Lecture 8 – Principal component analysis

Bulat Ibragimov

bulat@di.ku.dk

Department of Computer Science
University of Copenhagen

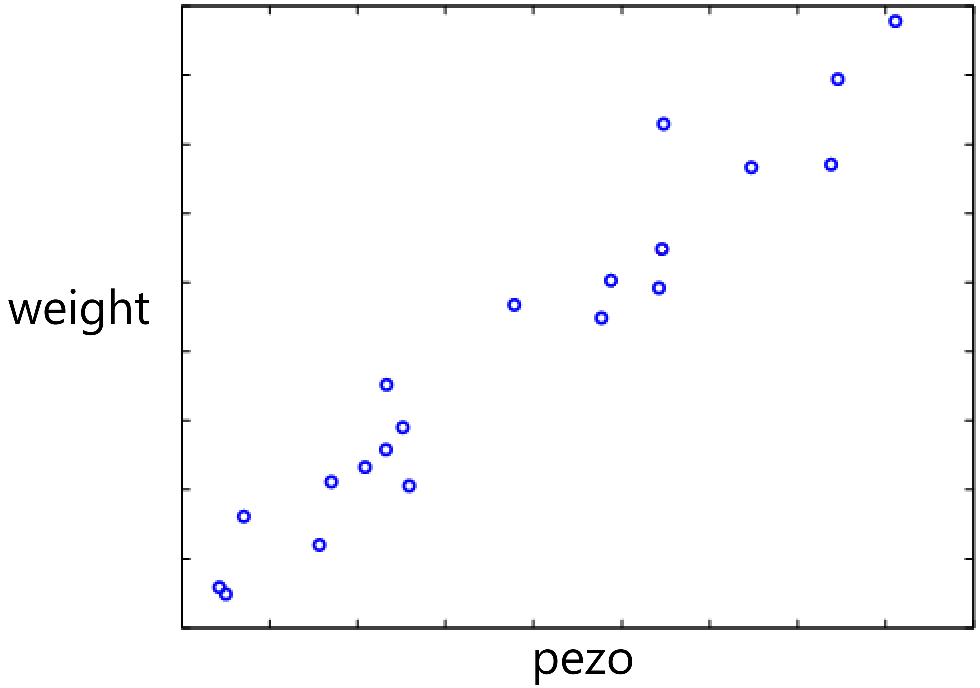
UNIVERSITY OF COPENHAGEN

Lecture 8 objectives

- Dimensionality reduction problem
- Dimensionality reduction as maximal variance
- Principal component analysis
- Selection of dimensions in PCA
- Applications

Dimensionality

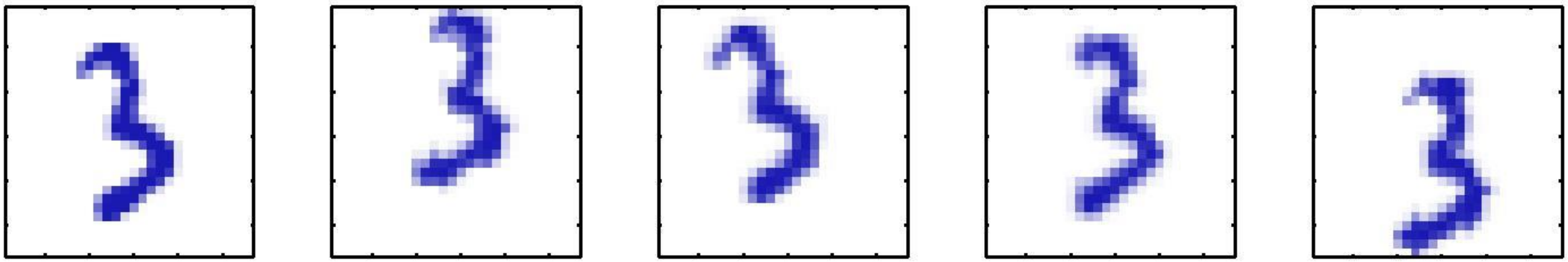
- Let's say you got a set of observations with two features {weight, pezo}.
- What is the dimensionality of this set, considering its plot?



- The true dimensionality is one, and there was just some imperfection in measurements (pezo = weight in Esperanto)

Curse of dimensionality

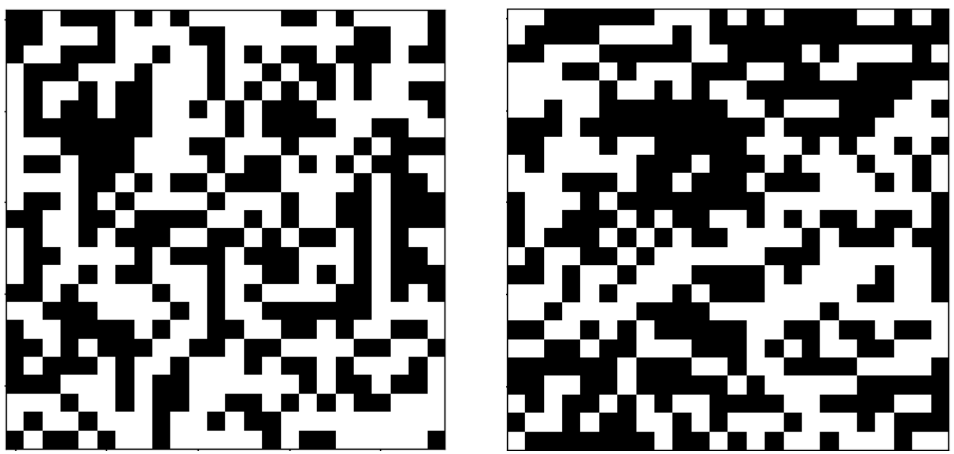
- Often high dimensional data have few degrees of freedom, i.e. a low intrinsic dimensionality.
- Example: Images of hand written digits



- Intrinsic degrees of freedom (< 24 x 24):
  - **Easy:** Translation (2), rotation (1)
  - **Complicated:** Degrees of freedom coming from the variability in how to write the digit 3.
  - Not all images represents valid digits – the set of digit images is sparsely distributed in the space of images.

Curse of dimensionality

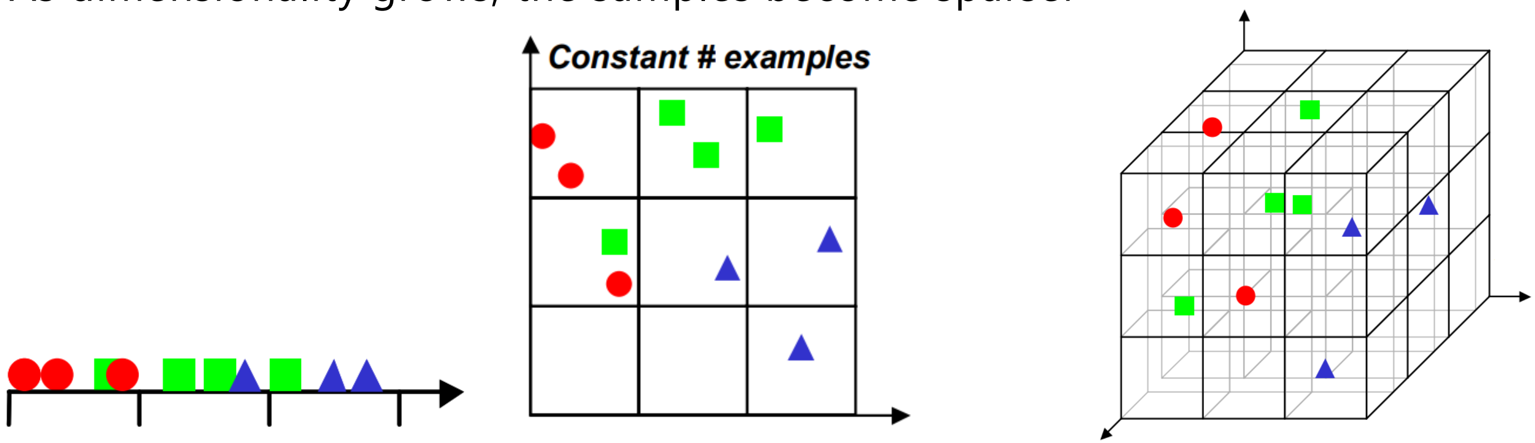
- How do we know that the true dimensionality of handwritten digits is way lower that (24x24)?
- The total number of binary images of 24x24 is 2^(24x24). If we start generating them, we will almost never get digit-like images



- The true dimensionality is restricted by the variations of continues lines with a relatively low length

Curse of dimensionality

- Why do we need to estimate the true dimensionality?
- Machine learning relies very much on statistics:
  - As dimensionality grows, the samples become sparser



Sample in 1D Sample in 2D Sample in 3D

- We will not be able to generalize if the number of samples is not appropriate for problem dimensionality:
  - One person with a rare name X is carpenter, dose this mean

$$P(\text{job} = \text{carpenter} \mid \text{name} = X) = 1$$

Figure from http://research.cs.tamu.edu/prism/lectures/iss/iss_l10.pdf

Dimensionality reduction

- Use prior knowledge:
  - We know that hair color has nothing to do with person's salary. We can remove hair color feature from salary prediction model

- Use feature selection:
  - Go through all features and compute its importance for the prediction
  - Gini coefficient, entropy

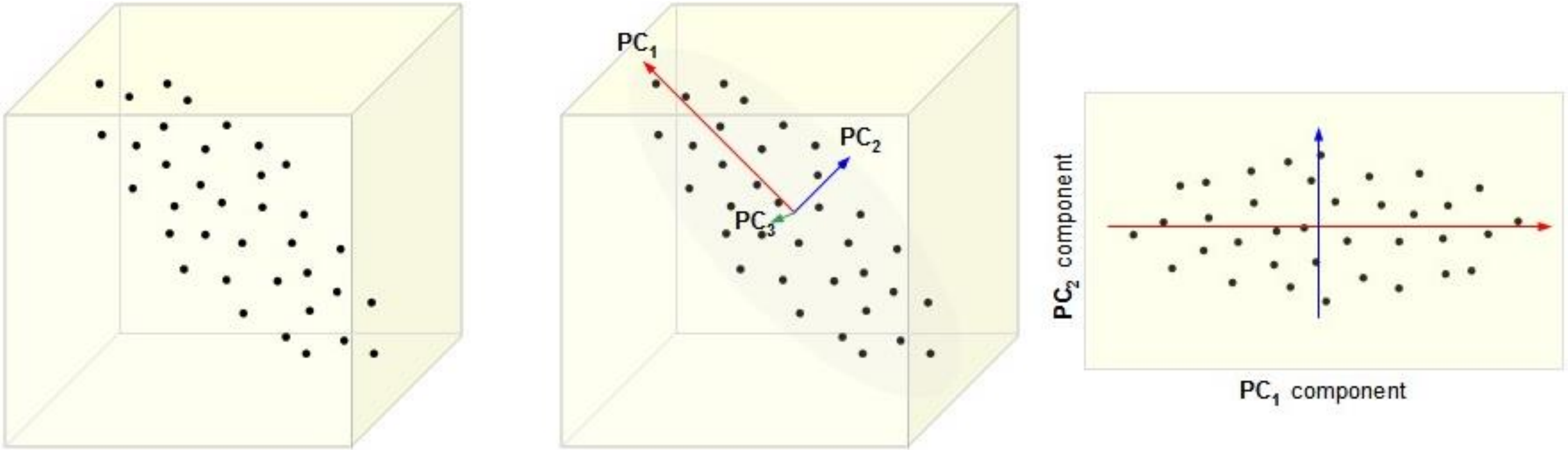
- Use feature extraction:
  - Construct new set of features $Y = \{y_1, y_2, y_K\}$ from the original set $X = \{x_1, x_2, x_N\}$, where $K << N$
  - $y_i = f_i(x_1, x_2, x_N)$

- Principal component analysis is based on the idea of feature extraction

Principal component analysis (PCA)

- Data is defined with principal components:
  - 1st component captures the direction of the greatest data variability
  - 2nd component is orthogonal to 1st and captures the greatest variability of what is left
  - …

- First m components form $f_i()$ to generate new data features:

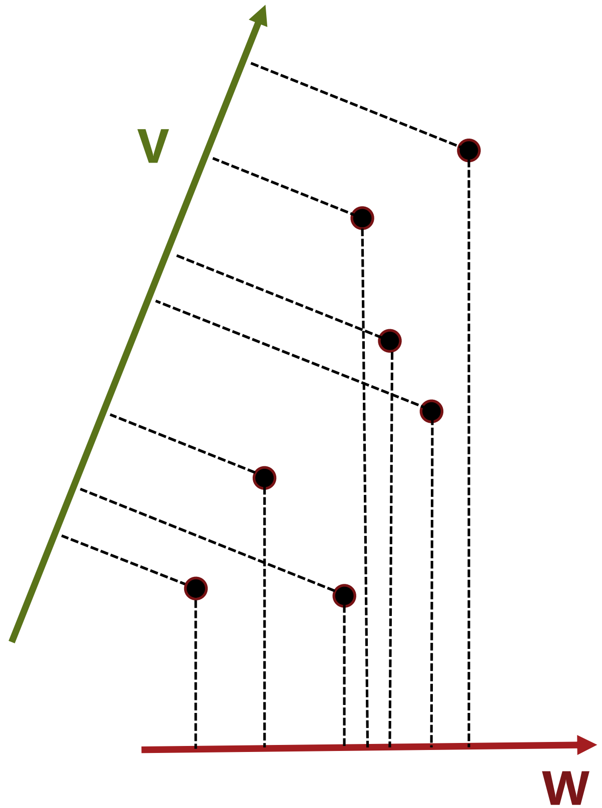


https://cnx.org/resources/269e78e2506cdbdc7f45ca17bb7d83491ce0c644/pca.jpg

PCA: maximal variance formulation

- Idea is to iteratively project the data to the direction of the maximum variance. Why do we care about maximizing it?

- Example: reduce dimensionality of 2D points to 1D:
  - Projection to V results in higher variance than projection to W
  - Projection to V preserve distances between way better
  - We would like distant objects to remain distant to preserve the logical differences between them



https://cnx.org/resources/269e78e2506cdbdc7f45ca17bb7d83491ce0c644/pca.jpg

PCA: maximal variance formulation

- Center all the data to zero mean:

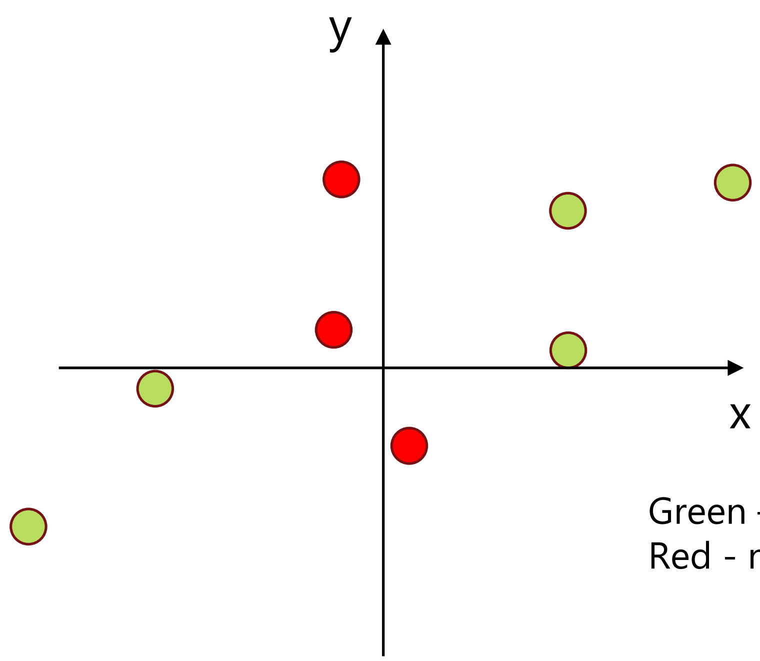
$$\mathbf{p}_\mathrm{i} = \mathbf{r}_\mathrm{i} - \bar{\mathbf{r}} \qquad \bar{\mathbf{r}} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{r}_\mathrm{i}$$

- Compute covariance matrix:
  - Covariance matrix shows if different dimensions increase/decrease together

$$\boldsymbol{\Sigma} = \mathbf{cov}\{\mathbf{r}\} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{p}\mathbf{p}^\mathrm{T}$$

$$\begin{array}{ccc} & \mathrm{x} & \mathrm{y} \\ \mathrm{x} & \multicolumn{2}{c}{\multirow{2}{*}{$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix}$}} \\ \mathrm{y} & & \end{array}$$

Positive; x and y increase and decrease together



Green – points that contribute positively to covariance
Red - negatively

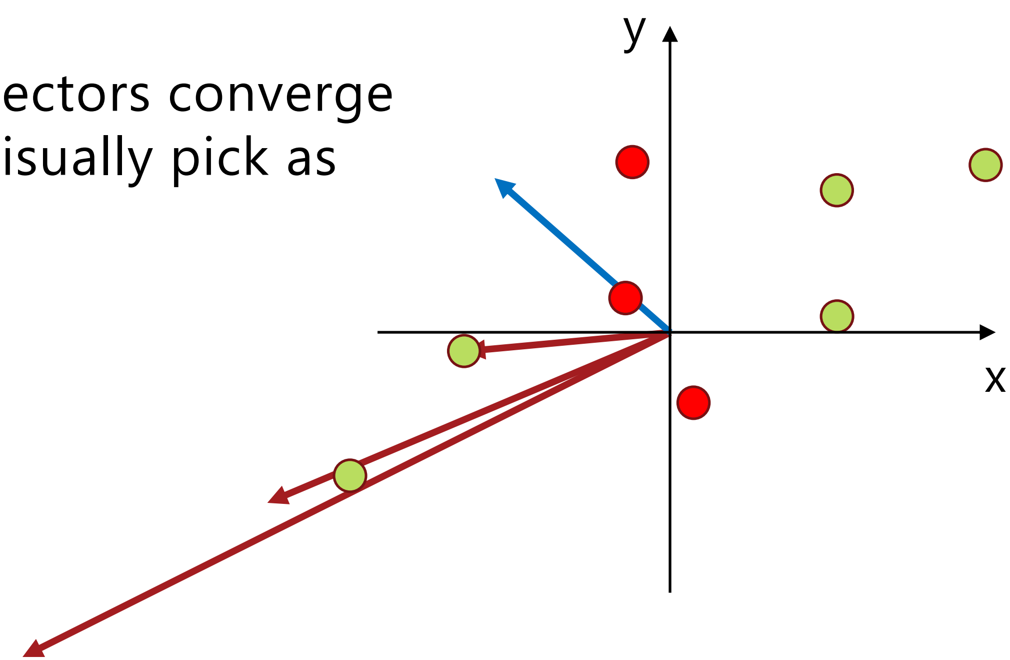
PCA: maximal variance formulation

- Let's take an arbitrary vector (-1, 1) and multiply with $\boldsymbol{\Sigma}$: $\begin{pmatrix} 2 & 0.8 \\ 0.8 & 0.6 \end{pmatrix}$

$$\begin{pmatrix} 2 & 0.8 \\ 0.8 & 0.6 \end{pmatrix}\begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} -1.2 \\ -0.2 \end{pmatrix}$$

- Let's multiply the result with $\boldsymbol{\Sigma}$ several times:
  - 1st multiplication = $\begin{pmatrix} -1.2 \\ -0.2 \end{pmatrix}$; 2nd = $\begin{pmatrix} -2.5 \\ -1.0 \end{pmatrix}$; 3rd = $\begin{pmatrix} -6.0 \\ -2.7 \end{pmatrix}$; 4th = $\begin{pmatrix} -14.1 \\ -6.4 \end{pmatrix}$; 5the = $\begin{pmatrix} -33.3 \\ -15.1 \end{pmatrix}$

- The slopes of the resulting vectors converge to the direction you would visually pick as the 1st principal component:
  - 1st slop = 0.17
  - 2nd = 0.4
  - 3rd = 0.45
  - 4th = 0.454
  - 5th = 0.454



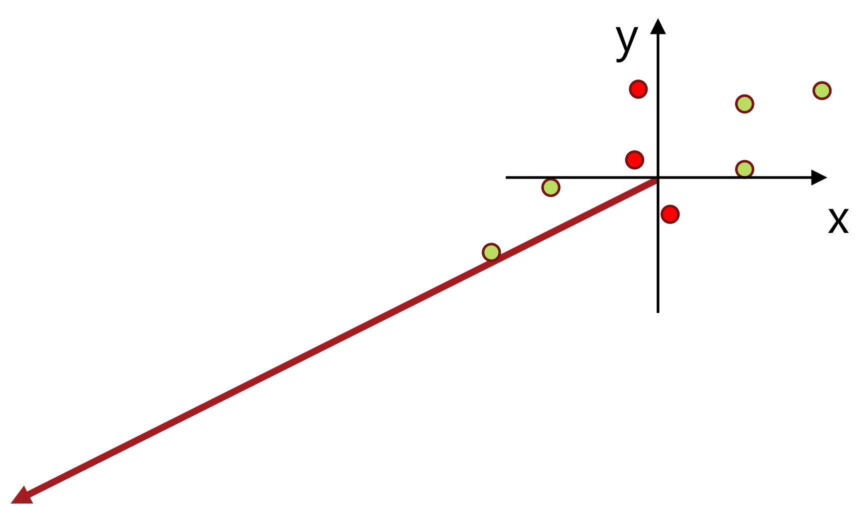
PCA: maximal variance formulation

- The eigenvectors $\mathbf{e}$ do not turn when multiplied by $\boldsymbol{\Sigma}$:

$$\boldsymbol{\Sigma}\mathbf{e} = \lambda\mathbf{e}; \qquad \|\mathbf{e}\| = 1$$

- The weighting coefficients λ are called eigenvalues; they encode contribution of the corresponding eigenvector

- From previous example
  - Vector at 5th multiplication = $\begin{pmatrix}-33.3\\-15.1\end{pmatrix}$
  - After normalization $\begin{pmatrix}-0.91\\-0.41\end{pmatrix}$
  - $\begin{pmatrix}2 & 0.8\\0.8 & 0.6\end{pmatrix}\begin{pmatrix}-0.91\\-0.41\end{pmatrix} = \begin{pmatrix}-2.15\\-0.97\end{pmatrix}$
  - $\begin{pmatrix}-2.15\\-0.97\end{pmatrix} \approx \lambda \begin{pmatrix}-0.91\\-0.41\end{pmatrix}$; λ=2.38



- The true first eigenvalue = 2.36

PCA: maximal variance formulation

- To find eigenvectors and eigenvalues, first solve $\det(\boldsymbol{\Sigma} - \lambda \mathrm{I}) = 0$:

$$\det\begin{pmatrix} 2.0-\lambda & 0.8 \\ 0.8 & 0.6-\lambda \end{pmatrix} = (2-\lambda)(0.6-\lambda) - 0.8 \cdot 0.8 = \lambda^2 - 2.6\lambda + 0.56 = 0$$

$$\{\lambda_1, \lambda_2\} = \{2.36, 0.23\}$$

- Find eigenvectors by solving $\boldsymbol{\Sigma}\mathbf{e} = \lambda\mathbf{e}$:

$$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix}\begin{pmatrix} e_{1,1} \\ e_{1,2} \end{pmatrix} = 2.36\begin{pmatrix} e_{1,1} \\ e_{1,2} \end{pmatrix} \rightarrow \begin{matrix} 2.0e_{1,1} + 0.8e_{1,2} = 2.36e_{1,1} \\ 0.8e_{1,1} + 0.6e_{1,2} = 2.36e_{1,2} \end{matrix}$$

What is the problem with this system?

This system is redundant so many solutions exists Simply multiply a solution $\mathbf{e}$ by a constant and get a new one. We can only get proportion:

$$e_{1,1} = 2.2e_{1,2}$$

- Impose condition of unit norm $||\mathbf{e}|| = 1$, and the proportion:

$$\mathbf{e} = [0.91, 0.41]$$

PCA: maximal variance formulation

- Projecting input date to the new dimension:
  - Select first M eigenvectors with the largest eigenvalues
  - Having each input points $\mathbf{r} = \{\mathbf{r}_i\}$ centered to zero mean: $\mathbf{p}_i = \mathbf{r_i} - \bar{\mathbf{r}}$
  - Transform $\boldsymbol{p}_i$ as follows:

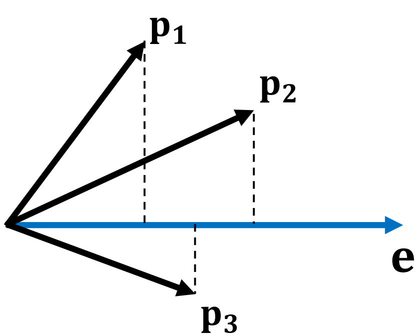
$$\begin{bmatrix} q_{i,1} \\ q_{i,2} \\ \cdots \\ q_{i,M} \end{bmatrix} = \begin{bmatrix} p_{i,1}e_{1,1} + p_{i,2}e_{1,2} + \cdots + p_{i,N}e_{1,N} \\ p_{i,1}e_{2,1} + p_{i,2}e_{2,2} + \cdots + p_{i,N}e_{2,N} \\ \cdots \\ p_{i,1}e_{M,1} + p_{i,2}e_{M,2} + \cdots + p_{i,N}e_{M,N} \end{bmatrix}$$

PCA: derivations

- Let's show that the direction of the eigenvector is actually the direction of maximal data variability:
  - Let's select vector $\mathbf{e}$, and project all data points $\mathbf{p}$ to this vector
- For $\mathbf{e}$ to the direction of maximal variance, we need to maximize:

$$\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{j=1}^{M} p_{i,j} e_j\right)^2$$

Can we simply compute the derivative to find the maximum?



- Vector $\mathbf{e}$ is not limited, so the maximum is not limited:
  - We need to add a constraint:

$$\lambda\left(\sum_{j=1}^{M} {e_j}^2 - 1\right)$$

PCA: derivations

- We want to maximize F:

$$F = \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{j=1}^{M} p_{i,j}e_j\right)^2 - \lambda\left(\sum_{j=1}^{M} {e_j}^2 - 1\right)$$

- Set all derivatives to zero:

$$\frac{\partial F}{\partial e_a} = \frac{2}{N}\sum_{i=1}^{N}\left(\sum_{j=1}^{M} p_{i,j}e_j\right)p_{i,a} - 2\lambda e_a = 0$$

- The equation can be rearranged:

$$2\sum_{j=1}^{M} e_j \underbrace{\left(\frac{1}{N}\sum_{i=1}^{N} p_{i,a}p_{i,j}\right)}_{\text{covariance of a,j}} = 2\lambda e_a$$

PCA: derivations

- For all derivatives, we get:

$$2\sum_{j=1}^{M} e_j \left(\frac{1}{N} \sum_{i=1}^{N} p_{i,a} p_{i,j} \right) = 2\lambda e_a \quad \Rightarrow \quad \left\{ \begin{array}{c} \sum_{j=1}^{M} \mathrm{cov}(1, \mathrm{j}) e_j = \lambda e_1 \\ \vdots \\ \sum_{j=1}^{M} \mathrm{cov}(\mathrm{M}, \mathrm{j}) e_j = \lambda e_M \end{array} \right\}$$

First row of cov matrix

Last row of cov matrix

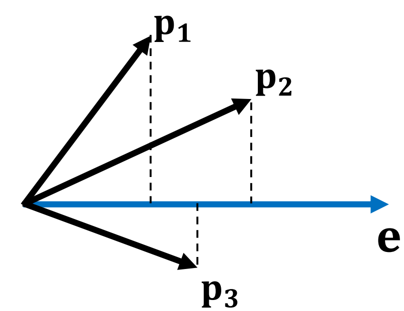
$$\Downarrow$$

$$\mathbf{\Sigma e} = \lambda \mathbf{e}$$

We come to our initial observation

PCA: derivations

- Variance of projected points ($\mathbf{p}^{\mathrm{T}}\mathbf{e}$):



$$\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{j=1}^{M} p_{i,j}e_j\right)\left(\sum_{a=1}^{M} p_{i,a}e_a\right)$$

We can drop the mean because it is zero

All components involving i

$$\sum_{a=1}^{M}\left(\sum_{j=1}^{M}\left(\frac{1}{N}\sum_{i=1}^{N} p_{i,a}p_{i,j}\right)e_j\right)e_a$$

$$\sum_{a=1}^{M}\left(\sum_{j=1}^{M}\left(\frac{1}{N}\sum_{i=1}^{N} p_{i,a}p_{i,j}\right)e_j\right)e_a$$

$$\mathrm{cov}(a,j) = \frac{1}{N}\sum_{i=1}^{N} p_{i,a}p_{i,j}$$

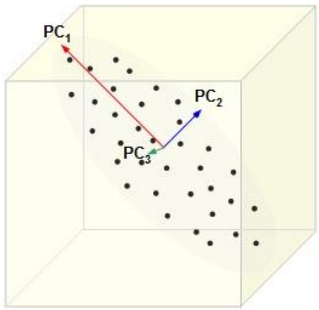
$$\sum_{j=1}^{M}\mathrm{cov}(\mathrm{a},\mathrm{j})e_j = \lambda e_a$$

$$\sum_{a=1}^{M}\lambda e_a e_a = \lambda\|\mathbf{e}\| = \lambda$$

$\mathbf{e}$ – has unit length

PCA: selecting resulting dimensions

- The number of eigenvectors equals the dimensionality of $\mathbf{p}$:
  - If we sort eigenvalues $\lambda_1, \lambda_2, \cdots \lambda_n$, every next eigenvalue will capture less and less variance in the system



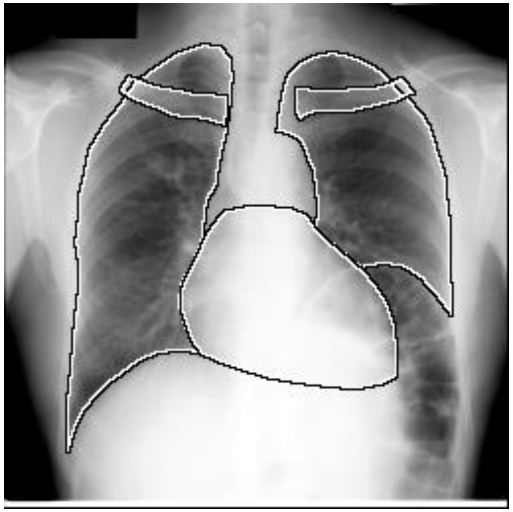
PC_3 is very short, the data is almost planar

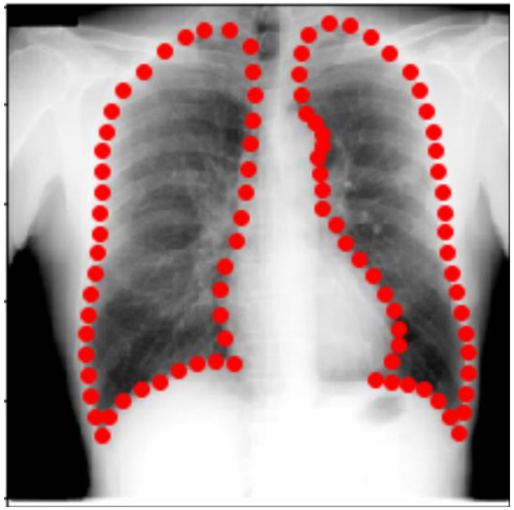
- Select the first eigenvectors for which:

$$\sum_{i=1}^{L} \frac{\lambda_i}{\sum_{j=1}^{N} \lambda_j}$$

PCA: applications

- Image analysis:
  - Morphometry of organs in medical images



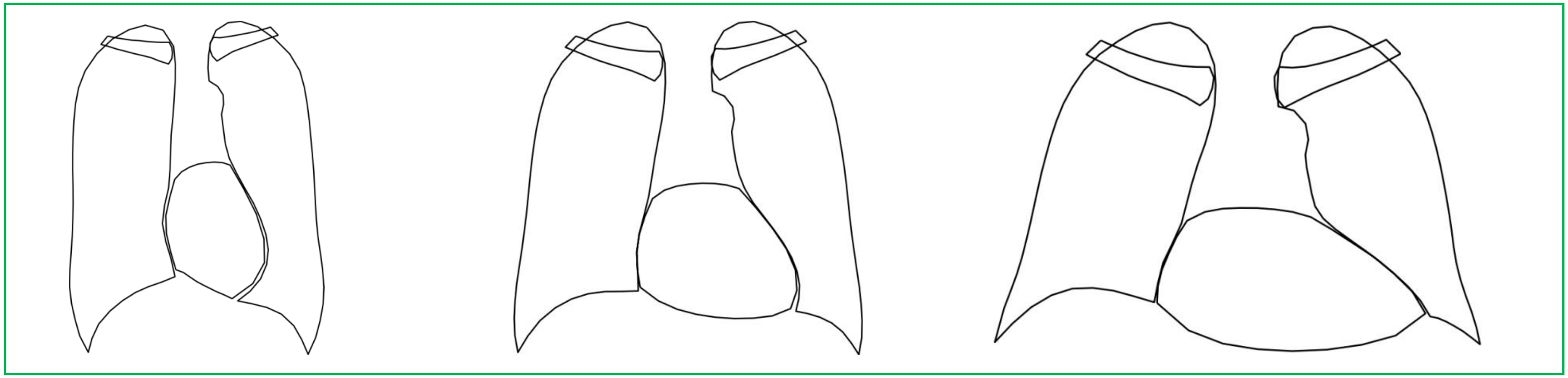


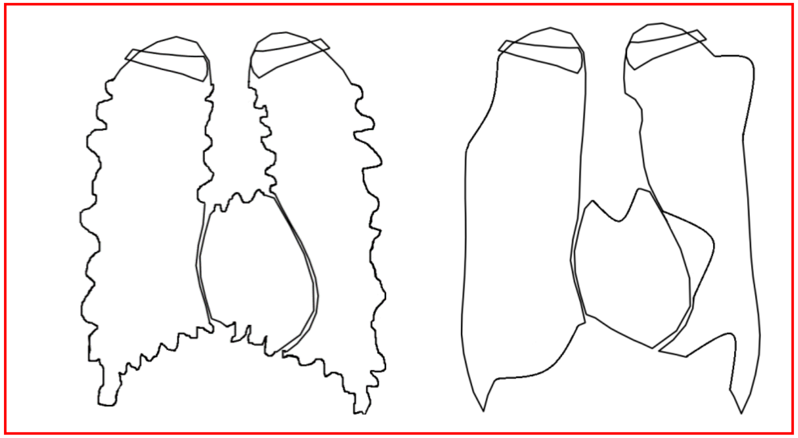
We can define the organ borders with a set of specific points, i.e. landmarks. These landmarks can be located on all images of the organ.

Database: 100 X-rays of lung fields (N = 100),
each X-ray is annotated with 92 landmarks (M = 92*2 = 184)

PCA: applications

- How can we numerically describe the shape of different lung fields?
- Can we recognize a realistic example of lung fields shape?
- Can we generate new lung fields that will look like real?



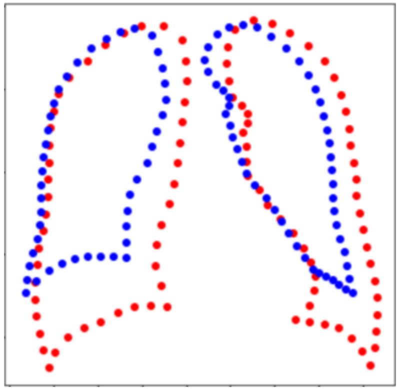


PCA: applications

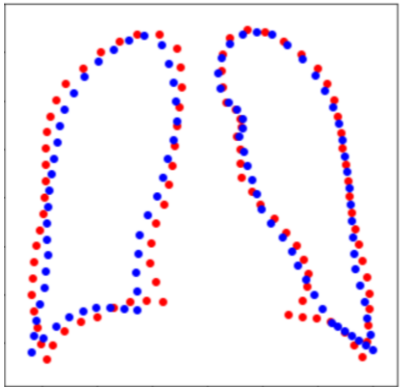
- i-th lung field shape can be described as:

$$S_i = [[x_1, y_1], [x_2, y_2], \dots, [x_{92}, y_{92}]]$$

- Perform normalization:
  - Subtract mean from each lung field shape $\bar{S}_i = [\bar{x}, \bar{y}]$
  - It is also important to normalize to scale and rotation (check Procrustes Analysis)



Two shapes before normalization



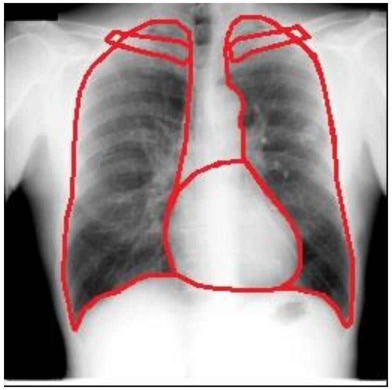
Two shapes after normalization

- We will get an array of N = 100 of M=184 dimensional samples:

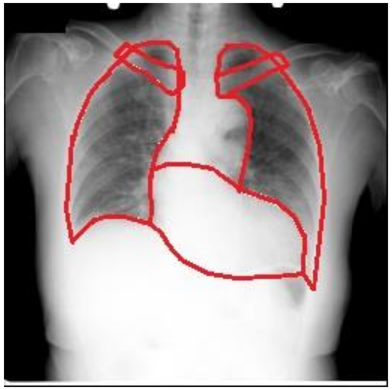
$$\mathbf{p} = \begin{bmatrix} p_{1,1} & \cdots & p_{1,M} \\ \vdots & \ddots & \vdots \\ p_{N,1} & \cdots & p_{N,M} \end{bmatrix}$$

PCA: applications

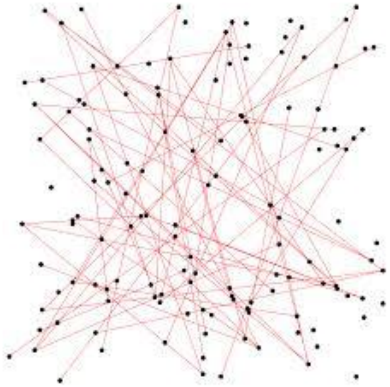
- From samples $\mathbf{p}$, we can compute eigenvectors $\mathbf{e}$ and eigenvalues λ:
  - The number of eigenvectors equals to M, i.e. the number of solutions of $\mathbf{\Sigma e} = \lambda \mathbf{e}$
  - All eigenvectors are mutually orthogonal
  - What kinds of set of M real numbers we can generate using all M eigenvectors?
- Any possible set!
- Let's try to generate these three samples:



Lung fields 1



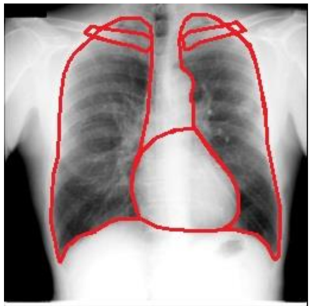
Lung fields 2



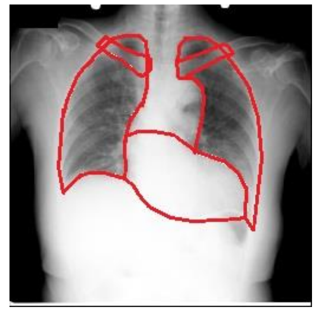
Random points

PCA: applications

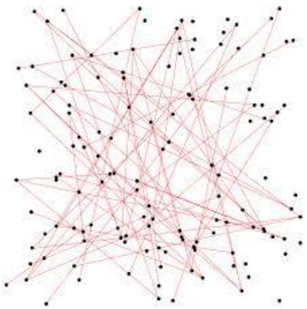
- Let's try to generate these three samples using eigenvectors:



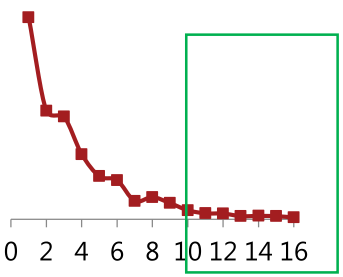
Lung fields 1

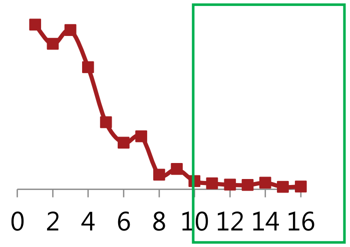


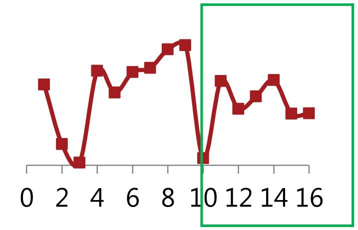
Lung fields 2



Random points





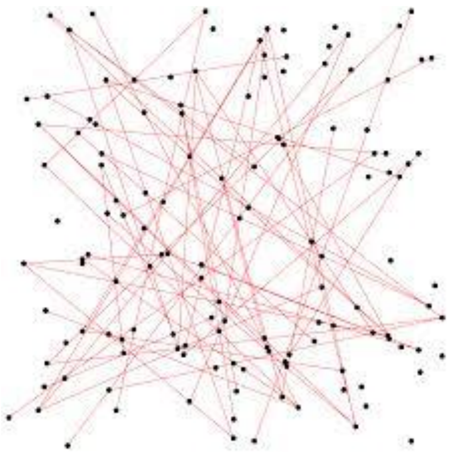


These two examples will be generated relatively well using only first eigenvectors, because last eigenvectors have low eigenvalues and contribute very little to the lung field shapes.

The coefficients of eigenvectors will be random, because points are randomly generated

PCA: applications

- If we try to fit most representative eigenvectors to random point:



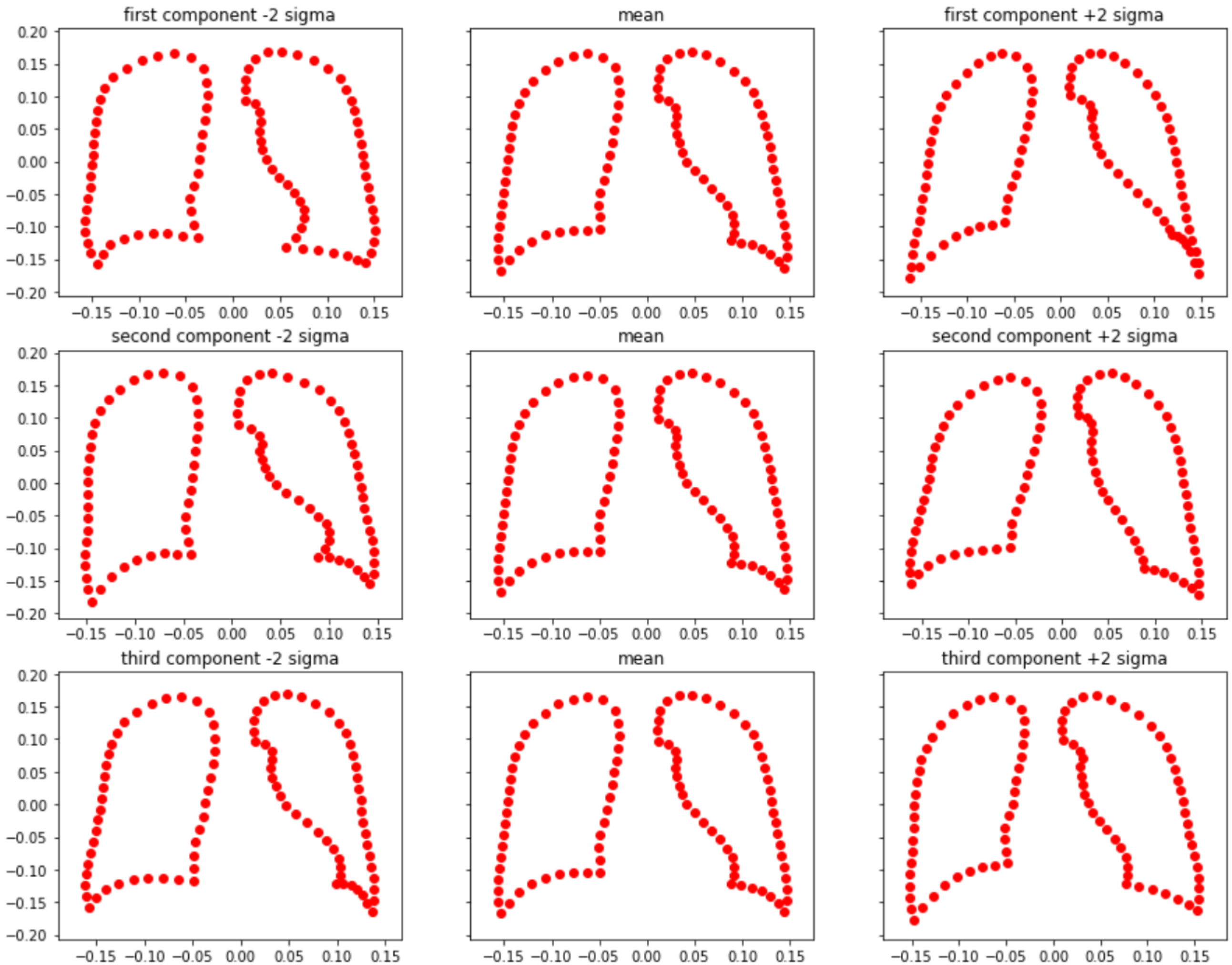
Random points

A lung fields shape that fits random points best

- So PCA give us an instrument to distinguish realistic data samples from unrealistic. If first eigenvectors are sufficient to accurately reconstruct a sample, it is likely to be a example of the target object

PCA: applications

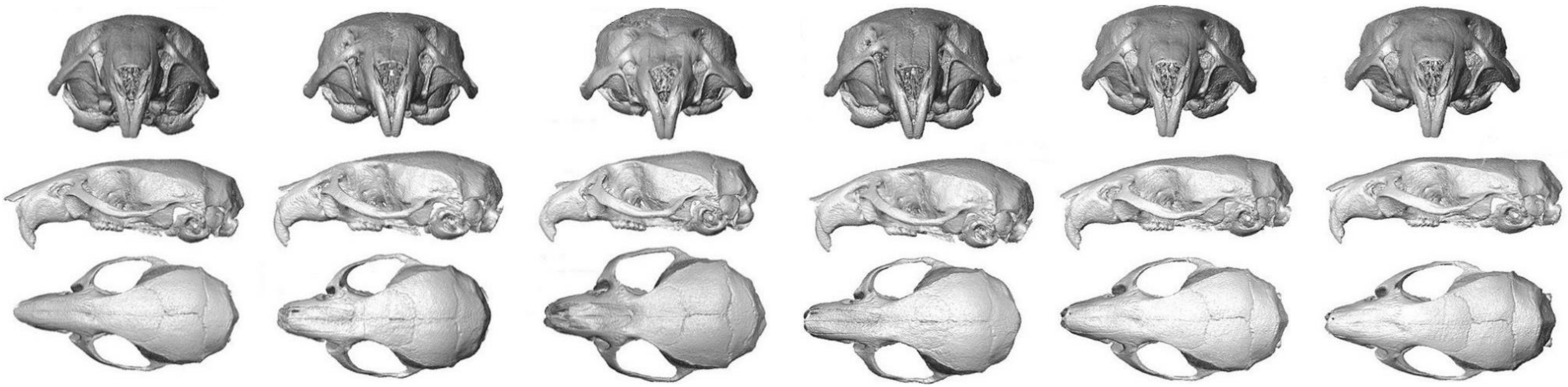
- We can use first eigenvectors to generate new object samples:



PCA: applications

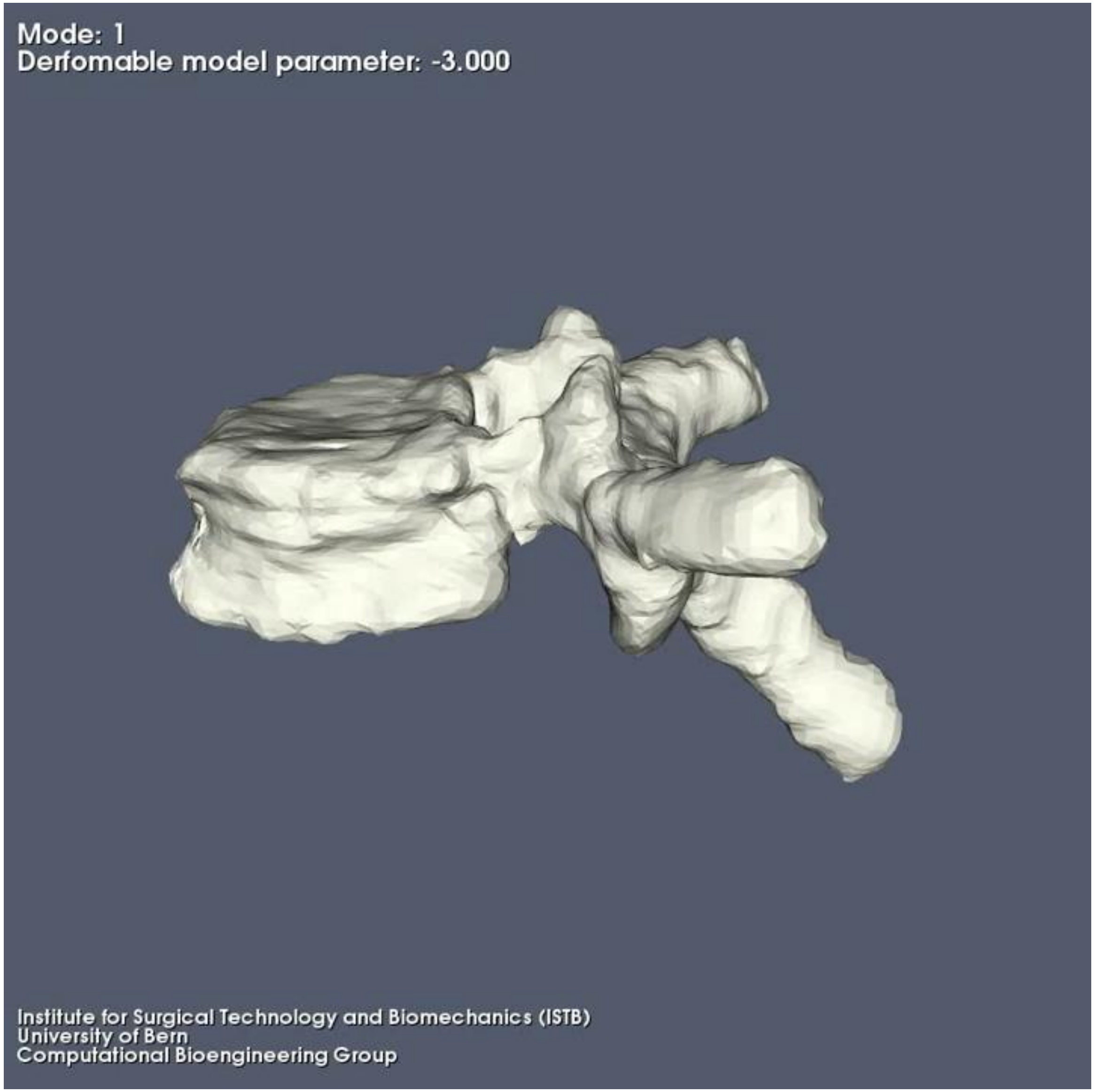
- Each eigenvector describe smooth object shape variations. So there will be no spikes that appear at random places. However, such spikes may be generated if a combination of eigenvectors with low eigenvalues are used.
- First eigenvectors usually correspond to meaningful shape variations. For example, changing an eigenvector may transform male pelvis towards female, systolic ventricle towards diastolic, etc.

Eigenvector 1 Eigenvector 2



PCA: applications

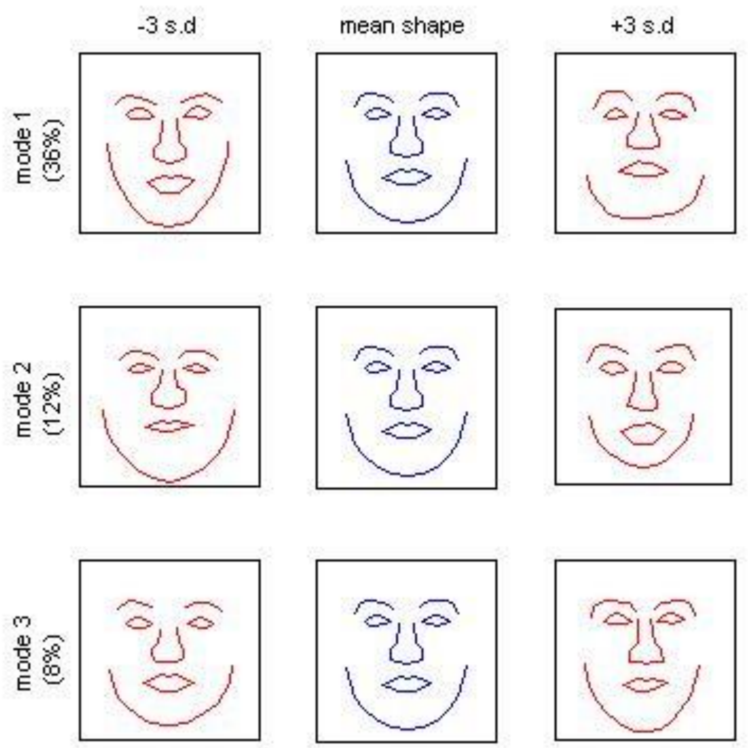
PCA: applications



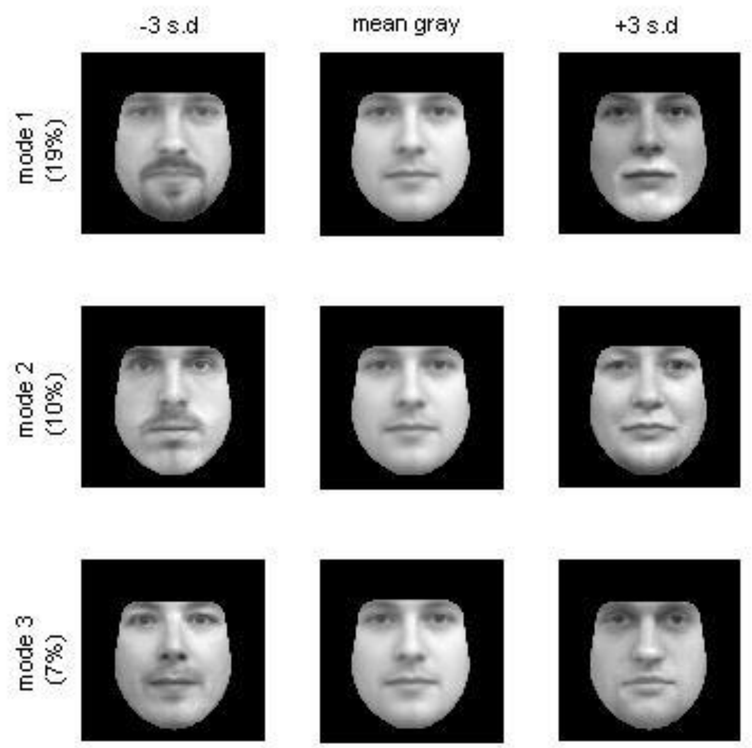
PCA: shape + appearance

PCA is not limited to landmarks, we can model both landmark positions and object intensities

Shape PCA



Intensity PCA



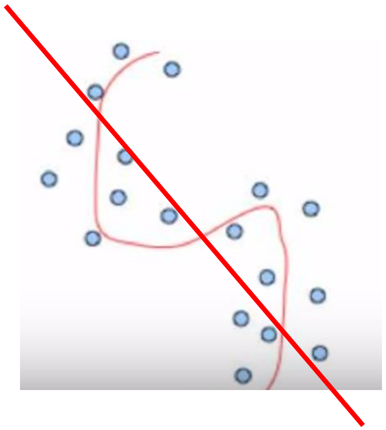
PCA: applications

What will happen if we the shape+intensity PCA:

- Train on caricature faces, and then provide real face picture modeling
- Train on male faces and then provide female for modeling

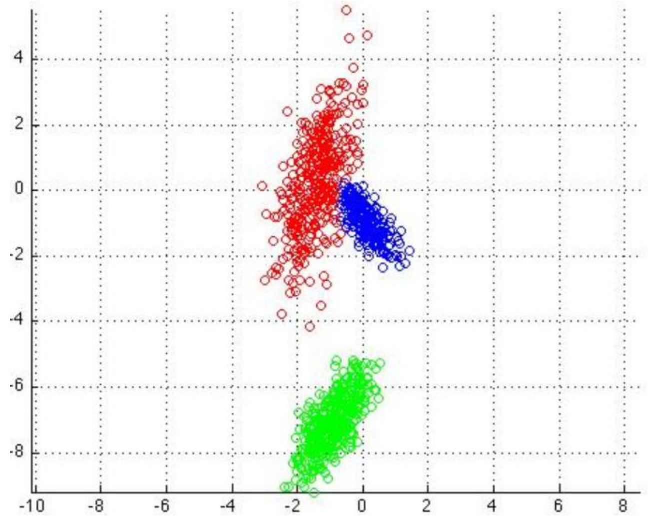
PCA: practical issues

- Underlying subspace should be linear



https://www.youtube.com/watch?v=6Ht-nIf_NKc&list=PLBv09BD7ez_5_yapAg86Od6JeeypkS4YM&index=11

- Unimodal Gaussian distribution is assumed



Questions?