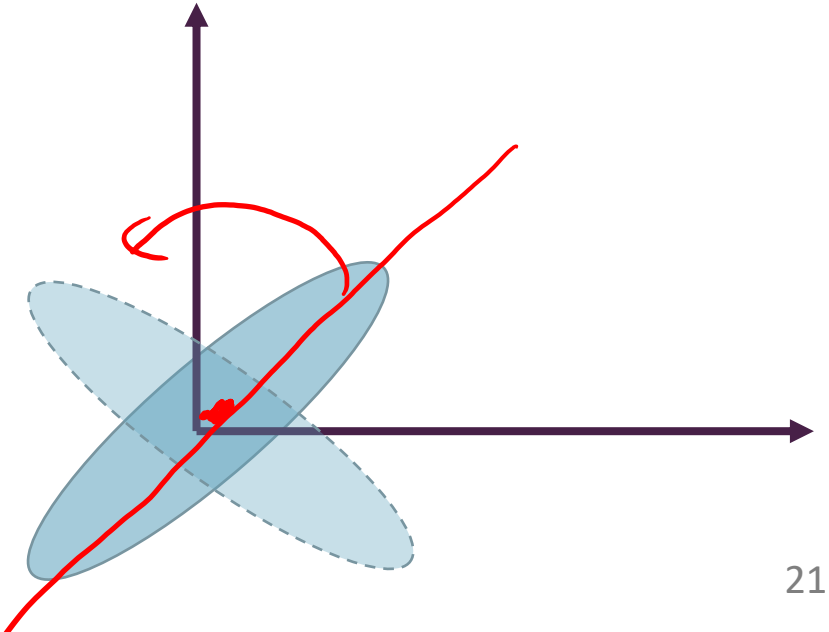
Rotation

Before rotating the data, we must center the data because… ?

original data



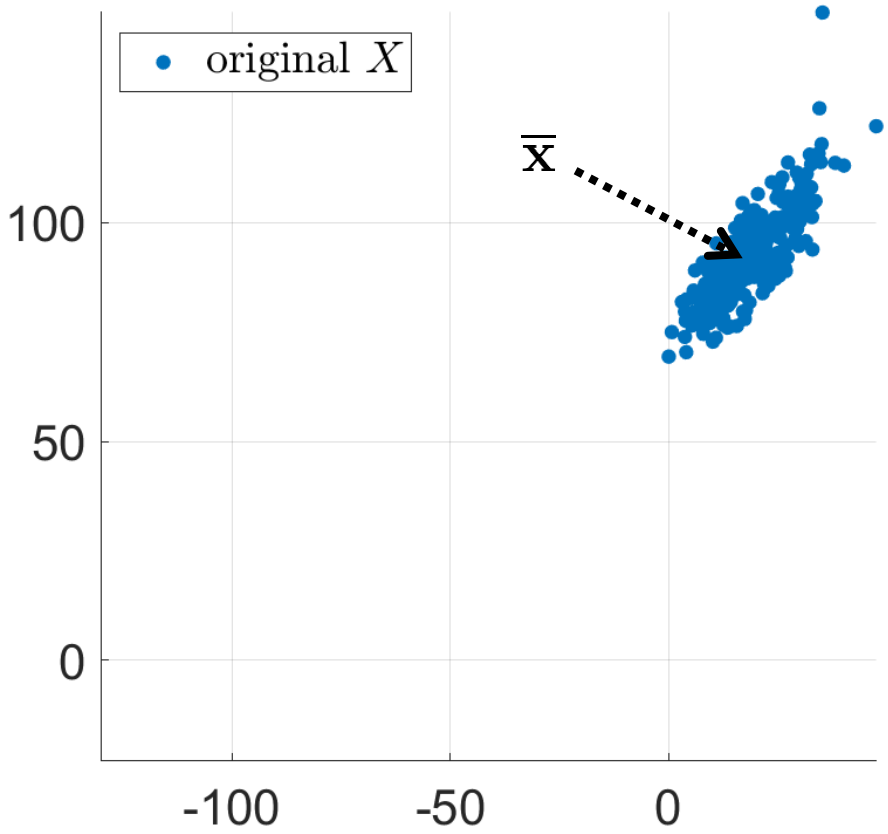
mean-free
centered data



PCA – How to 2D to 1D

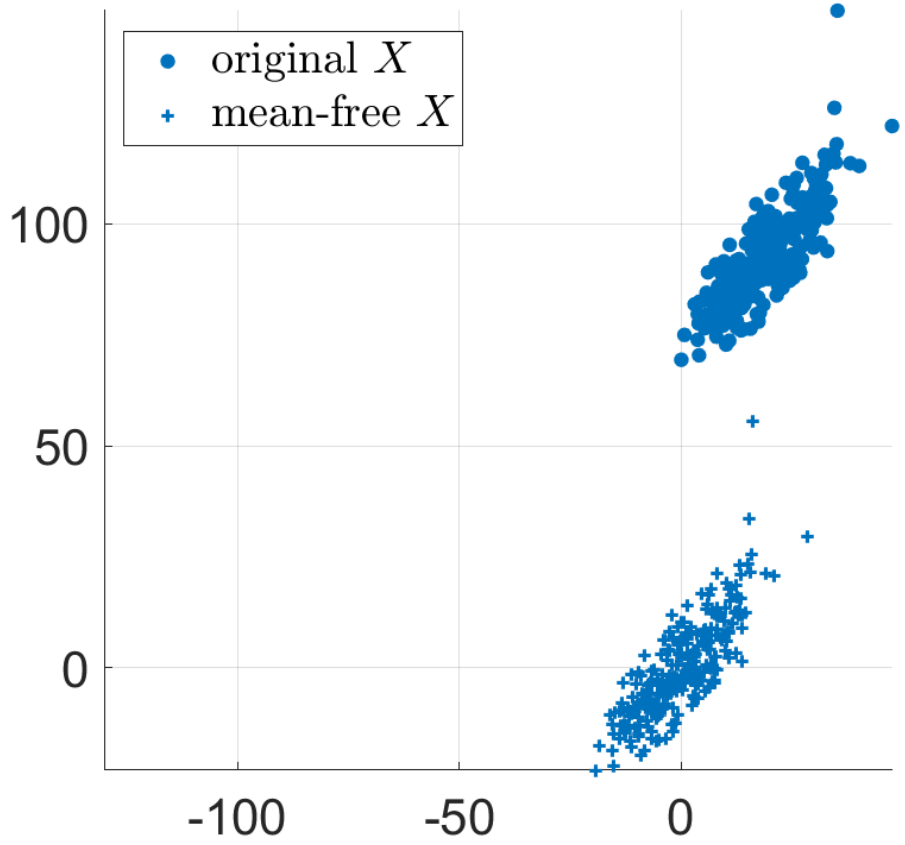
1. Subtract mean $\overline{\mathbf{x}}$

$$\mathbf{x}_n \in \mathbb{R}^2 \;\mapsto\; \mathbf{z}_n \in \mathbb{R}^1$$



PCA – How to 2D to 1D

$$\mathbf{x}_n \in \mathbb{R}^2 \;\mapsto\; \mathbf{z}_n \in \mathbb{R}^1$$



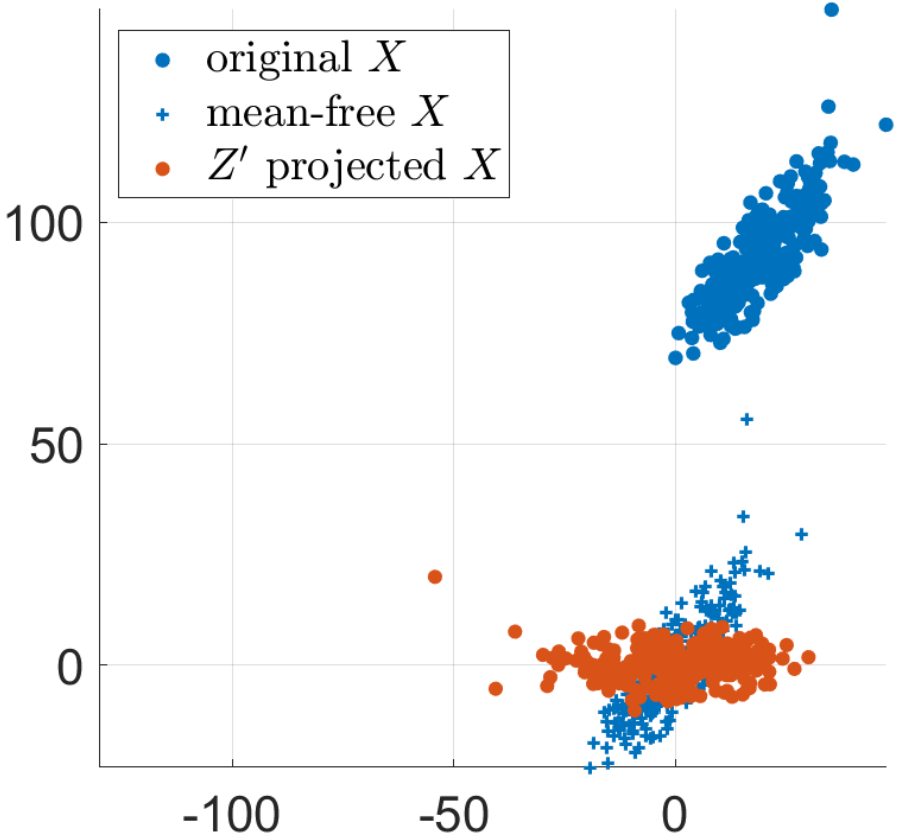
1. Compute mean $\overline{\mathbf{x}}$
2. Covariance matrix

data matrix

$$\boldsymbol{S} = \frac{1}{N-1}(\boldsymbol{X} - \boldsymbol{M})^{\mathrm{T}}(\boldsymbol{X} - \boldsymbol{M})$$

PCA – How to 2D to 1D

$$\mathbf{x}_n \in \mathbb{R}^2 \;\mapsto\; \mathbf{z}_n \in \mathbb{R}^1$$



1. Compute mean $\overline{\mathbf{x}}$
2. Covariance matrix

$$\boldsymbol{S} = \frac{1}{N-1}(\boldsymbol{X}-\boldsymbol{M})^{\mathrm{T}}(\boldsymbol{X}-\boldsymbol{M})$$

3. Get eigenvectors

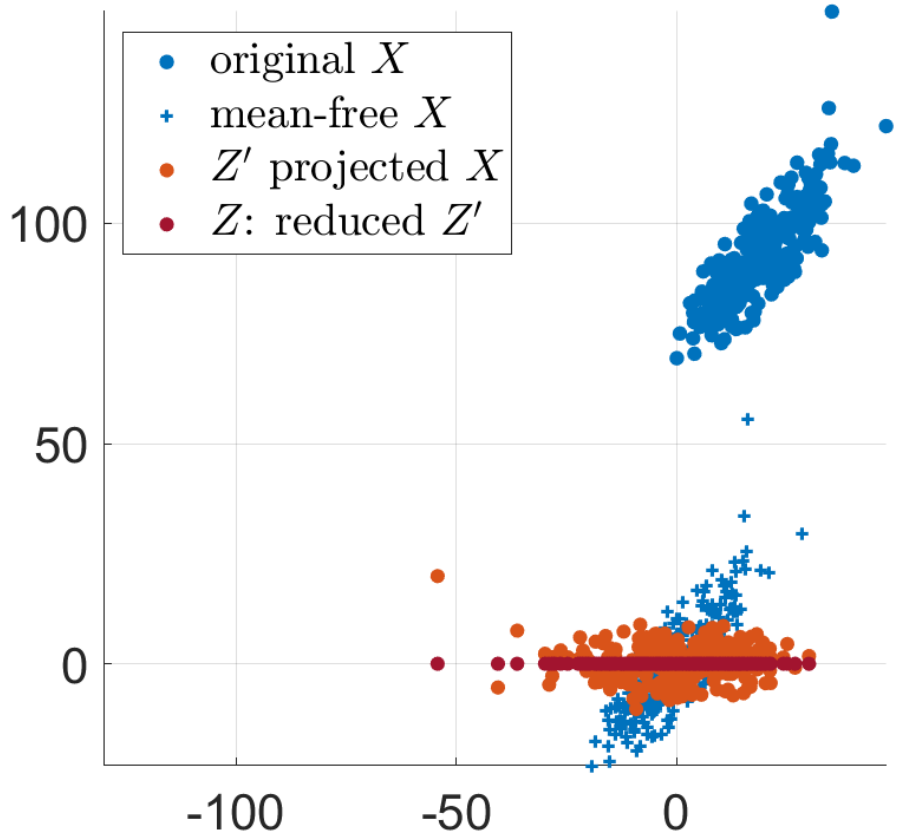
$$\boldsymbol{S}\mathbf{u}_k = \lambda_k \mathbf{u}_k, \;\; k = 1, 2$$

$$\boldsymbol{U} = [\mathbf{u}_1, \mathbf{u}_2]$$

$$\boldsymbol{Z} = (\boldsymbol{X}-\boldsymbol{M})\boldsymbol{U}$$

PCA – How to 2D to 1D

$$\mathbf{x}_n \in \mathbb{R}^2 \;\mapsto\; \mathbf{z}_n \in \mathbb{R}^1$$



1. Compute mean $\overline{\mathbf{x}}$
2. Covariance matrix

$$\boldsymbol{S} = \frac{1}{N-1}(\boldsymbol{X} - \boldsymbol{M})^{\mathrm{T}}(\boldsymbol{X} - \boldsymbol{M})$$

3. Get eigenvectors

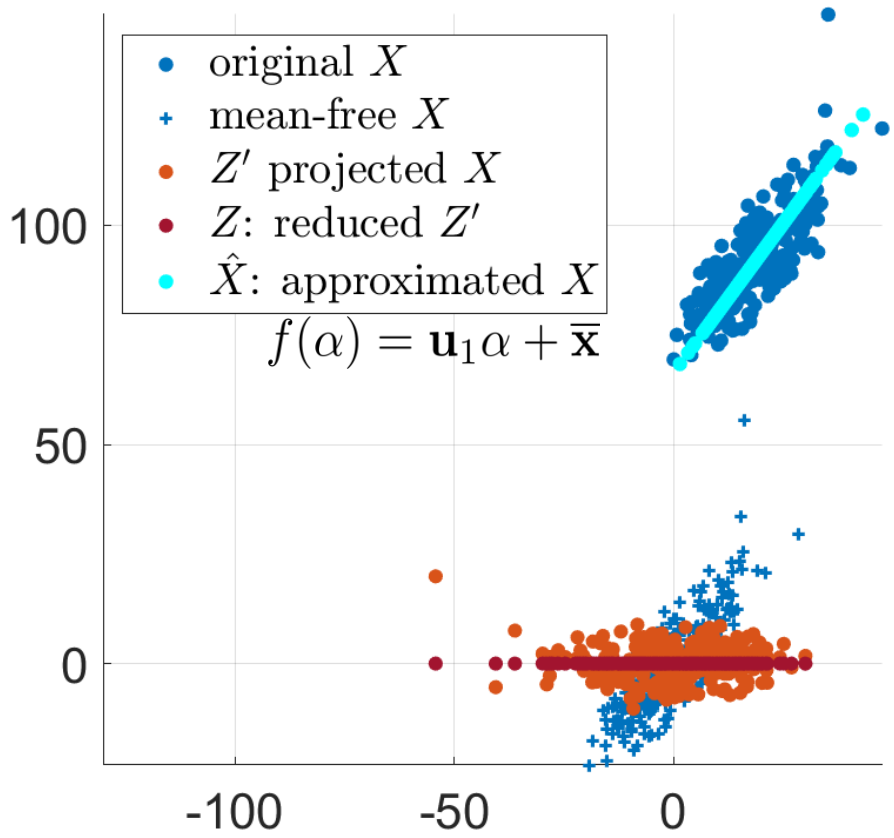
$$\boldsymbol{S}\mathbf{u}_k = \lambda_k \mathbf{u}_k, \;\; k = 1, 2$$
$$\boldsymbol{U} = [\mathbf{u}_1, \mathbf{u}_2]$$
$$\boldsymbol{Z} = (\boldsymbol{X} - \boldsymbol{M})\boldsymbol{U}$$

4. Choose K=1
   - Reduction

$$z_n = \boldsymbol{u}_1^{\mathrm{T}}(\mathbf{x}_n - \overline{\mathbf{x}}) \in \mathbb{R}^1$$

PCA – How to 2D to 1D

$$\mathbf{x}_n \in \mathbb{R}^2 \;\mapsto\; \mathbf{z}_n \in \mathbb{R}^1$$



1. Compute mean $\overline{\mathbf{x}}$
2. Covariance matrix

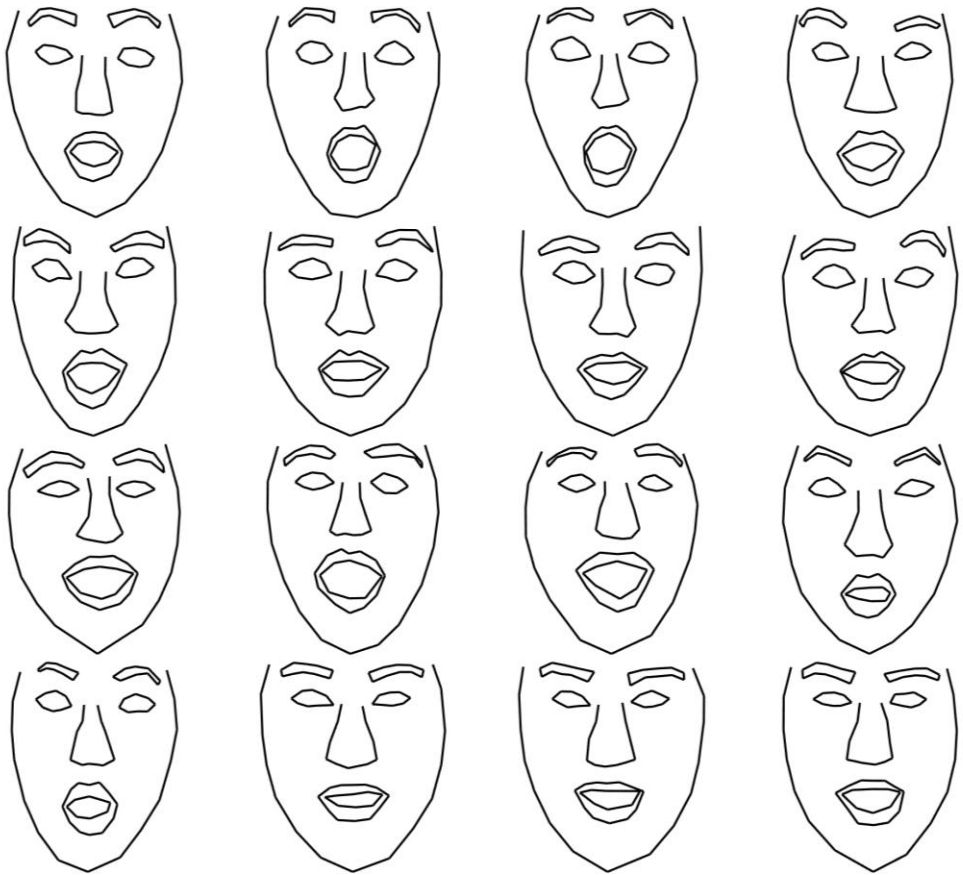
$$\boldsymbol{S} = \frac{1}{N-1}(\boldsymbol{X} - \boldsymbol{M})^{\mathrm{T}}(\boldsymbol{X} - \boldsymbol{M})$$

3. Get eigenvectors

$$\boldsymbol{S}\mathbf{u}_k = \lambda_k \mathbf{u}_k, \;\; k = 1, 2$$
$$\boldsymbol{U} = [\mathbf{u}_1, \mathbf{u}_2]$$
$$\boldsymbol{Z} = (\boldsymbol{X} - \boldsymbol{M})\boldsymbol{U}$$

4. Choose K=1
   - Reduction
     $z_n = \boldsymbol{u}_1^{\mathrm{T}}(\mathbf{x}_n - \overline{\mathbf{x}}) \in \mathbb{R}^1$
   - Reconstruction
     $\widehat{\mathbf{x}}_n = \boldsymbol{u}_1 z_n + \overline{\mathbf{x}} \in \mathbb{R}^2$

Example: PCA Model for Surprised Faces



Data:

- 400 3D faces
- 83 features points

$$N = 400$$

$$D = 3 \cdot 83 = 249$$

$$\mathbf{X} \in \mathbb{R}^{N \times D}$$

PCA gives:

- Eigenvectors

$$\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_D]$$

- Eigenvalues

$$\lambda_1, \ldots, \lambda_D$$

The effect of preprocessing

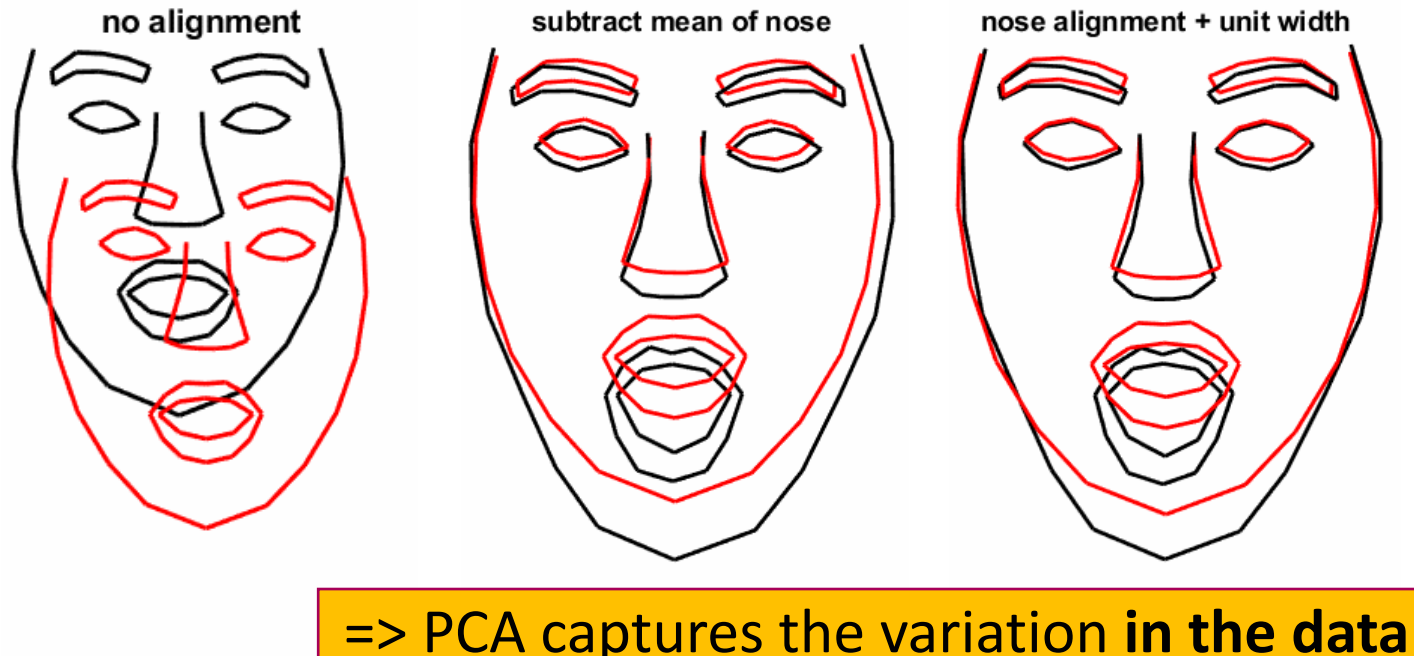
$$\hat{\mathbf{x}} = \hat{\boldsymbol{\mu}} + \alpha_1 \mathbf{u}_1$$

← 1. principal component = 1. eigenvector

vary this scalar $-3\sqrt{\lambda_k} \leq \alpha_k \leq 3\sqrt{\lambda_k}$

new face

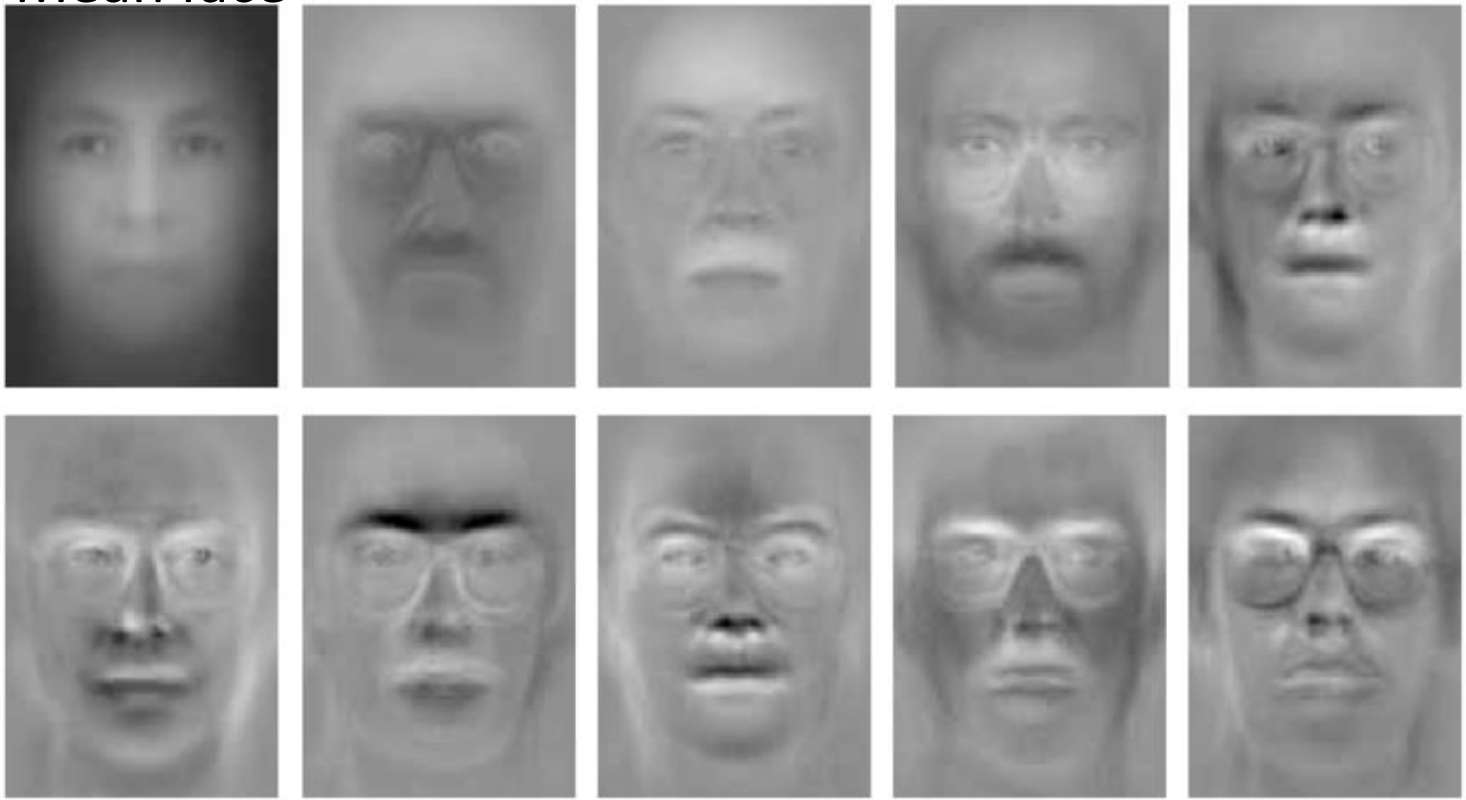
mean face (static)



=> PCA captures the variation **in the data**

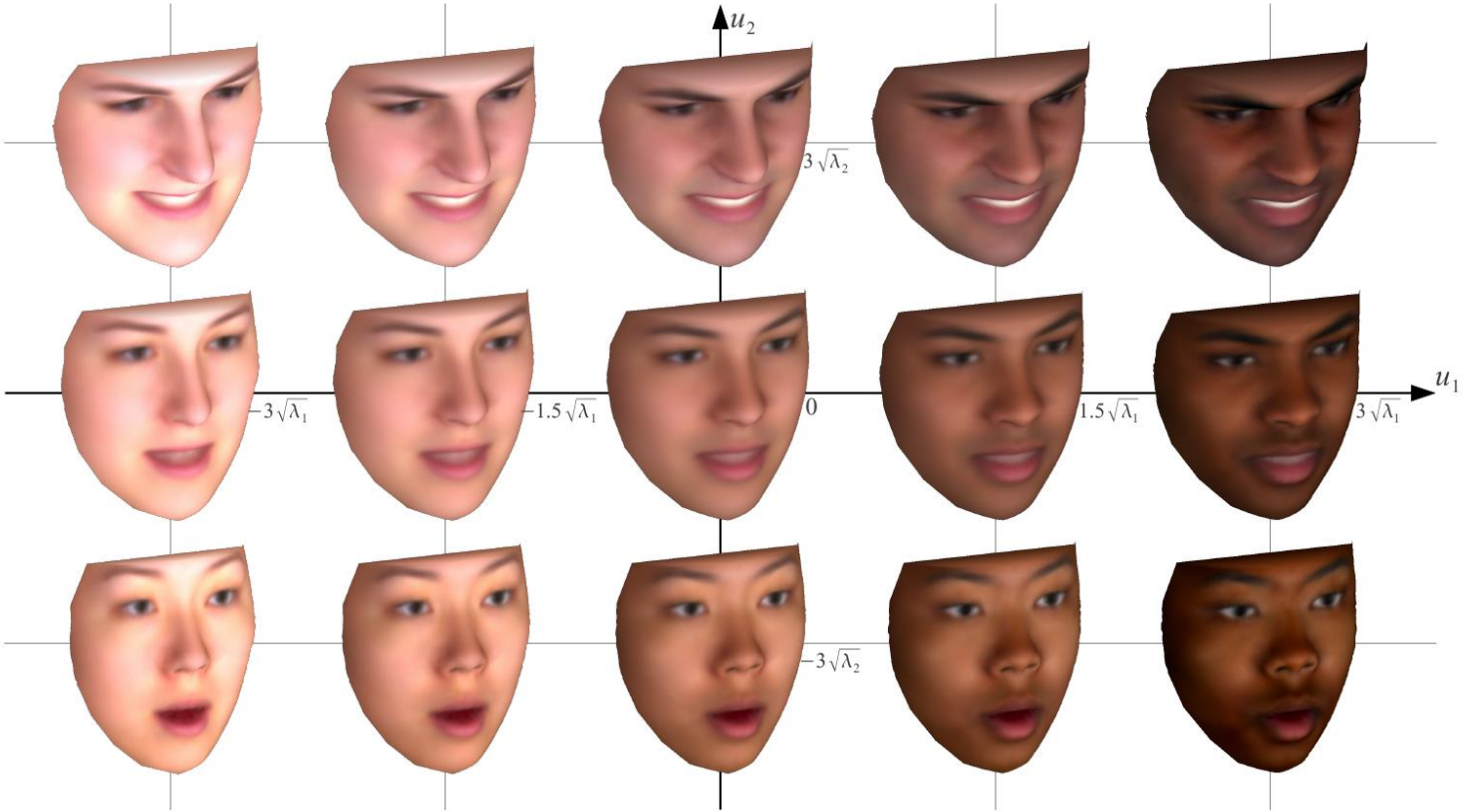
Eigenfaces (1991)

Mean face



Source: Jean-Luc Nagel, https://www.researchgate.net/figure/1-Example-of-eigenfaces-Example-obtained-from-the-X2MVTS-database-cf-Subsection_fig3_33682412

PCA – Face Model



"Dense point-to-point correspondences between 3D faces using parametric remeshing for constructing 3D Morphable Models", Kaiser, et al., 2011

PCA for high-dimensional data

Problem: if N<D
Covariance matrix **S** is large, eigenvalue decomposition costly

Goal:
Transfer eigenvalue problem to lower dimension

$$\boldsymbol{X} := \boldsymbol{X}_0 - \boldsymbol{M} \in \mathbb{R}^{N \times D}$$

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} \in \mathbb{R}^{D \times D} \rightsquigarrow \boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} \in \mathbb{R}^{N \times N}$$

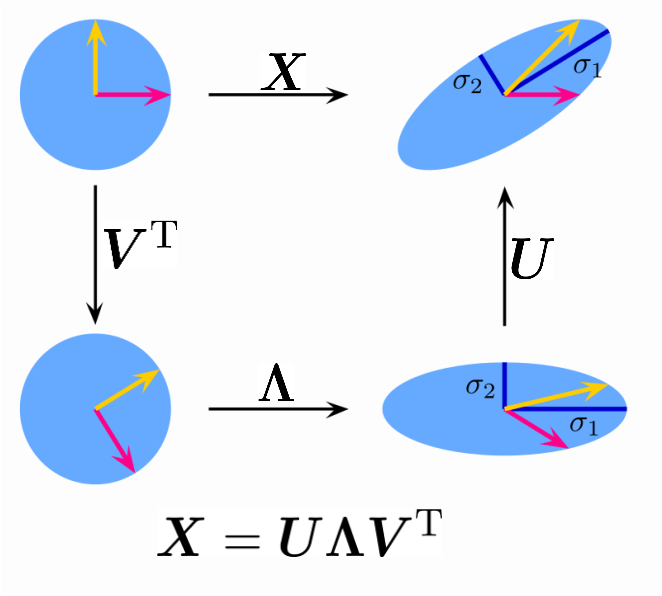
$$\frac{1}{N-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\mathbf{u}_i = \lambda_i \mathbf{u}_i$$

$$\Leftrightarrow \quad \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\underbrace{\boldsymbol{X}\mathbf{u}_i}_{\mathbf{v}_i} = \lambda_i \underbrace{\boldsymbol{X}\mathbf{u}_i}_{\mathbf{v}_i}$$

$$\Leftrightarrow \quad \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\mathbf{v}_i = \lambda_i \mathbf{v}_i \qquad \Rightarrow \mathbf{u}_i = \frac{1}{\sqrt{N\lambda_i}}\boldsymbol{X}^{\mathrm{T}}\mathbf{v}_i$$

$$\Leftrightarrow \left(\frac{1}{N}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)\left(\boldsymbol{X}^{\mathrm{T}}\mathbf{v}_i\right) = \lambda_i \left(\boldsymbol{X}^{\mathrm{T}}\mathbf{v}_i\right)$$

PCA vs. SVD



Singular Value Decomposition (SVD)

X is *N*x*D*

SVD is a factorization

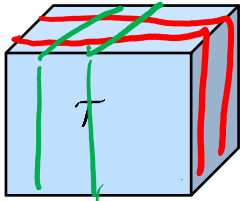
$$X = U\Lambda V^{\mathrm{T}}$$

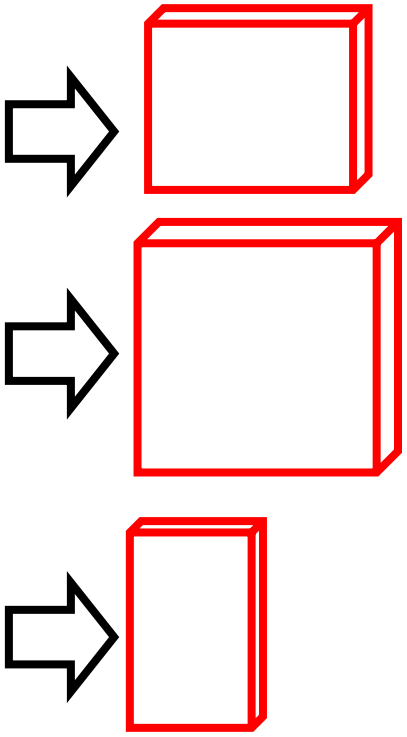
- U : *N*x*N* eigenvectors of XX^{T}
- Λ : *N*x*D* singular values on first K diagonal
- V : *D*x*D* eigenvectors of $X^{\mathrm{T}}X$

https://en.wikipedia.org/wiki/Singular_value_decomposition

Higher-Order SVD – SVD on data tensors

Data: 2500 3D face scans





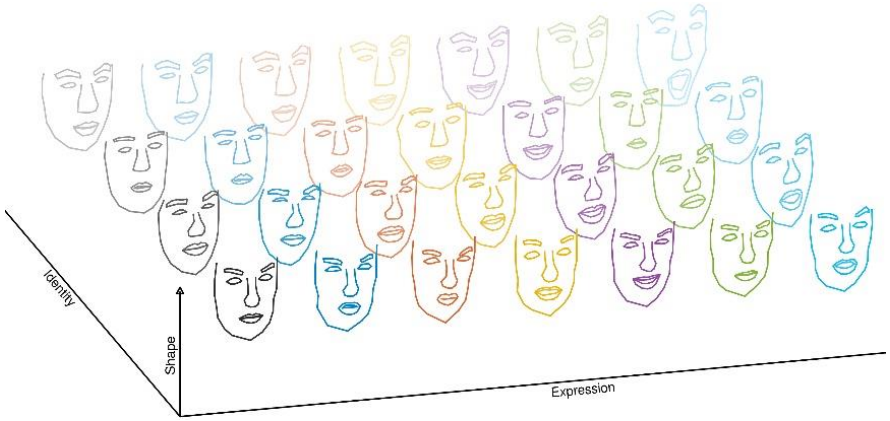
$$\boldsymbol{X}^{(1)} = \boldsymbol{U}^{(1)}\boldsymbol{\Lambda}^{(1)}\boldsymbol{V}^{(1)^{\mathrm{T}}}$$

$$\boldsymbol{X}^{(2)} = \boldsymbol{U}^{(2)}\boldsymbol{\Lambda}^{(2)}\boldsymbol{V}^{(2)^{\mathrm{T}}}$$

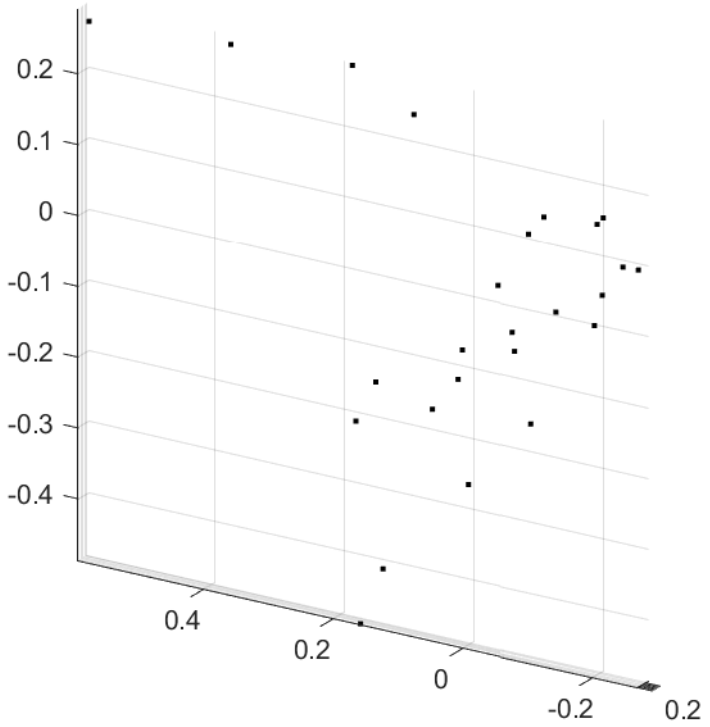
$$\boldsymbol{X}^{(3)} = \boldsymbol{U}^{(3)}\boldsymbol{\Lambda}^{(3)}\boldsymbol{V}^{(3)^{\mathrm{T}}}$$

Higher-Order SVD

Data: 2500 3D face scans



$$\boldsymbol{X}^{(3)} = \boldsymbol{U}^{(3)} \boldsymbol{\Lambda}^{(3)} \boldsymbol{V}^{(3)\mathrm{T}}$$

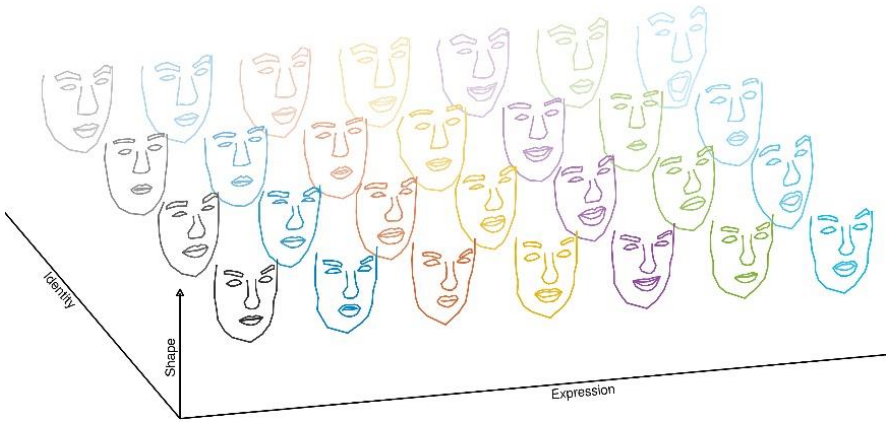


$$\widetilde{\boldsymbol{U}}^{(3)} \in \mathbb{R}^{E \times 3}$$

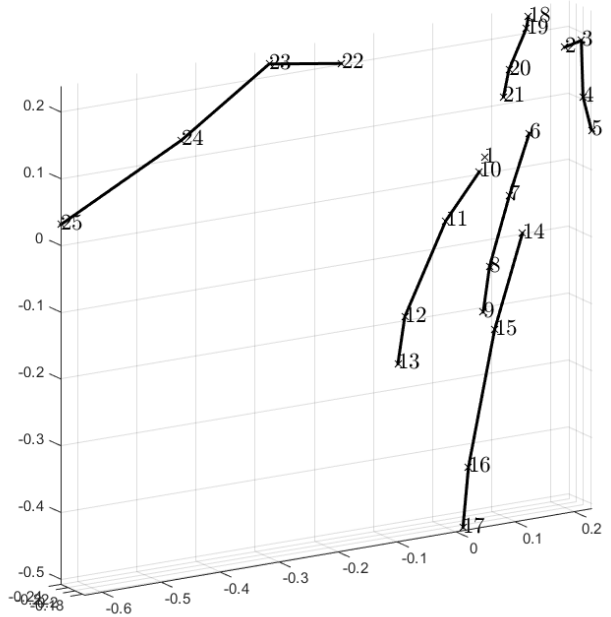
$$E = 25$$

Higher-Order SVD

Data: 2500 3D face scans



$$\boldsymbol{X}^{(3)} = \boldsymbol{U}^{(3)} \boldsymbol{\Lambda}^{(3)} \boldsymbol{V}^{(3)^{\mathrm{T}}}$$

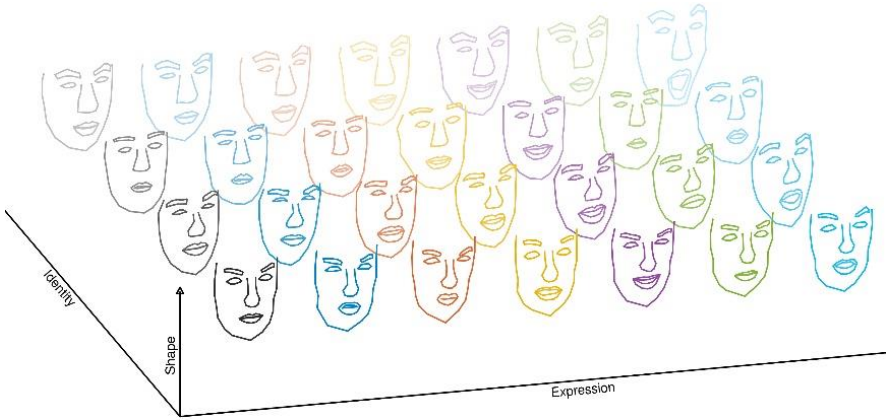


$$\widetilde{\boldsymbol{U}}^{(3)} \in \mathbb{R}^{E \times 3}$$

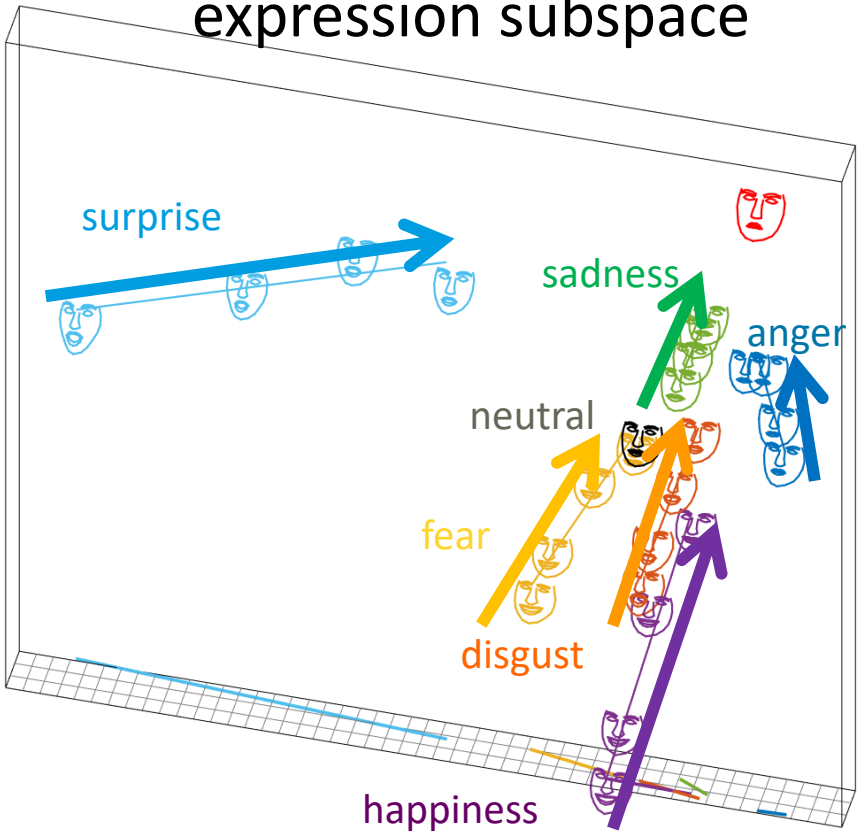
$$E = 25$$

Higher-Order SVD

Data: 2500 3D face scans



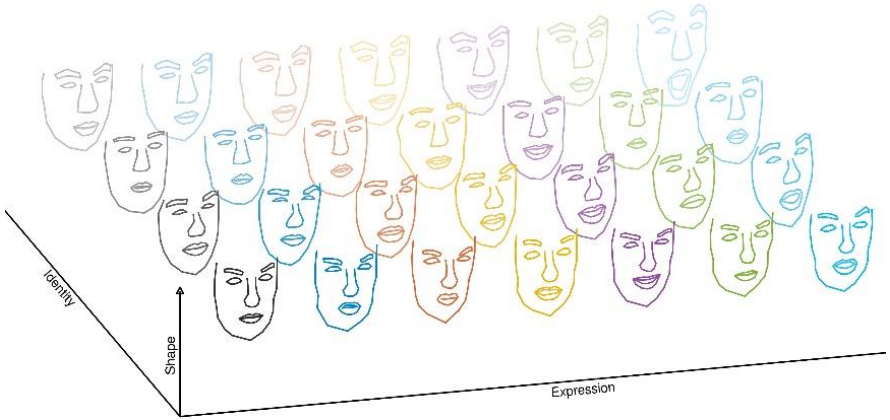
expression subspace



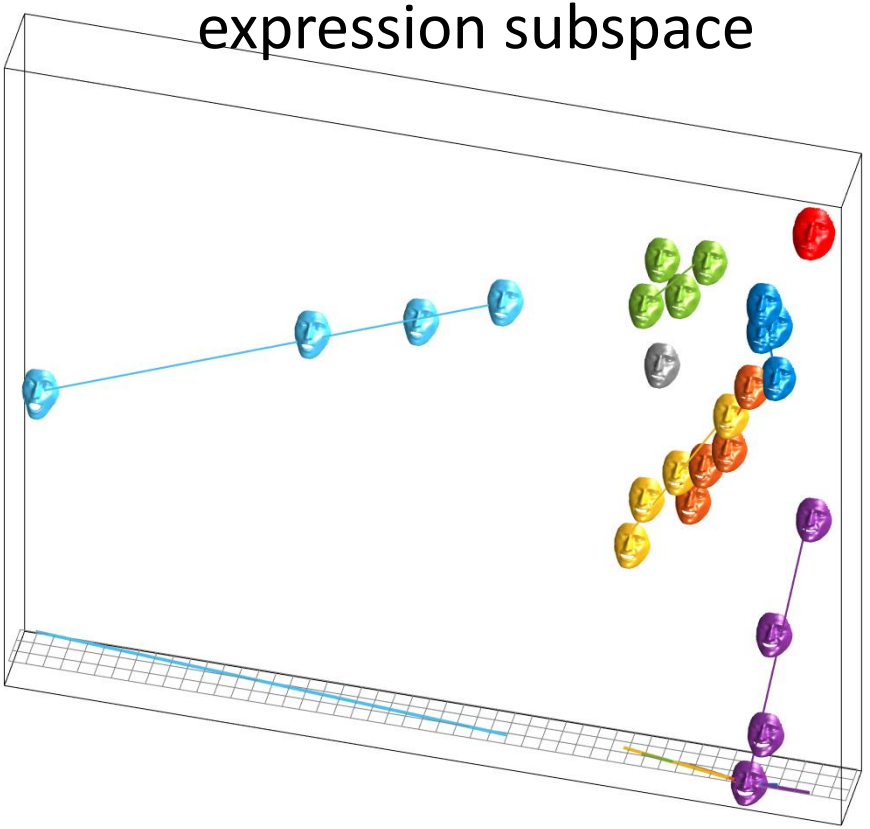
Apathy : emotion with zero strength

Higher-Order SVD

Data: 2500 3D face scans



expression subspace



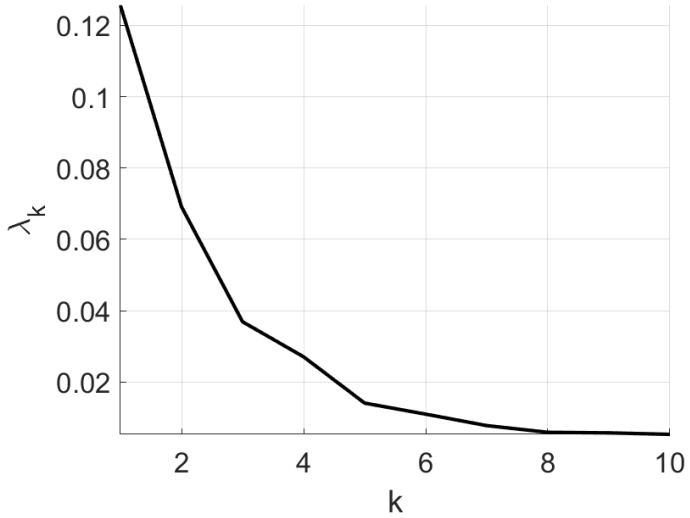
Apathy : emotion with zero strength

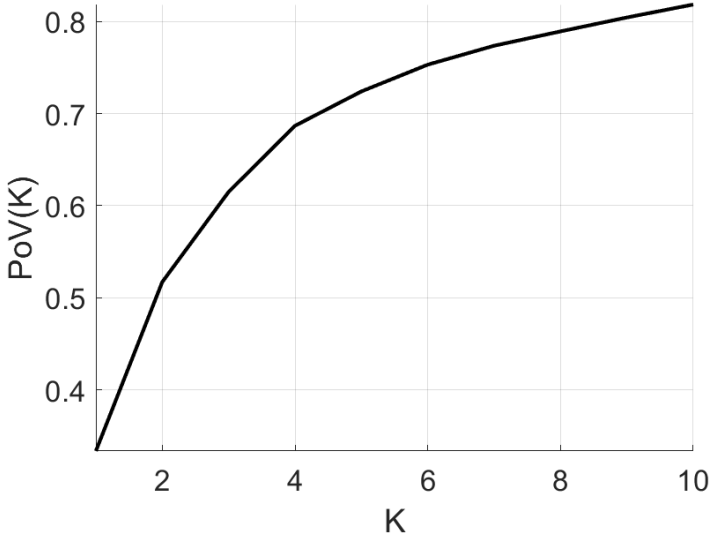
Proportion of explained Variance (PoV)

$$\mathrm{PoV}(K) = \frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{D} \lambda_i}, \quad K \leq D$$

eigenvalues

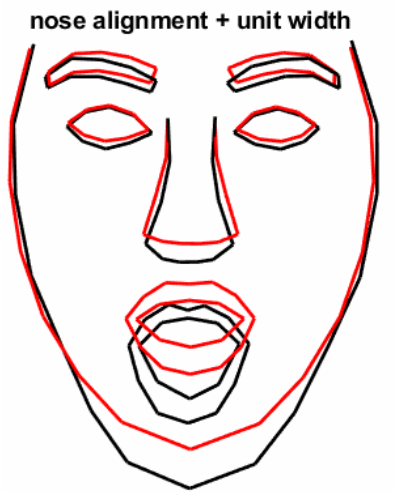
$\lambda_1 \geq \ldots \geq \lambda_D$

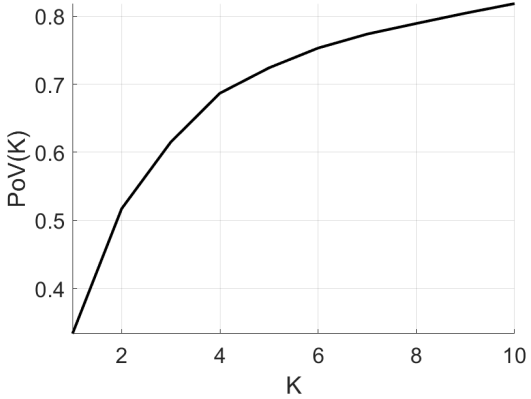




D=249=3*83 feature points

PCA - Properties



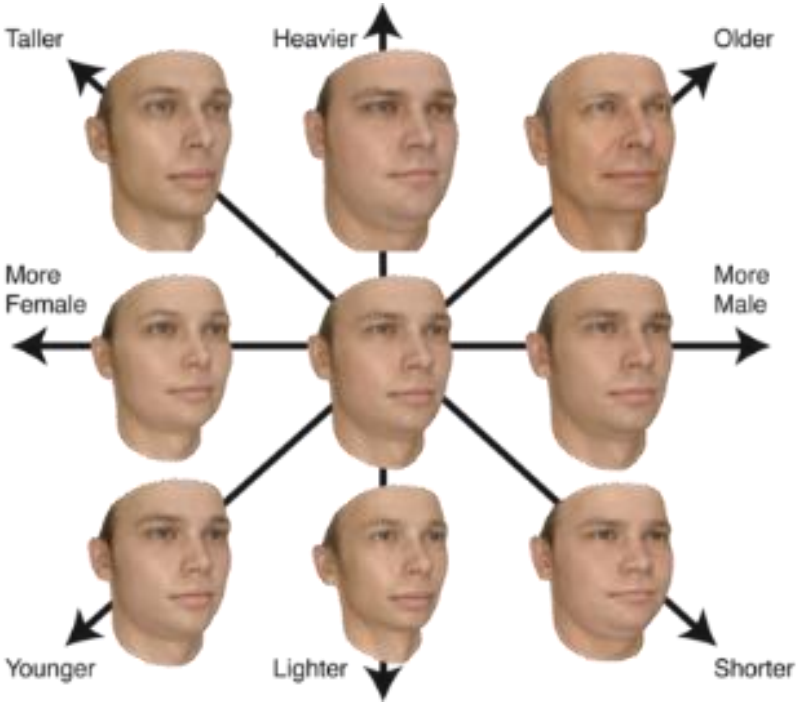


- Unsupervised
- Deterministic
- Analytical solution
- Linear combination of samples

Choices to make:

- Sensible preprocessing is crucial
- Number of principal components
- Not as suitable for big data

PCA - Applications

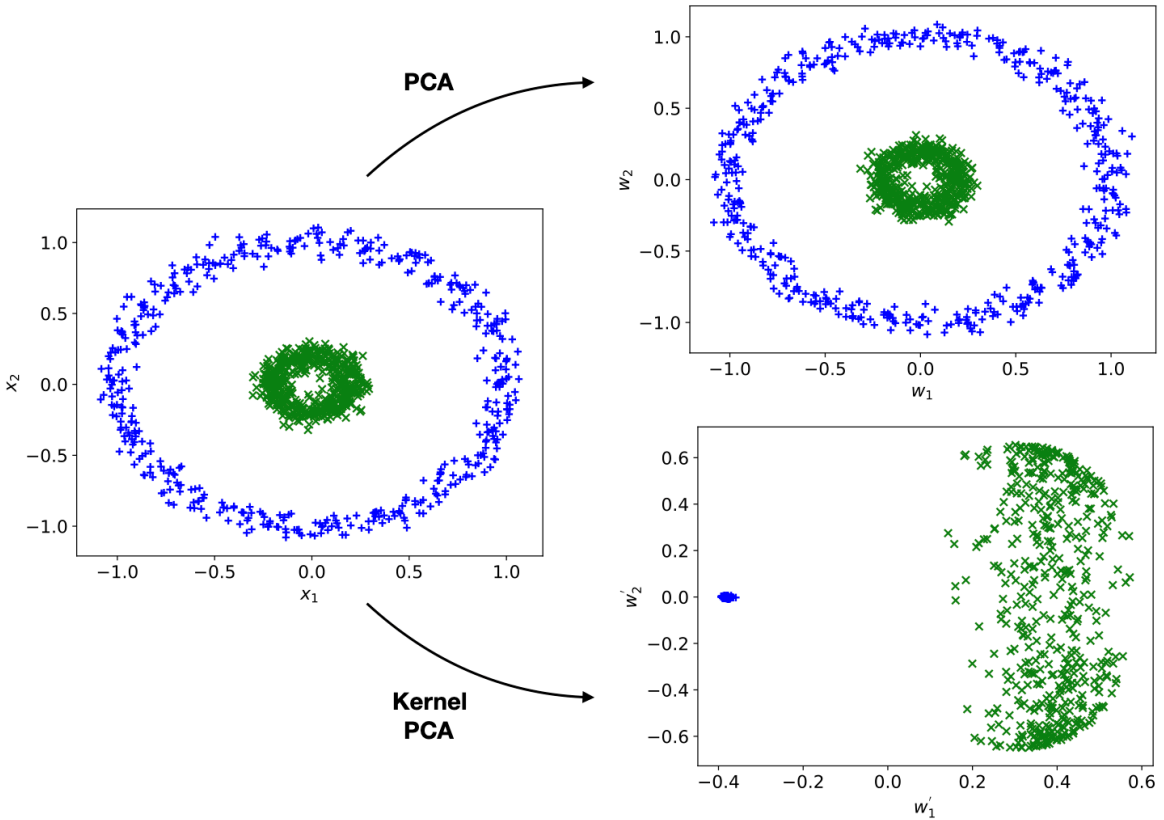


Blanz and Vetter, 1999

- Data compression
- Dimensionality reduction
- Data preprocessing
- Creating a model
  - Approximation
  - Interpolation
  - New samples

Attention:
no semantic directions!

Kernel PCA



$\mathbf{x}_n$

$\mathbf{x}_n = \boldsymbol{\phi}(\mathbf{x}_n)$

https://ml-lectures.org/docs/structuring_data/ml_without_neural_network-2.html

Extensions of PCA and other Latent Variable Models

- Incremental PCA: on batches
- kernel PCA: nonlinear relations $\mathbf{x}_n = \phi(\mathbf{x}_n)$

- Probabilistic PCA:
  prior on latent variables

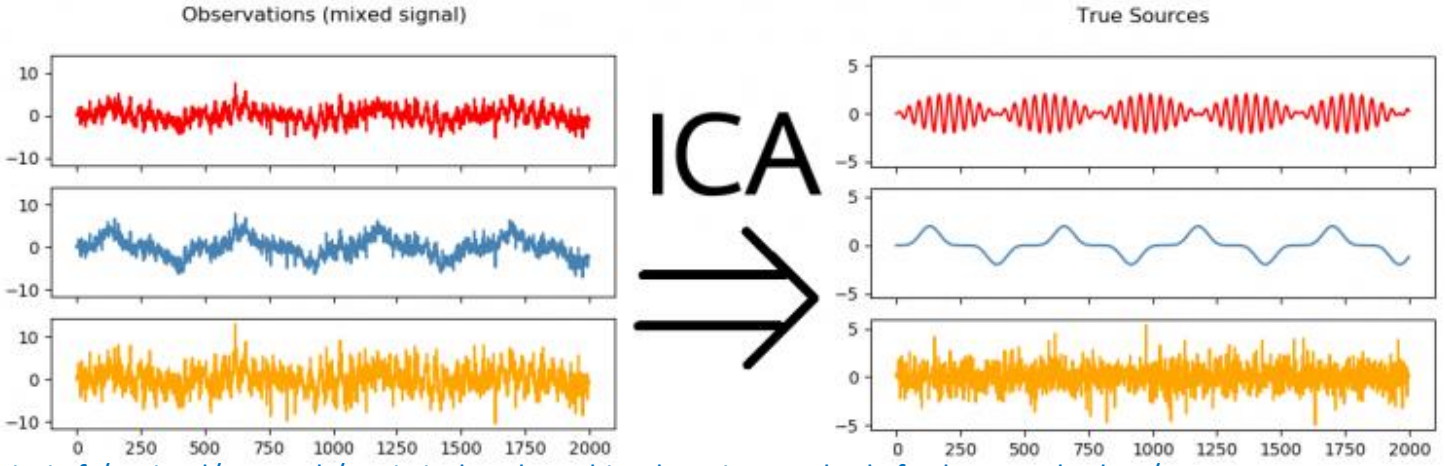
$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$$

$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I})$$

- Bayesian PCA:
  prior on model parameters

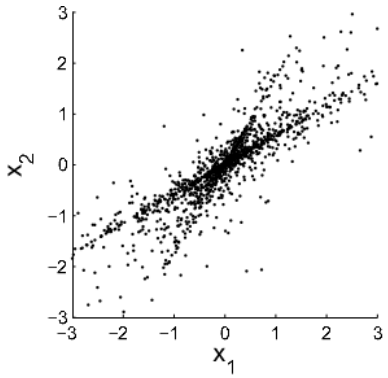
Independent Component Analysis (ICA)

- PCA gives uncorrelated components (features)
- ICA gives independent components
  - **Given:** mixture of signals $\mathbf{x} = \boldsymbol{A}\mathbf{s}$
    Goal: find the source signals (independent)
  - e.g. Cocktail party problem, source separation
  - problem: no unique solution



https://team.inria.fr/parietal/research/statistical-and-machine-learning-methods-for-large-scale-data/faster-independent-component-analysis-for-real-data/

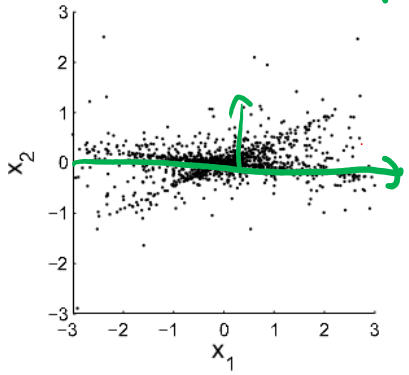
Independent Component Analysis (ICA)



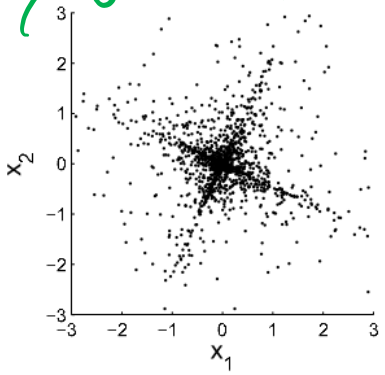
uncorrelated

unit variance

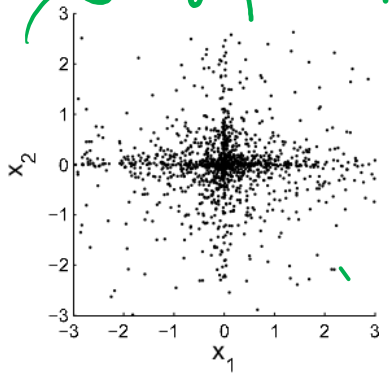
independent



PCA



Whitening



ICA

Motivation: Why decorrelate data?

Motivation: decorrelate data

Imagine task: color segmentation, e.g. yellow stars

Motivation: Why decorrelate data?

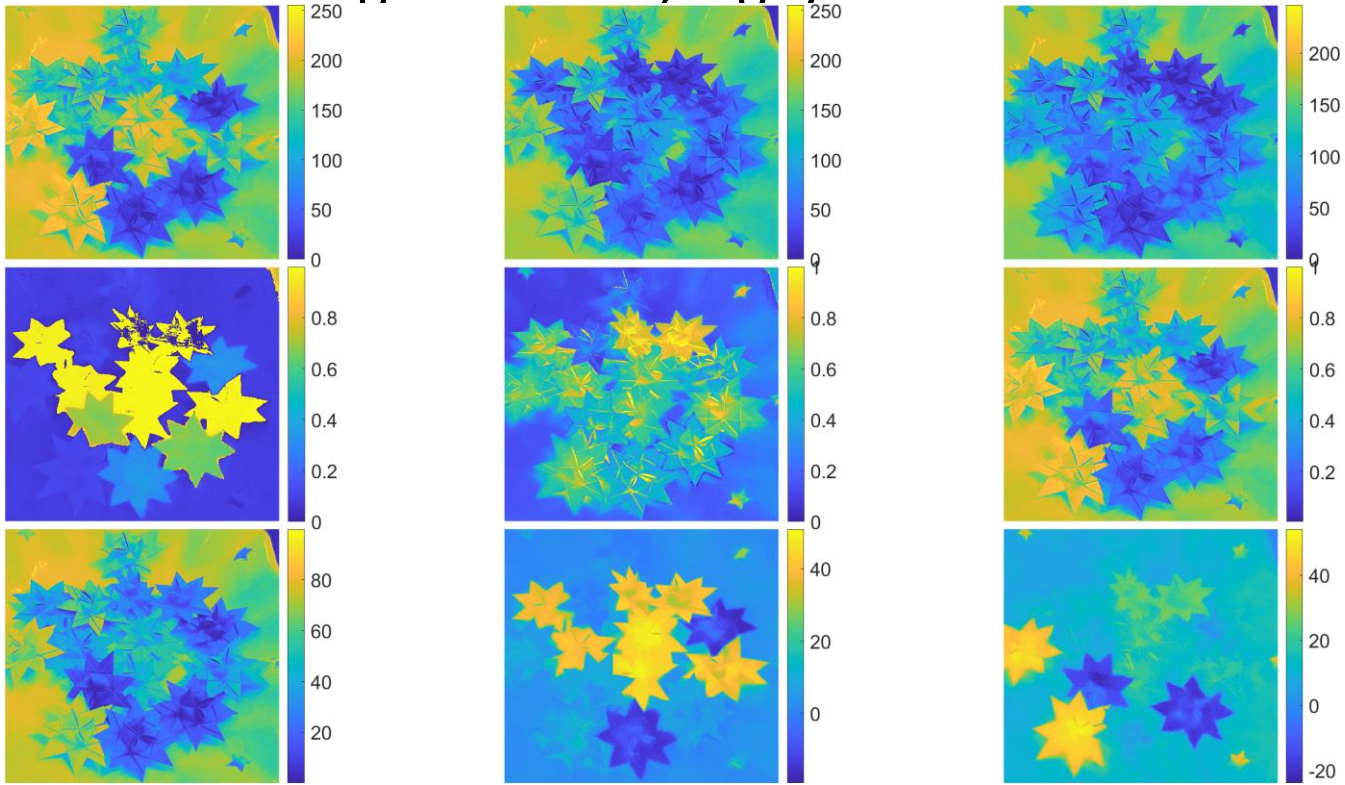
Motivation: decorrelate data

Imagine task: color segmentation, e.g. yellow stars

RGB

HSV

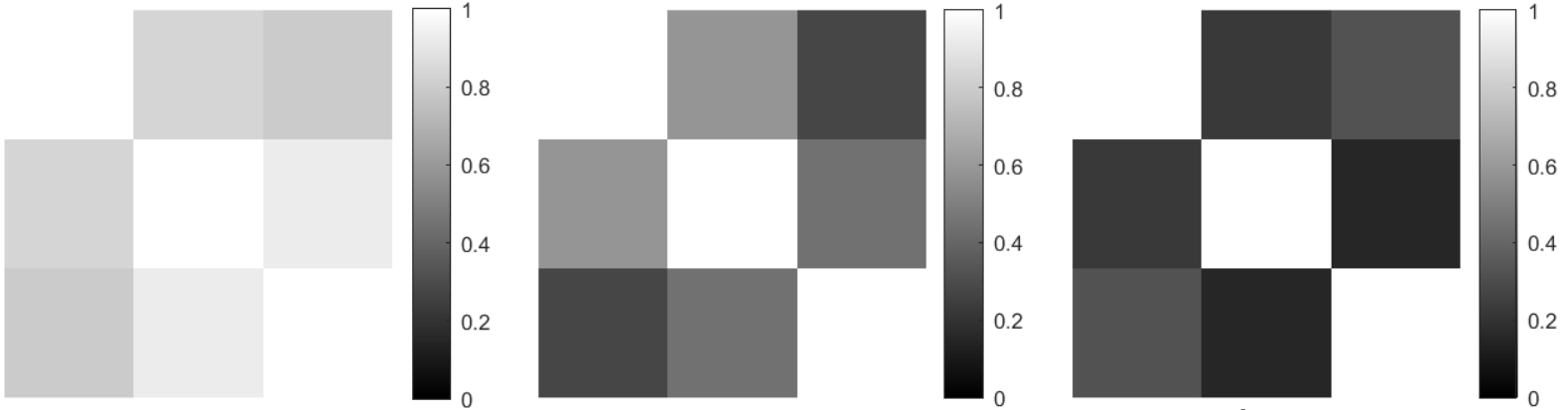
Lab



Motivation: Why decorrelate data?

Motivation: decorrelate data

Imagine task: color segmentation, e.g. yellow stars



RGB HSV Lab

Channels of …

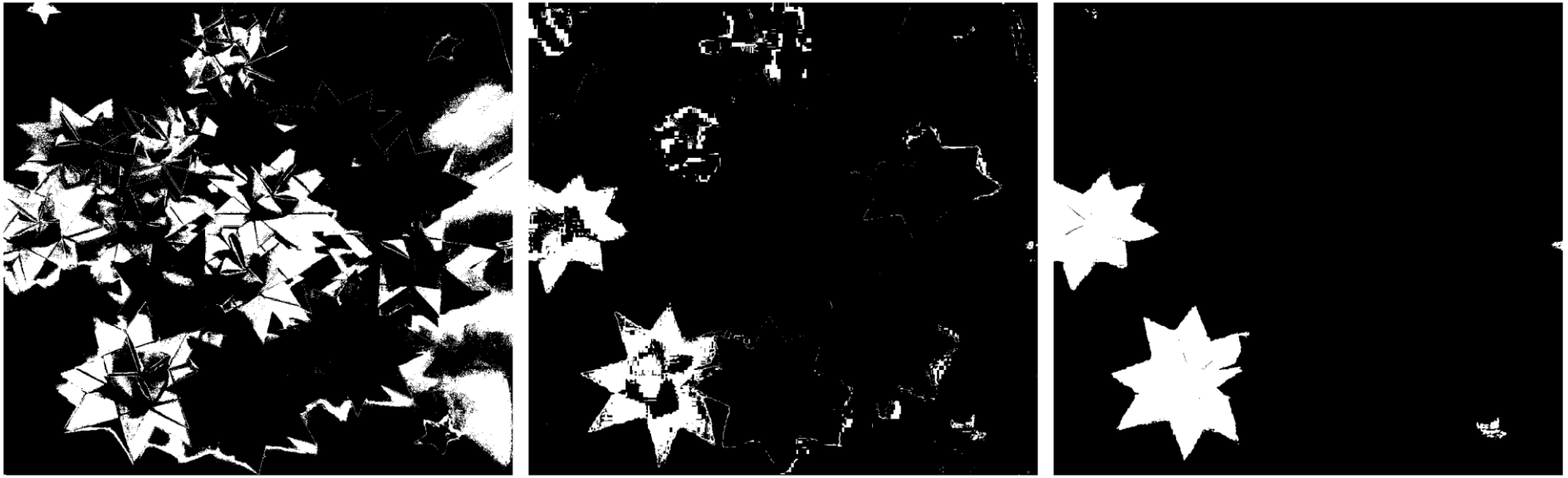
- RGB are highly correlated
- HSV are somewhat correlated
- Lab are “not” correlated

Motivation: Why decorrelate data?

Motivation: decorrelate data

Imagine task: color segmentation, e.g. yellow stars

RG[B] : $|B-\lambda_B|<\tau_B$ [H]SV : $|H-\lambda_H|<\tau_H$ La[b] : $|b-\lambda_b|<\tau_b$

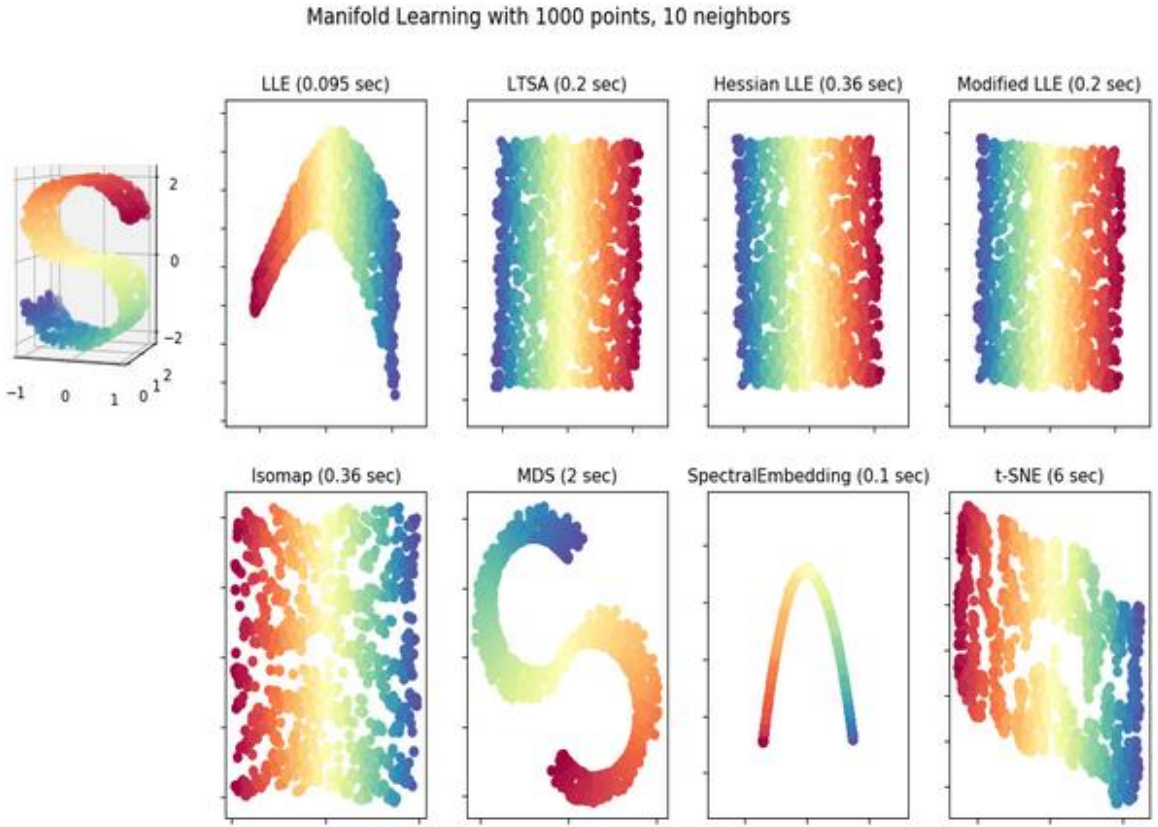


Results of manual thresholding

Reflection

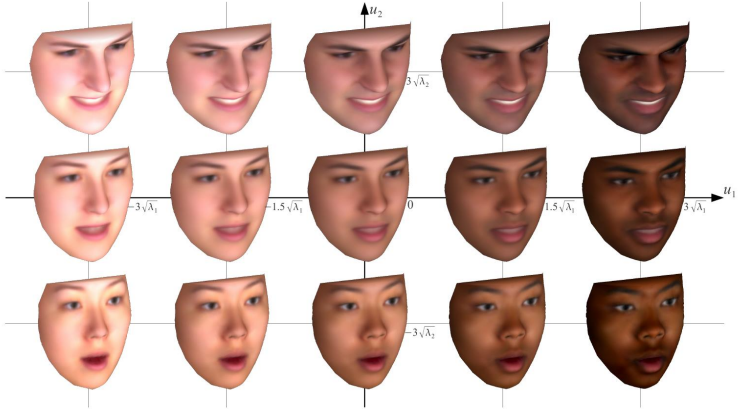
- Feature extraction vs. Feature Selection
- What is an embedding?

There is much more



https://scikit-learn.org/stable/modules/manifold.html

Summary



Principal Component Analysis

- Unsupervised method
- Linear factorization approach
- Various applications
- Different extensions