Universität Stuttgart
KI – Institute for Artificial Intelligence
Analytic Computing

Machine Learning
3 Dimensionality Reduction & Clustering

Prof. Dr. Steffen Staab
Nadeen Fatallah
Daniel Frank
Akram Sadat Hosseini
Jiaxin Pan
Osama Mohamed
Arvindh Arunbabu
Tim Schneider
Yi Wang

https://www.ki.uni-stuttgart.de/

Learning Objectives

- What is dimensionality reduction?
  - How does principal component analysis work?
- What is clustering?
- How can we evaluate clustering?
- What are intrinsic and extrinsic evaluation measures?
- How does K-Means work?
- How to choose k for K-Means?
- What is the EM algorithm?

1 Motivation

Cf.

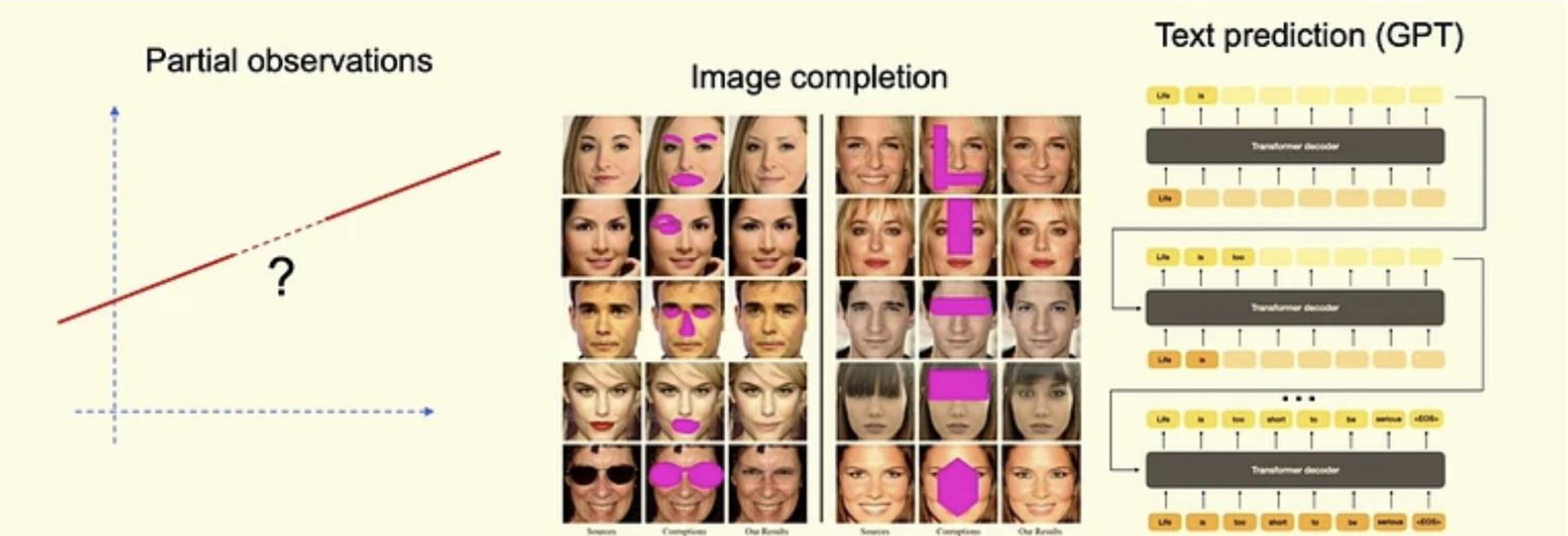
Ma, Y., Tsao, D., & Shum, H. Y. (2022). On the principles of parsimony and self-consistency for the emergence of intelligence. *Frontiers of Information Technology & Electronic Engineering*, *23*(9), 1298-1323.

Why can humans predict at all?

- Input to human visual cortex (estimated):
  - 1 MPixel to 500 Mpixel
  - 10 Mbit/s
- Predictions:
  - where the ball will be going, how to catch it…
  - whether it will be raining in few minutes…
  - whether the person you talk to likes you…
- **The world is not entirely random and predictable at large!**

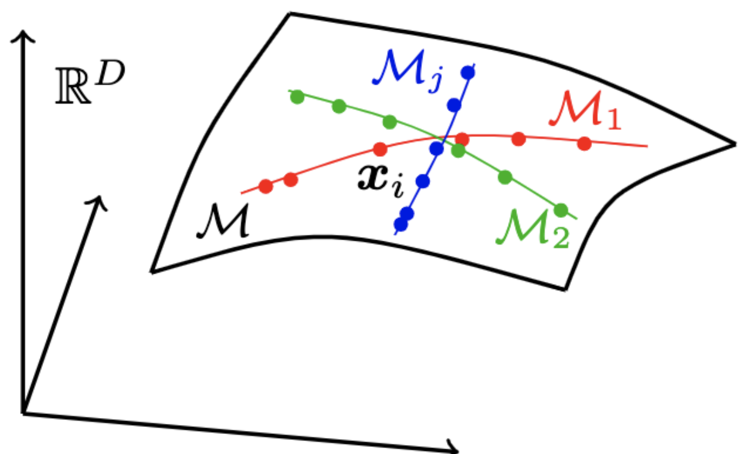
Key hypothesis in (machine) learning

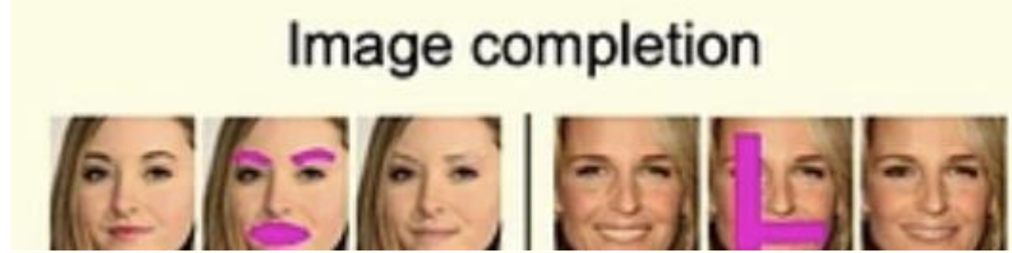
- Human **experiences** and **predictions** are low-dimensional
  - others is noise
  - others is missing observation
  - intelligence fights against entropy/noise/diffusion



The mathematics of predictions

The **mathematics of predictions** are probability distributions $P(X)$
of **low-dimensional support** (so called manifolds)
in **observed high-dimensional data space**



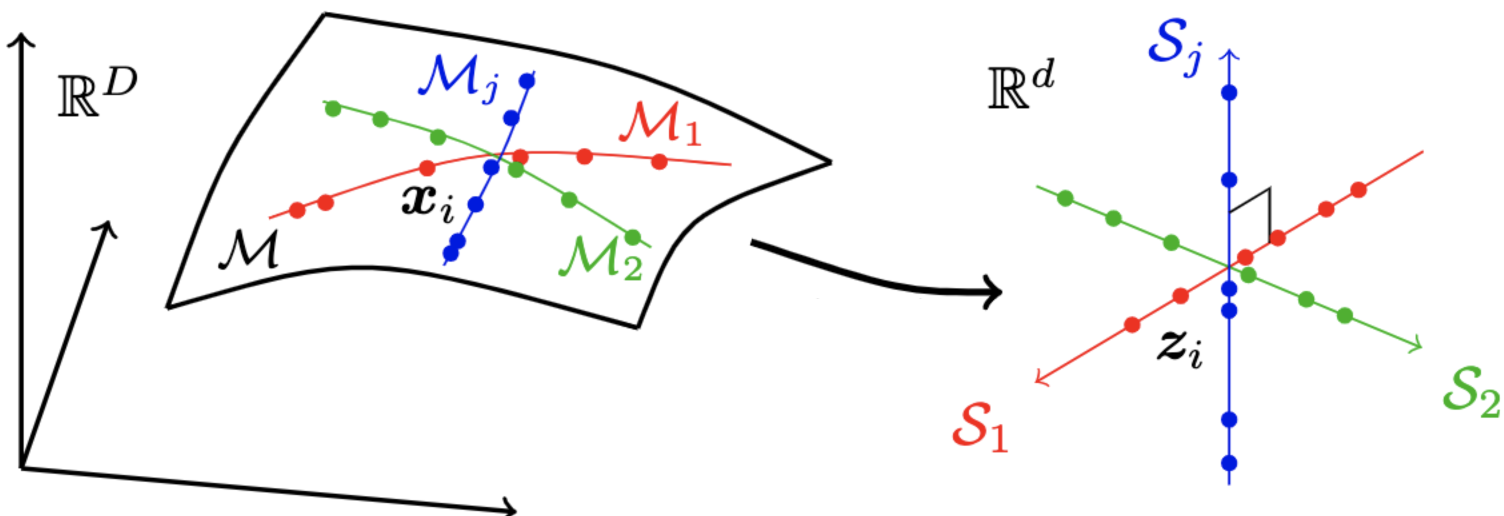


Ma, Y., Tsao, D., & Shum, H. Y. (2022). On the principles of parsimony and self-consistency for the emergence of intelligence. *Frontiers of Information Technology & Electronic Engineering*, *23*(9), 1298-1323.

What should humans or machines learn?

Principle of parsimony

- Learn what is predictable
- Learn a low-dimensional, latent representation
  - high (D) to low (d) dimensionality: 10^4 to 10^{10} $\rightarrow 10^0$ to 10^3
  - from feature space to latent space/embedding space



How to reduce dimensionality?

- from dimensionality reduction techniques
  - PCA [Pearson 1901] and others
- clustering (1930s)
- autoencoders (1980s)
- LSA (1989), PLSA, LDA
- language models (2010s)
- diffusion models (2020s)

2 Dimensionality Reduction

Cf.

Kevin P. Murphy

Probabilistic Machine Learning. An Introduction

book1.pdf, Chapter 20

Problem: High dimensionality of data

- Text documents are often represented as bag-of-words
  - For each document count how often each term occurs: $\mathrm{tf}(d, t)$

  → Each document is represented by a vector in $\mathbb{R}^{10,000}$, $\mathbb{R}^{10^5}$ or more
- Images are represented as vectors of RGB pixels, e.g. $\mathbb{R}^{3\times 200\times 300}$ or more
- Relational databases have
  - tables with 10^2 columns, 10^2 or 10^3 tables

  → Universal relation with dimensionality 10^4

Idea: identify the most important dimensions

- to understand the data
- to remove redundancies in the data
- to simplify the machine learning problem and make ML more effective
- to run machine learning algorithms more efficiently

(Semi-)Formal problem of dimensionality reduction

- Given a data set $\mathcal{D} = \{x_1, \dots, x_N\} \subseteq \mathbb{R}^D$
- Find a function $\hat{f}\colon \mathbb{R}^D \to \mathbb{R}^d$, with $d \ll D$
  to derive a data set $\widehat{\mathcal{D}} = \{\hat{f}(x_1), \dots, \hat{f}(x_N)\} \subseteq \mathbb{R}^d$
- such that the main characteristics of $\mathcal{D}$ are preserved
- If the "main characteristics" are formalized,
  we have formalized an optimization problem

$$\mathcal{L}(\hat{f}) = \sum_{i=1}^{N} \left(x_i - \hat{f}^{-1}\left(\hat{f}(x_i)\right) \right)^2$$

Dimensionality reduction techniques

varying in their definition of the loss $\mathcal{L}$

- **Principal component analysis**
- Factor analysis (generalization of PCA)
- Latent discriminant analysis (LDA)
  - not to be confused with the clustering technique Latent Dirichlet assignment (LDA)
- Independent component analysis (ICA)
- Latent semantic analysis (LSA)
- (Non-negative) matrix factorization
- t-distributed stochastic neighbor embedding (t-SNE)
- topological data analysis (TDA)
- self-organizing maps (SOM / SOFM)
- …

Know your data to judge whether/which dimensionality reduction techniques make sense on your data

Borderline between dimensionality reduction techniques and clustering is permeable

3 Principal Component Analysis (PCA)

Reminder: some linear transformations

- Given vector $x = (x_1\ x_2)^\intercal \in \mathbb{R}^2$
  - Scale by $s_1, s_2 \in \mathbb{R}$: $\begin{pmatrix} s_1 & 0 \\ 0 & s_2 \end{pmatrix} x = \begin{pmatrix} s_1 x_1 \\ s_2 x_2 \end{pmatrix}$
  - Rotate by φ degree: $\begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix} x$
  - Project onto one axis: $\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} x = \begin{pmatrix} x_1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} x = \begin{pmatrix} 0 \\ x_2 \end{pmatrix}$
- Given vectors $x, t \in \mathbb{R}^2$
  - Translate x by t: $x + t$
    - Can be captured in a matrix with homogeneous coordinates
    - $\begin{pmatrix} 1 & 0 & t_1 \\ 0 & 1 & t_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} = \begin{pmatrix} x_1 + t_1 \\ x_2 + t_2 \\ 1 \end{pmatrix}$

Reminder: Sample covariance matrix

- Given a data set $\mathcal{D} = \{x_1, \dots, x_N\} \subseteq \mathbb{R}^D$
- Represent as $N \times D$ matrix $\boldsymbol{X} = \begin{pmatrix} x_{1,1} & \cdots & x_{1,D} \\ \vdots & \ddots & \cdots \\ x_{N,1} & \cdots & x_{N,D} \end{pmatrix}$
- Then the sample covariance matrix $\hat{\Sigma} = [q_{j,k}] \in \mathbb{R}^{D \times D}$ is defined by

$$q_{j,k} = \frac{1}{N-1} \sum_{i=1}^{N} (x_{i,j} - \bar{x}_j)(x_{i,k} - \bar{x}_k)$$

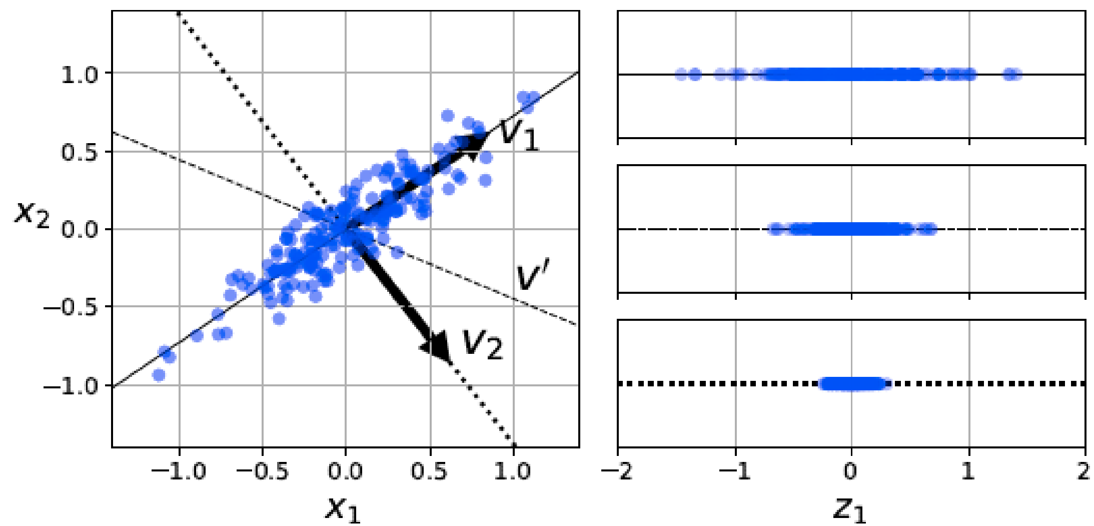
PCA

- Assume that $\hat{f}$ is linear: $\hat{f}(\mathrm{x}) = \mathrm{Vx}$
- Then the loss is

$$\mathcal{L}\left(\widehat{f_V}\right) = \mathcal{L}(V) = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - \widehat{f_V}(x_i)\right)^2$$

Idea of PCA

- Given a data set $\mathcal{D} = \{x_1, \dots, x_N\} \subseteq \mathbb{R}^D$
- Assume that $\sum_{i=1}^{N} x_i = \mathbf{0}$, by translating the data
- Represent as $N \times D$ matrix $\boldsymbol{X} = \begin{pmatrix} x_{1,1} & \cdots & x_{1,D} \\ \vdots & \ddots & \cdots \\ x_{N,1} & \cdots & x_{N,D} \end{pmatrix}$
- **Idea:**
  1. Rotate and project data onto the **most important dimension**
  2. Subtract “this part”
  3. Repeat from 1.



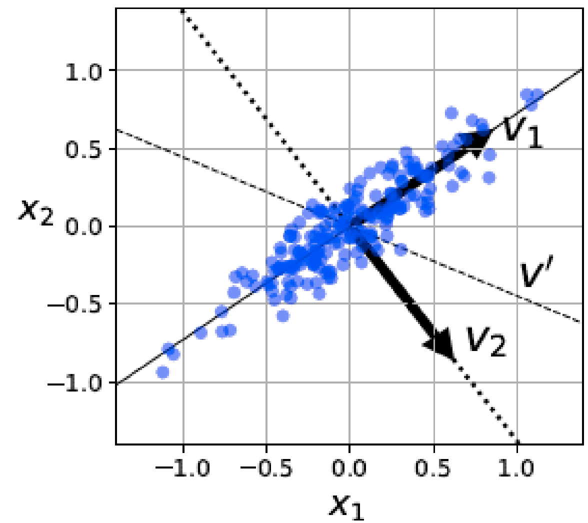
Finding the best d axes

Find unit vectors $v_1, v_2, \dots, v_d \in \mathbb{R}^D$ and new data coordinates $z_1, \dots, z_d \in \mathbb{R}^d$

$$Z = \begin{pmatrix} z_1^\top \\ \vdots \\ z_d^\top \end{pmatrix}, V = \begin{pmatrix} v_1^\top \\ \vdots \\ v_d^\top \end{pmatrix}$$

$$\mathcal{L}(V, Z) = \frac{1}{N} \sum_{i=1}^{N} \|x_i - V z_i\|^2$$

Find v_1 and all $z_{i,1}$



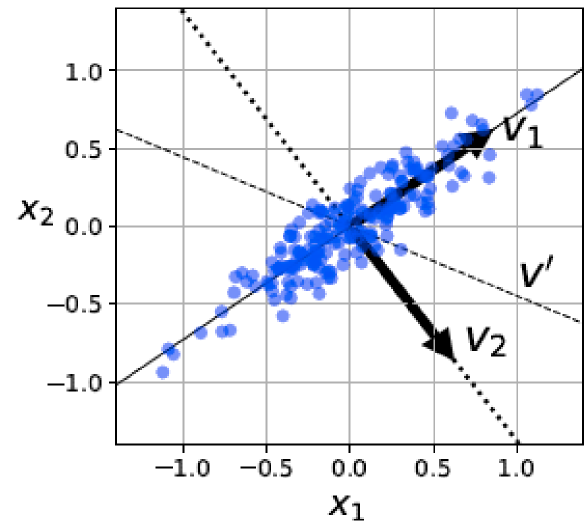
Finding the first axis

Find unit vectors $v_1, v_2, \dots, v_d \in \mathbb{R}^D$, $\|v_i\| = 1$ and new data coordinates $z_1, \dots, z_d \in \mathbb{R}^d$

Find v_1 and all $z_{i,1}$

$$\mathcal{L}(v_1, z_{\cdot,1}) = \frac{1}{N}\sum_{i=1}^{N}\left\|x_i - z_{i,1}v_1\right\|^2 =$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left[x_i^\top x_i - 2z_{i,1}v_1^\top x_i + z_{i,1}^2\right]$$



Optimize wrt $z_{i,1}$:

$$\frac{\partial}{\partial z_{i,1}}\mathcal{L}(v_1, z_{i,1}) = 0 \implies z_{i,1} = v_1^\top x_i$$

(orthogonal projection onto v_1)

Finding the first axis

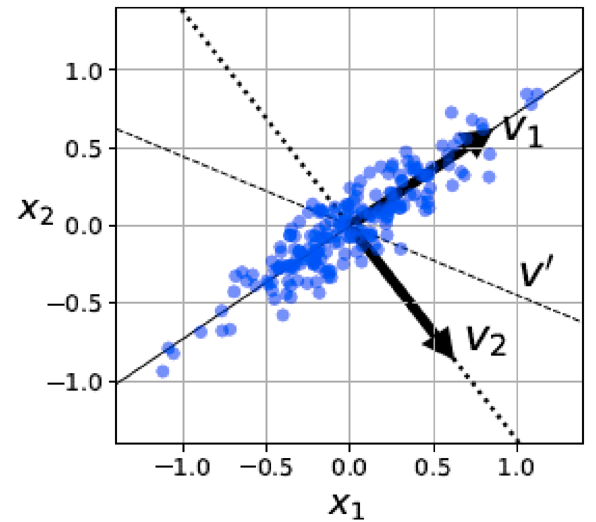
Find unit vectors $v_1, v_2, \ldots, v_d \in \mathbb{R}^D$, $\|v_i\| = 1$ and new data coordinates $z_1, \ldots, z_d \in \mathbb{R}^d$

$$\mathcal{L}(v_1, z_{\cdot,1}) = \mathcal{L}(v_1) = \frac{1}{N}\sum_{i=1}^{N}\left[x_i^\intercal x_i - 2z_{i,1}v_1^\intercal x_i + z_{i,1}^2\right] =$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left[x_i^\intercal x_i - 2v_1^\intercal x_i v_1^\intercal x_i + (v_1^\intercal x_i)^2\right] =$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left[x_i^\intercal x_i - (v_1^\intercal x_i)^2\right] =$$

$$= const - \frac{1}{N}\sum_{i=1}^{N} v_1^\intercal x_i x_i^\intercal v_1 = -v_1^\intercal \hat{\Sigma} v_1$$



$\hat{\Sigma}$ is the empirical co-variance matrix of $\boldsymbol{X}$

Optimize $\mathcal{L}(v_1) = -v_1^\intercal \hat{\Sigma} v_1$, given $\|v_1\| = 1$

Finding the first axis

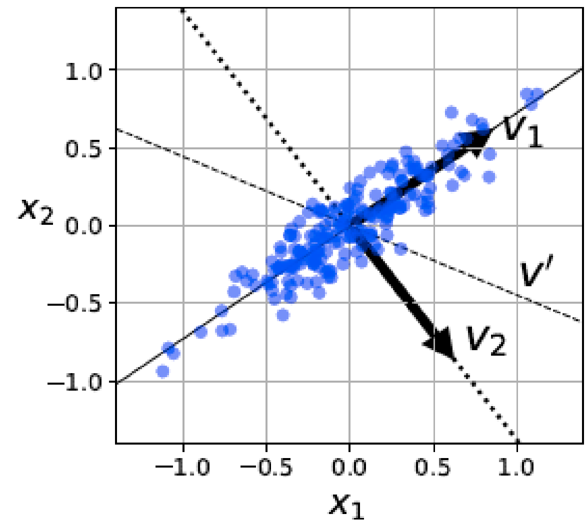
Find unit vectors $v_1, v_2, \dots, v_d \in \mathbb{R}^D$, $\|v_i\| = 1$ and new data coordinates $z_1, \dots, z_d \in \mathbb{R}^d$

Optimize $\mathcal{L}(v_1) = -v_1^\intercal \hat{\Sigma} v_1$, given $\|v_1\| = 1$

Optimize $\tilde{\mathcal{L}}(v_1) = v_1^\intercal \hat{\Sigma} v_1 - \lambda_1 (v_1^\intercal v_1 - 1)$

Optimize by $\frac{\partial}{\partial v_1} \tilde{\mathcal{L}}(v_1) = 0 = 2\hat{\Sigma} v_1 - 2\lambda_1 v_1 \Leftrightarrow$

$$\hat{\Sigma} v_1 = \lambda_1 v_1$$



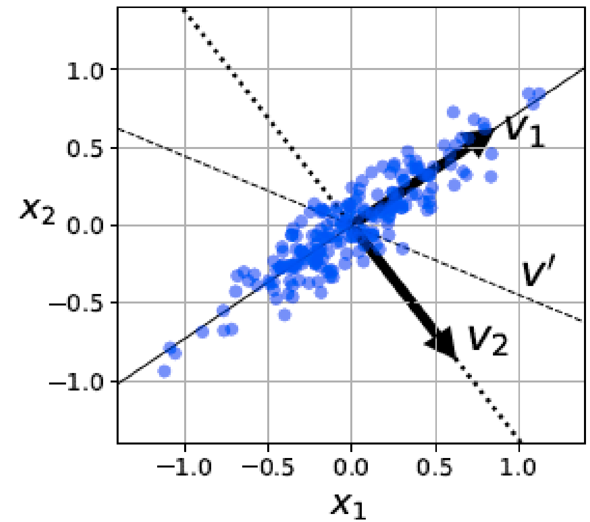
Eigenvalue problem of $\hat{\Sigma} = X^\intercal X$

The eigenvector corresponding to the largest eigenvalue signifies the axis with largest variance

λ_1 is a Langrange multiplier used for optimization under constraints

Induction step

- Orthonormal basis $v_1, v_2, \dots, v_d \in \mathbb{R}^D$, $\|v_i\| = 1$ means that $v_i \cdot v_j = 0$ for $i \neq j$



$$\mathcal{L}(v_1, z_{\cdot,1}, v_2, z_{\cdot,2}) = \frac{1}{N}\sum_{i=1}^{N} \left\| x_i - z_{i,1} v_1 - z_{i,2} v_2 \right\|^2 =$$

$$= \mathcal{L}(v_2) = \frac{1}{N}\sum_{i=1}^{N} \left[x_i^\intercal x_i - v_1^\intercal x_i^\intercal x_i v_1 \ - v_2^\intercal x_i^\intercal x_i v_2 \ \right]$$

Optimization leads to second largest eigenvalue and corresponding eigenvector

Covariance matrix vs correlation matrix

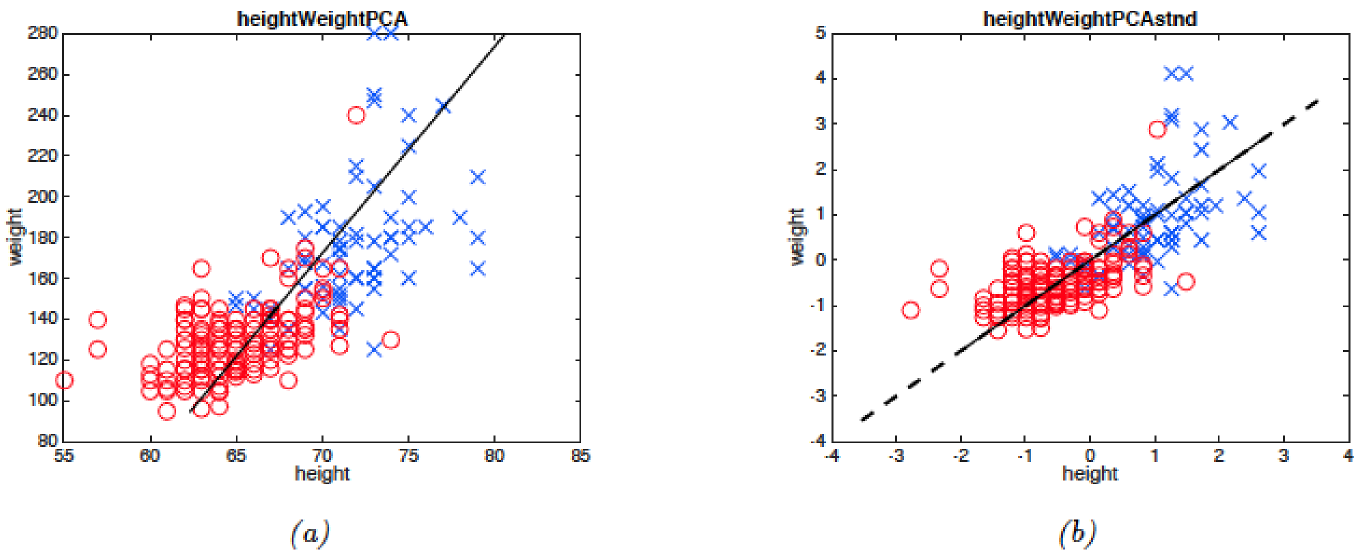
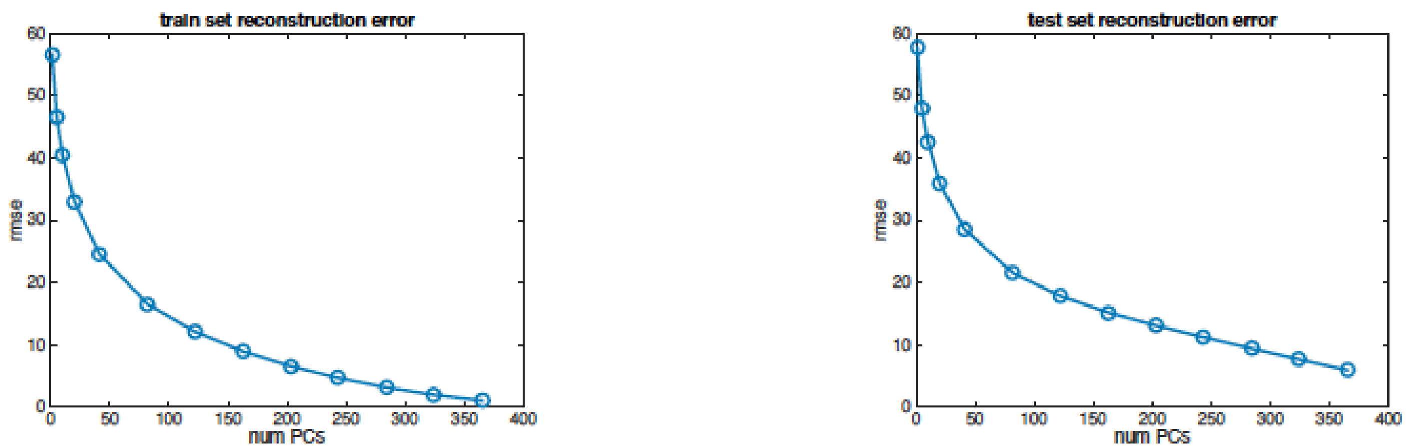


Figure 20.5: Effect of standardization on PCA applied to the height/weight dataset. (Red=female, blue=male.) Left: PCA of raw data. Right: PCA of standardized data. Generated by pcaStandardization.ipynb.

Check reconstruction error

for different numbers of dimension d

$$\mathcal{L}(V) = \frac{1}{N}\sum_{i=1}^{N} \|x_i - V^\top x_i\|^2$$



(a) *(b)*

Figure 20.6: Reconstruction error on MNIST vs number of latent dimensions used by PCA. (a) Training set. (b) Test set. Generated by pcaOverfitDemo.ipynb.

4 Cluster Analysis

Comparison of Supervised and Unsupervised ML

Supervised Machine Learning

- Regression – Learning:

$$\hat{y} = \hat{f}(x)$$

- Classification - Learning:

$$\hat{y} = \hat{f}(x) = \operatorname*{argmax}_{y} P(y|x)$$

- Desired output:
  - How to classify?
  - (what makes the model classify?)
- Issues:
  - which model?
  - which loss function?
  - which solving?
  - which evaluation measures?

Unsupervised Machine Learning

- Learning:

$$\hat{f}(x) = P_\theta(x)$$

- Desired output:
  - Where do we find data?
  - What are the parameters θ that determine this distribution?
- Issues:
  - which model?
  - which loss function?
  - which solving?
  - which evaluation measures?

Examples


User Profiles

Example purpose: **marketing**

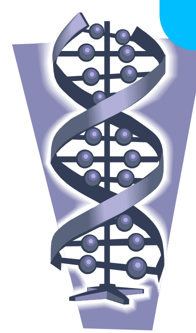

Web document templates

Data pre-processing e.g. for information extraction

Find structure in scientific data

infor-mation manage ment

Documents



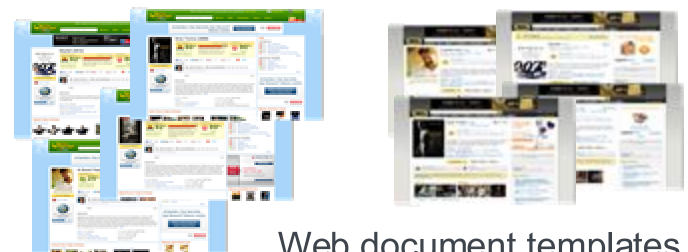
Genetic sequences

Language dialects

Goal of clustering

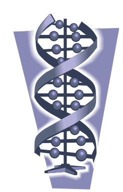
- Identification of a finite set of *clusters* (= categories, “classes“, groups) in the data
- Objects in the same cluster should be as *similar* as possible.
- Objects in different clusters should be as *dissimilar* as possible
- *“Unsupervised learning”* => no groups given

User profiles



Web document templates

Documents

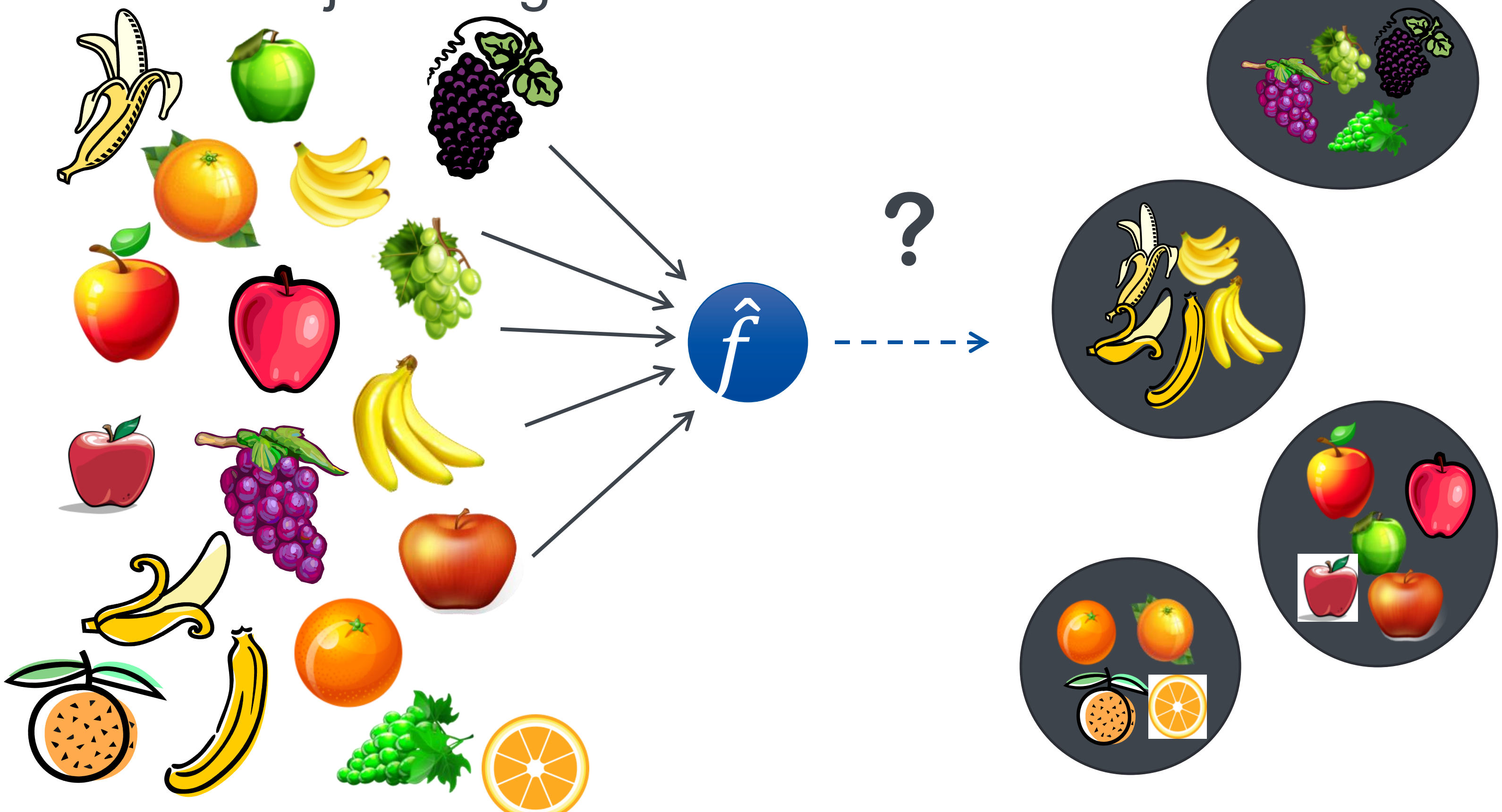


Genetic sequences

Language dialects

Clustering

- Given a set of objects, find a function $\hat{f}$ to group similar objects together



What does *similar* mean?

Clustering Task

- Objects
  - $X = \{x_1, x_2, \dots, x_N\}$
- An object is characterized by attributes
  - $x_i = (x_{i,1}\ x_{i,2} \dots x_{i,m})^\intercal$

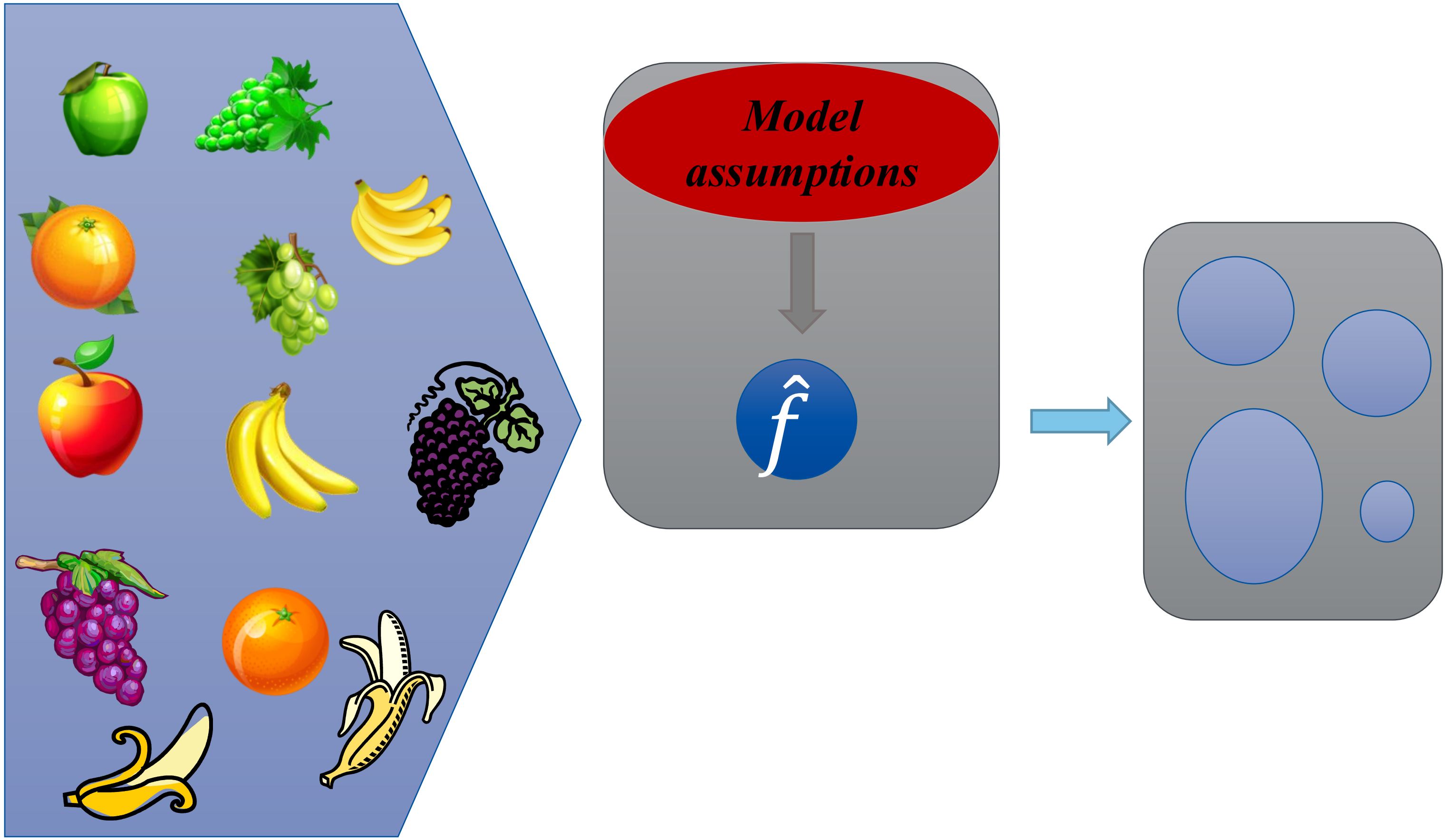
(green, round, even)

(orange, round, rough)

- Task:
  - Find groups
    - $\Omega = \{\omega_1, \omega_2, \dots, \omega_K\}$
  - Find function
    - $\hat{f} : X \to \Omega$

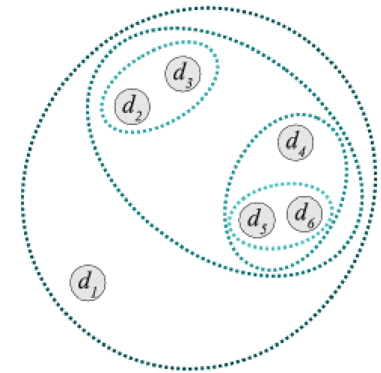
Difference to classification:
Groups are not given!

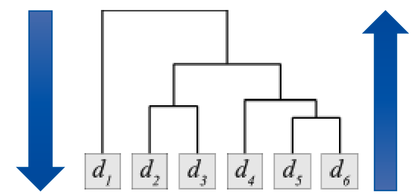
Task



Variations of the Task

- Cluster types:
  - Flat vs. Hierarchical
  - Exclusive vs. Multiple clusters
    - $\hat{f} : X \to \wp(\Omega)$

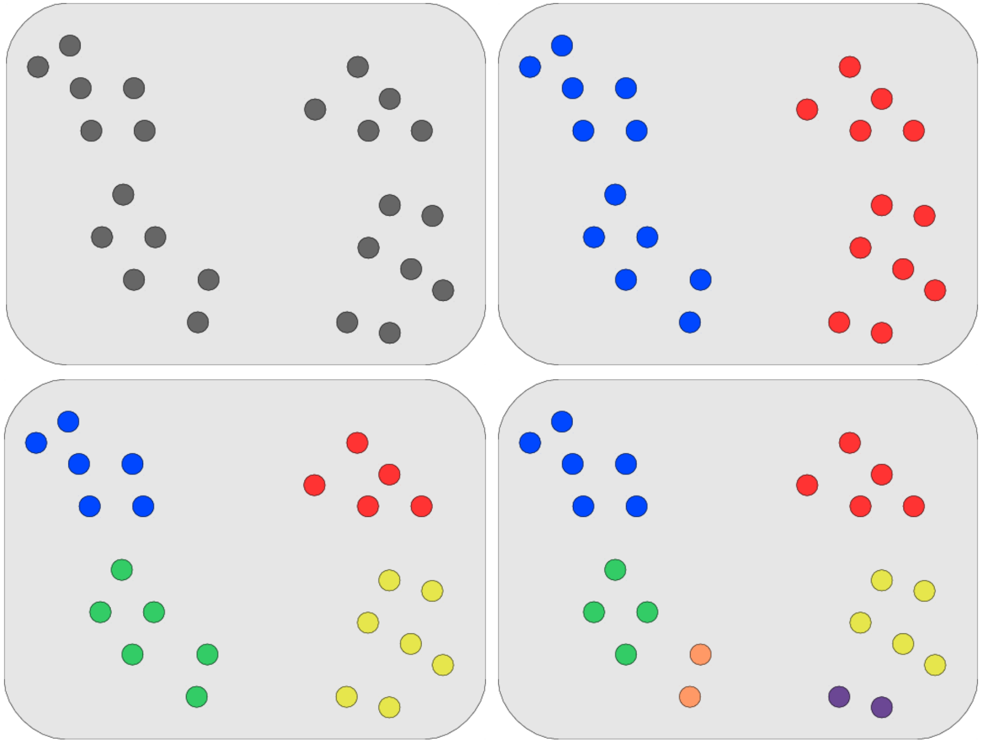




- Function $\hat{f}$
  - Hard vs. Soft assignments
    - $\hat{f} : X \to \mathbb{R}^{|\Omega|}$
  - Based on shape, density, estimates of distribution mixture

Cardinality / Number of clusters

- Provided (externally) or
- to be defined (given explicit hyperparameter) or
- to be found over the data (given other hyperparameters, e.g. density or density distribution).



5 Intrinsic Metrics
for the Evaluation of Clustering Results

Before you start to develop or use an ML algorithm,

you must know (no excuses!):

- Structure of input (input datatypes)
  - know **complete, real** examples
- Structure of output (output datatypes)
  - know **complete, real** output for your real examples

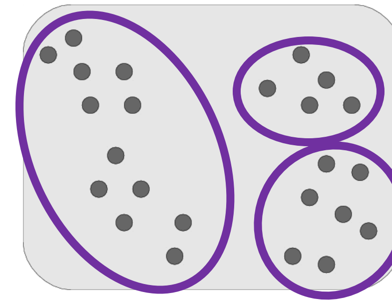
defines the algorithms that can be applied

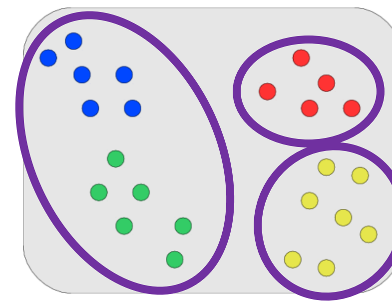
- how to evaluate whether your algorithm
  - is better than stupid baselines
    - e.g. choosing majority category, random decisions, etc.
  - is better than existing algorithms

A whole graduate course – great overview of topics in evaluation:
https://www.argmin.net/p/machine-learning-evaluation-631

Evaluation

- Intrinsic
  - Evaluate quality of clusters directly
  - E.g. compactness, separation of groups, etc.
- Extrinsic
  - Employ external knowledge
    - Ground truth from classification data
    - Assuming categories to be optimal clusters
  - Compare found clusters and pre-defined clusters
    - Difficulty of finding a matching
- Indirect
  - User testing (satisfaction, task performance)
  - Application specific metrics





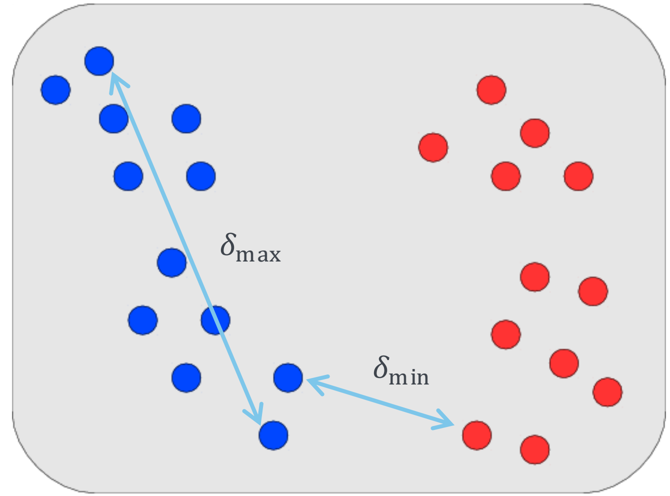
Intrinsic metrics

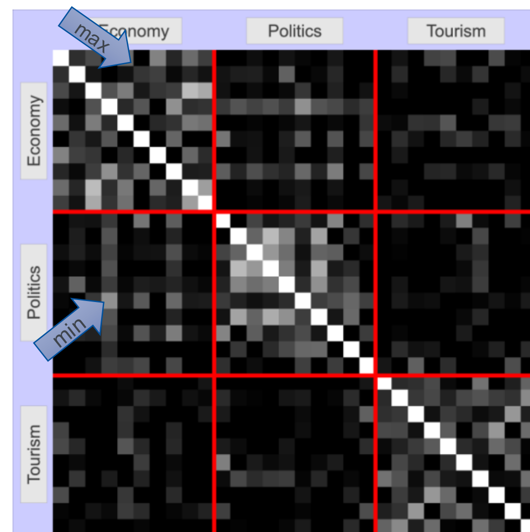
- Dunn Index
  - Notion of cluster separation

$$I_{\text{Dunn}}(\Omega) = \frac{\delta_{\min}}{\delta_{\max}}$$

  - $\delta_{\min}$ smallest inter-cluster distance
  - $\delta_{\max}$ largest intra-cluster distance
- Requires pair-wise distances
  - Distance matrix
  - Graphical representation
    - Minimal distance: white
    - Maximal distance: black
- Applicable also to ground truth
  - Notion of difficulty of cluster problem
  - Example

$$I_{\text{Dunn}}(\{c_E, c_P, c_T\}) = \frac{0.577}{1.414} = 0.435$$

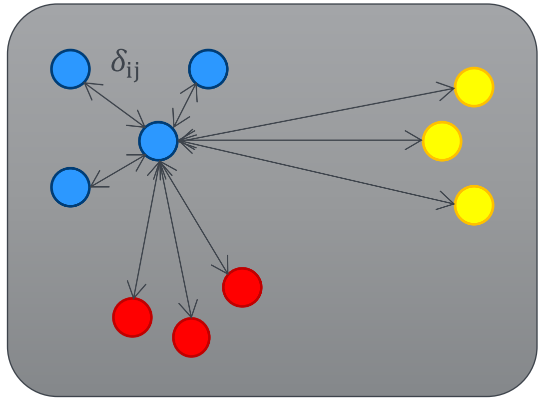




Intrinsic metrics

- Silhouette coefficient $s(i)$ for object x_i
  - Average distance $a(i)$ to all other objects in same cluster ω

$$a(i) = \sum_{x \in \omega, x \neq x_i} \frac{1}{|\omega| - 1} \delta(x_i, x)$$



  - Average distance to some other cluster ω':

$$d(i, \omega') = \sum_{x \in \omega'} \frac{1}{|\omega'|} \delta(x_i, x)$$

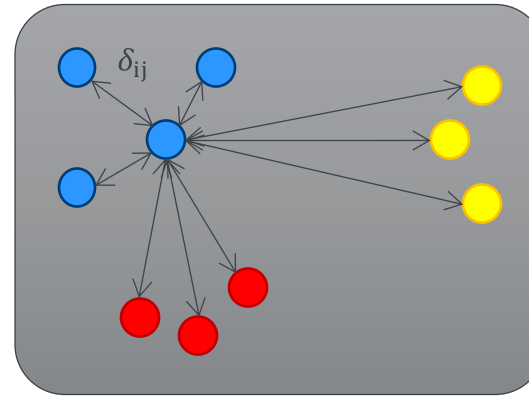
  - Average distance $b(i)$ to closest other cluster

$$b(i) = \min_{\omega' \in \Omega, \omega' \neq \omega} d(i, \omega')$$

  - Silhouette coefficient: $s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))}$

Intrinsic metrics

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))}$$



- Silhouette coefficient
  - Values: $-1 \leq s(i) \leq 1$
  - Value close to 1:
    - $a(i)$ much smaller than $b(i)$
    - Distances within cluster very small in comparison to distances with other clusters
  - Value close to 0:
    - $a(i) \approx b(i)$
    - Same internal as external distance
  - Value close to -1:
    - $b(i)$ much smaller than $a(i)$
    - Other instances are (on average) closer then same cluster
- Aggregation: Average silhouette coefficient $\frac{1}{N}\sum_{i=1}^{N} s(i)$

6 Extrinsic Metrics
for the Evaluation of
Clustering Results

Extrinsic metrics

- **Given**: External knowledge (ground truth of categories)

$$\mathrm{C} = \{c_1, \cdots, c_J\}$$

- **Idea**: Compare clusters Ω and ground truth categories C
- **Approach**:
  - Determine: $n_j^{(i)}$: number of objects from c_i being clustered into ω_j

Extrinsic metrics: Purity

- **Given**: External knowledge (ground truth of categories)
$$\mathrm{C} = \{c_1, \cdots, c_J\}$$
- **Idea**: Compare clusters Ω and ground truth categories C
- **Approach**:
  - Determine: $n_j^{(i)}$: number of objects from c_i being clustered into ω_j
- **Purity:**
  - Ratio of strongest represented category
$$\mathrm{Purity}(\omega_j) = \frac{1}{|\omega_j|} \cdot \max_{i=1,\cdots,J} n_j^{(i)}$$
  - Aggregate over all clusters
$$\mathrm{Purity}(\Omega) = \sum_{j=1}^{K} \frac{|\omega_j|}{N} \cdot \mathrm{Purity}(\omega_j)$$

Example (categories c_1, c_2, c_3 are color coded green/yellow/red)

1. $\omega_1 = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{20}\}$
2. $\omega_2 = \{x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}\}$
3. $\omega_3 = \{x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{27}, x_{28}, x_{30}\}$
4. $\omega_4 = \{x_{26}, x_{29}\}$

Purity:

$$\text{Purity}(\omega_1) = \frac{10}{12} = 0.83$$

$$\text{Purity}(\Omega) = \frac{12}{30} \cdot 0.83 + \frac{8}{30} \cdot 1.0 + \frac{8}{30} \cdot 1.0 + \frac{2}{30} \cdot 1.0 = 0.93$$

Extrinsic metrics: Mutual Information

- **Given**: External knowledge (ground truth of categories)
$$\mathrm{C} = \{c_1, \cdots, c_J\}$$
- **Idea**: Compare clusters Ω and ground truth categories C
- **Approach**:
  - Determine: $n_j^{(i)}$: number of objects from c_i being clustered into ω_j
- **Mutual Information:**
  - Mutual agreement between clustering and categories

$$\mathrm{MI}(\Omega) = \frac{1}{N} \sum_{j=1}^{K} \sum_{i=1}^{J} n_j^{(i)} \cdot \log \frac{n_j^{(i)} \cdot N}{\sum_{m=1}^{J} n_j^{(m)} \cdot \sum_{l=1}^{K} n_l^{(i)}}$$

  - Log base: 2 or $K \cdot J$

Mutual Information between two random variables X, Y

$$MI(X;Y) = D_{KL}(P_{(X,Y)} || P_X \times P_Y) =$$

$$= \sum_{y \in Y} \sum_{x \in X} P_{(X,Y)}(x,y) \log\left(\frac{P_{(X,Y)}(x,y)}{P_X(x) P_Y(y)}\right)$$

Mutual information measures the information that X and Y share. It measures how much knowing one of these variables reduces uncertainty about the other.

Example (categories c_1, c_2, c_3 are color coded green/yellow/red)

1. $\omega_1 = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{20}\}$
2. $\omega_2 = \{x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}\}$
3. $\omega_3 = \{x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{27}, x_{28}, x_{30}\}$
4. $\omega_4 = \{x_{26}, x_{29}\}$

Mutual Information

$$\mathrm{MI}(\Omega) = \frac{1}{N} \sum_{j=1}^{K} \sum_{i=1}^{J} n_j^{(i)} \cdot \log \frac{n_j^{(i)} \cdot N}{\sum_{m=1}^{J} n_j^{(m)} \cdot \sum_{l=1}^{K} n_l^{(i)}}$$

- $n_j^{(i)}$: number of objects from c_i being clustered into ω_j

category \ cluster	ω_1	ω_2	ω_3	ω_4
c_1	10	0	0	0
c_2	2	8	0	0
c_3	0	0	8	2

Example (categories c_1, c_2, c_3 are color coded green/yellow/red)

1. $\omega_1 = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{20}\}$
2. $\omega_2 = \{x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}\}$
3. $\omega_3 = \{x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{27}, x_{28}, x_{30}\}$
4. $\omega_4 = \{x_{26}, x_{29}\}$

Mutual Information

$$\mathrm{MI}(\Omega) = \frac{1}{N} \sum_{j=1}^{K} \sum_{i=1}^{J} n_j^{(i)} \cdot \log \frac{n_j^{(i)} \cdot N}{\sum_{m=1}^{J} n_j^{(m)} \cdot \sum_{l=1}^{K} n_l^{(i)}}$$

- Log base $K \cdot J$
- Several values are 0

$$\mathrm{MI}(\Omega) =$$

$$\frac{1}{30} \cdot \left(10 \cdot \log \frac{10 \cdot 30}{12 \cdot 10} + 2 \cdot \log \frac{2 \cdot 30}{12 \cdot 10} + 8 \cdot \log \frac{8 \cdot 30}{8 \cdot 10} + 8 \cdot \log \frac{8 \cdot 30}{8 \cdot 10} + 2 \cdot \log \frac{2 \cdot 30}{2 \cdot 10}\right) = 0.370$$

Extrinsic metrics: Rand Index

- **Given**: External knowledge (ground truth of categories)

$$\mathrm{C} = \{c_1, \cdots, c_J\}$$

- **Idea**: Compare clusters Ω and ground truth categories C
- **Approach**:
  - Determine: $n_j^{(i)}$: number of objects from c_i being clustered into ω_j
- **Rand Index**:
  - Consider pairs of data items on categories and clusters
    - Agreements: same-same (ss), different-different (dd)
    - Disagreements: same-different (sd), different-same (ds)
  - Agreement-ratio:

$$I_{\mathrm{Rand}}(\Omega) = \frac{\mathrm{ss} + \mathrm{dd}}{\mathrm{ss} + \mathrm{dd} + \mathrm{sd} + \mathrm{ds}}$$

Example (categories c_1, c_2, c_3 are color coded green/yellow/red)

1. $\omega_1 = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{20}\}$
2. $\omega_2 = \{x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}\}$
3. $\omega_3 = \{x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{27}, x_{28}, x_{30}\}$
4. $\omega_4 = \{x_{26}, x_{29}\}$

Rand Index

- Agreements:
  - $ss = 103 = \binom{10}{2} + 1 + \binom{8}{2} + \binom{8}{2} + 1$
  - $dd = 280 = 10 \cdot 18 + 2 \cdot 10 + 8 \cdot 10$
- Disagreements
  - $sd = 32 = 2 \cdot 8 + 2 \cdot 8$
  - $ds = 20 = 2 \cdot 10$

$$I_{\mathrm{Rand}}(\Omega) = \frac{\mathrm{ss} + \mathrm{dd}}{\mathrm{ss} + \mathrm{dd} + \mathrm{sd} + \mathrm{ds}} = \frac{383}{435} = 0.88$$

Do not confuse evaluation metrics and loss functions
(many students do)

Evaluation metrics

- represents what (a majority of) human users find correct
  - task specific
  - independent from prior knowledge
  - must be algorithm independent
  - allows to compare resulting quality of different algorithms
- must be human understandable
  - should be as simple as possible
- may possibly ask a human
  - though this has disadvantages

Loss function

- guides the algorithm to find the right solution
  - algorithm specific
  - regularized to represent prior knowledge
  - improvement of loss function value need not indicate improvement for user
- can be very complex
  - though you have to be careful not to misguide the algorithm
- must be efficiently evaluable

A premature comparison for later reference – not further elaborated now

Evaluation metrics

- Accuracy
- precision
- recall
- f1
- MSE
- BLEU, ROUGE
- …

Loss function

- cross entropy (CE)
- hinge loss
- exponential loss
- regularized MSE

During teaching some metrics are used as evaluation metrics and as loss function (MSE!) – but rarely in practice

7 K-Means

K-Means

- General clustering algorithm

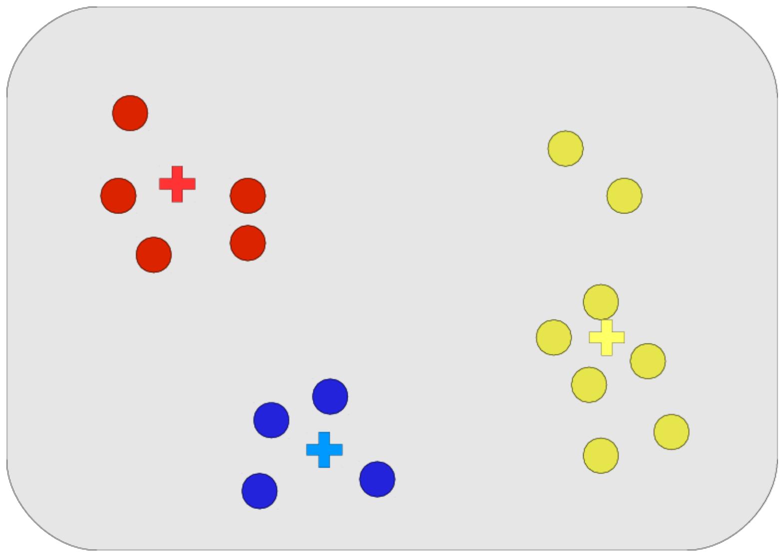
- Characteristics:
  - Flat clusters
  - No overlaps
  - Good runtime
  - Simple to implement

- Parameters
  - K : number of clusters
  - Initial random seed

K-Means Algorithm

- **Given** input data $\{x_i\}_{i=1}^{N}, x_i \in \mathbb{R}^d$, and $\mathrm{d}, K \in \mathbb{N}$
- Choose randomly K cluster centroid seeds $Z = \{z_i | z_i \in \mathbb{R}^d\}_{i=1}^{K}$
- **repeat**
  - For all objects x
    - Assign x to cluster ω_i with minimal $\delta(x, z_i)$
  - For all clusters ω_i
    - Compute centroid $z_i = \frac{1}{|\omega_i|} \sum_{x \in \omega_i} x$

  until centroids do not change



Advantages and disadvantages of k-means

- Advantages:
  - Efficiency:
    time complexity: O(N) for each iteration,
    Number of iterations is usually very small (~ 5 - 10).
  - Simple implementation
  - Easy, good interpretability

  ⇨ K-means the most popular (partitional) cluster algorithm!

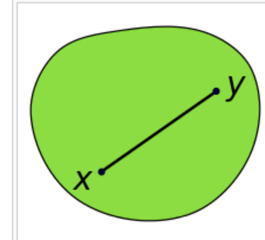


Illustration of a convex set which looks somewhat like a deformed circle. The (black) line segment joining points x and y lies completely within the (green) set. Since this is true for any points x and y within the set that we might choose, the set is convex.

- Disadvantages:
  - Susceptible to noise and outliers since all objects influence the computation of centroids
  - Cluster have always convex form
  - Number k of clusters is often difficult to determine
  - Strong dependency on initial partition (runtime + result!)

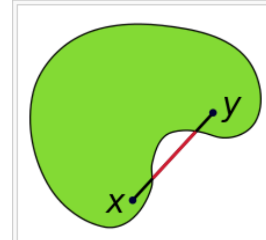


Illustration of a non-convex set. Since the red part of the (black and red) line-segment joining the points x and y lies *outside* of the (green) set, the set is non-convex.

Some Variations

- Random seed
  - Furthests points / modified furthest points

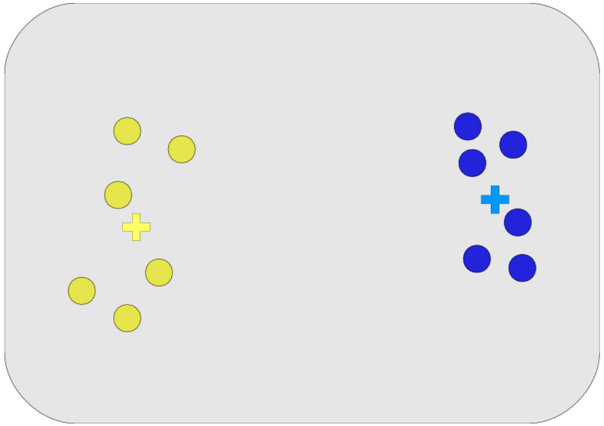
    David Arthur, Sergei Vassilvitskii: *K-means++: The Advantages of Careful Seeding*. In: *Proc. of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*. S. 1027–1035. http://theory.stanford.edu/~sergei/slides/BATS-Means.pdf
- Stop criterion
  - Small changes of the centroids
  - Fixed number of iterations
- K-Medoid
  - Non-standard metrics (e.g. string similarity)
    - Mean centroid cannot be computed
  - K-medoid: Use most central objects

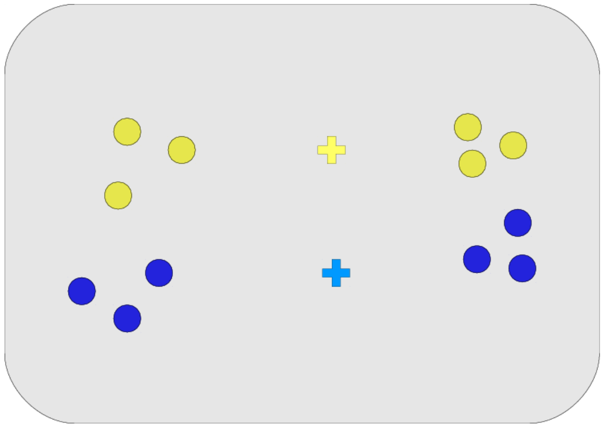
Initial configuration

- Choice of initial seed can cause different outcomes!
- Solution
  - Repeat with different seeds
  - Evaluate quality
    - Dunn index
    - Residual Sum of Squares

$$RSS(\Omega) = \sum_{j=1}^{K} \sum_{x \in \omega_j} \delta\left(x, z_j\right)^2$$

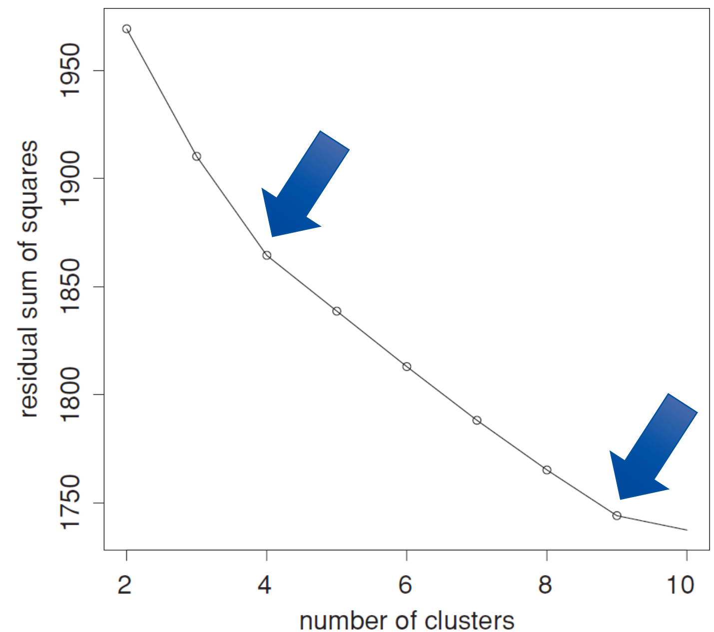
- Choose best performing setting





Choice of K

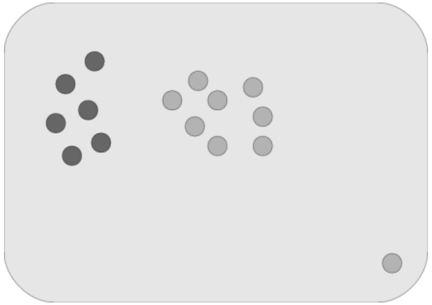
- Important parameter!
- Knowledge about the data
  - Expert insights
- Development of RSS
  - Monotonous decline
  - Typically two points where decline slows down



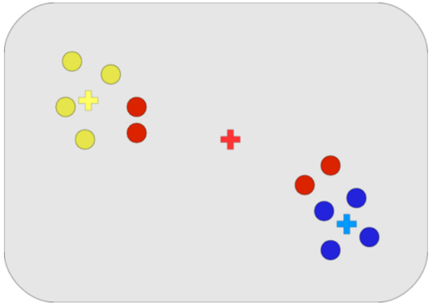
Schubert, Erich. "Stop using the elbow criterion for k-means and how to choose the number of clusters instead." ACM SIGKDD Explorations Newsletter 25.1 (2023): 36-42.

Problematic configurations

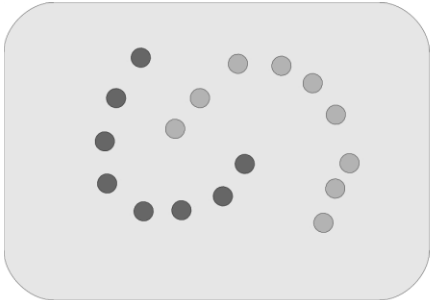
- Outliers
  - Cause singleton clusters
  - Solution:
    - Remove and treat separately

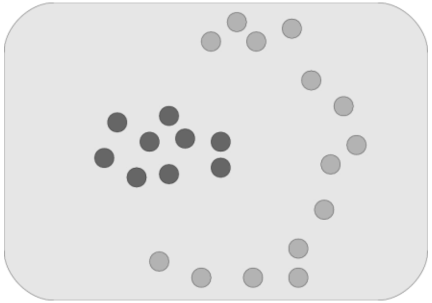


- Empty clusters
  - Unlucky position of centroids
  - Solution:
    - Split large cluster



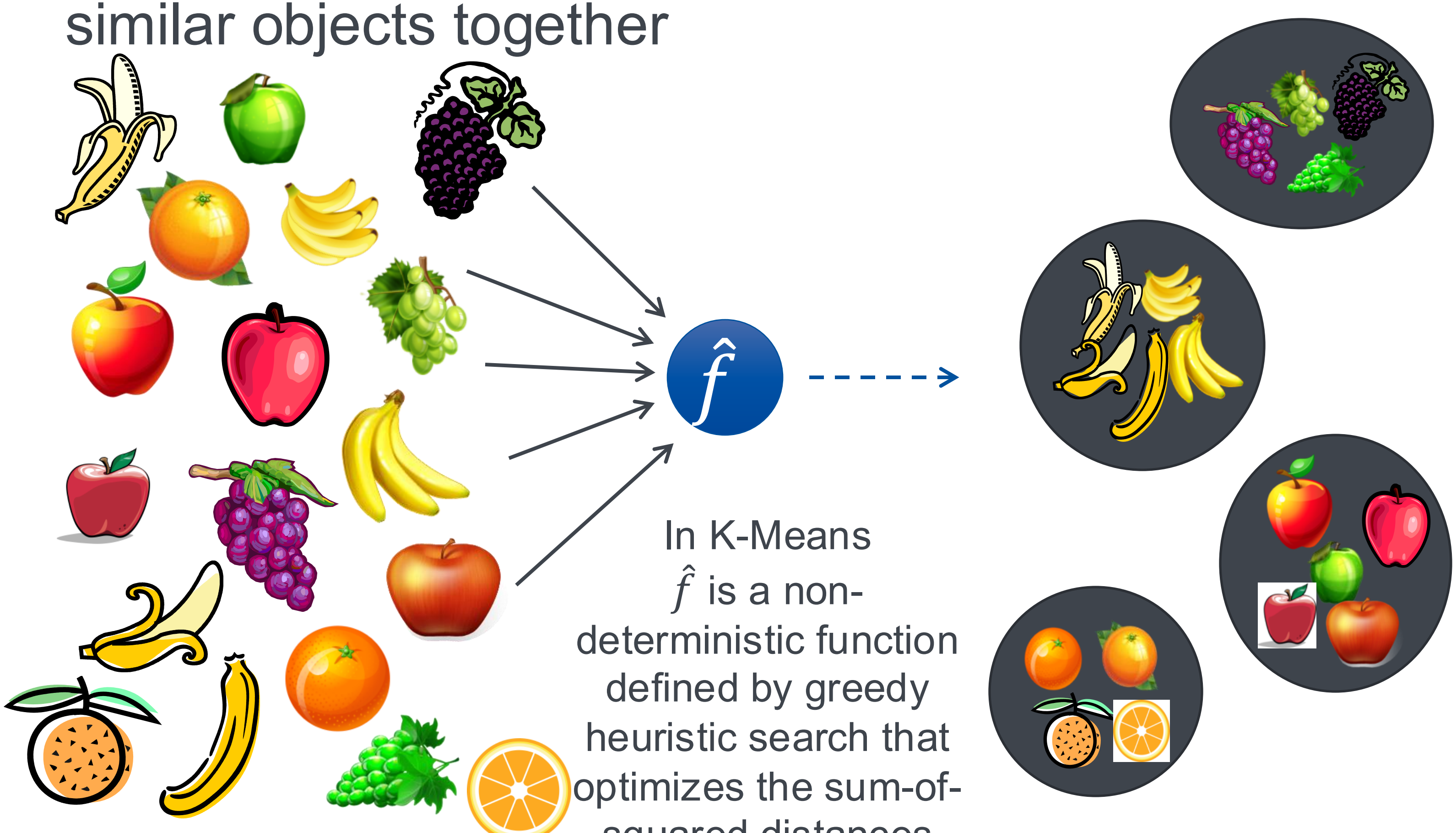
- Non-spheric shapes
  - Cannot be handled!





K-Means-Clustering

- Given a set of objects, find a function $\hat{f}$ to group similar objects together



In K-Means
$\hat{f}$ is a non-deterministic function defined by greedy heuristic search that optimizes the sum-of-squared distances

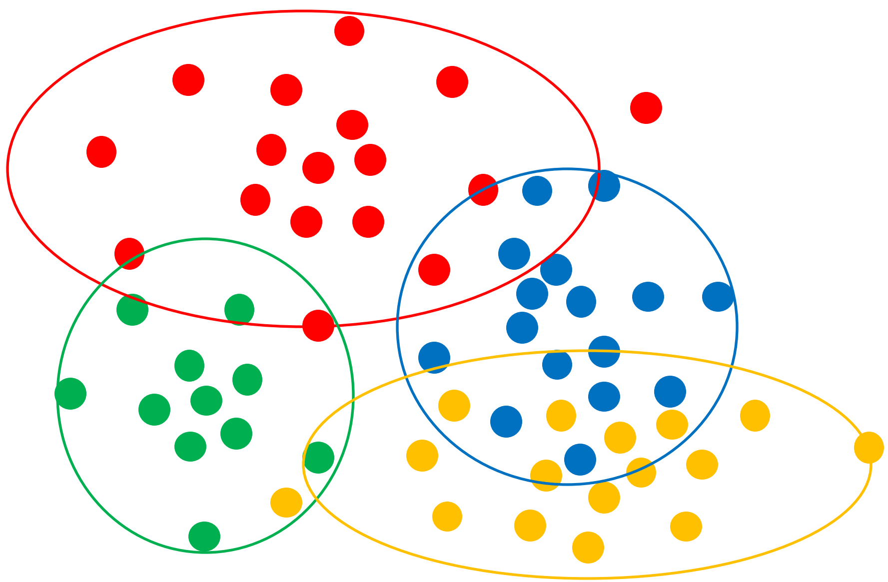
Beware!

- When you run K-Means you will **always** end up with a result.
- That result may be **extremely poor**.
- Whether it is good or bad may be **hard to tell**.
- Often you may want to run a classifier to **determine the important attributes** using the found clusters as target classes.

8 Expectation Maximization

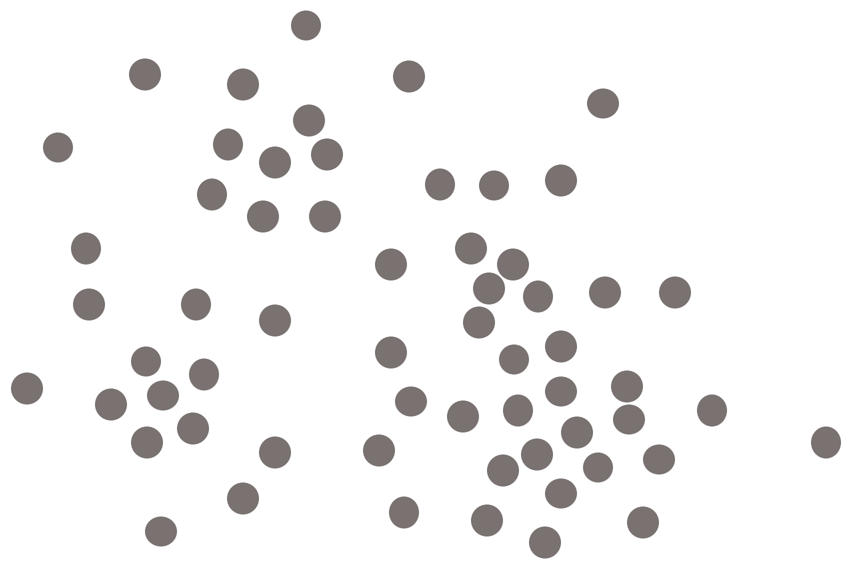
Idea

- Assumption: Data can be explained by a mixture of parametrized probability distributions – one per cluster



Idea

- Data can be explained by a mixture of parametrized probability distributions – one per cluster
- Problem: the true distributions are unknown, all we see is the data



Expectation Maximization (EM)

- General clustering algorithm

- Characteristics:
  - Probabilistic approach
  - Soft assignments to clusters
  - Generalization of K-Means

- Parameters
  - K : number of clusters
  - Initial random seed
  - Model for the distribution

Remember: Continuous Probability Distributions

Discrete case:

$\forall x \in \mathrm{dom}(X)$:

$P(x) \in [0,1]$

$$\sum_{x \in \mathrm{dom}(X)} P(x) = 1$$

Continuous case:

What is $P(x)$ if $\mathrm{dom}(X) = \Omega = \mathbb{R}^d$? → (almost) always 0

Probability density function f with $f(x) \geq 0$:

$$\int_{\Omega} f(x)dx = 1$$

For univariate case $d = 1$:
Cumulative probability distribution F with $\forall x \in \Omega$: $F(x) \in [0,1]$

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(t)dt$$

Multivariate case $d = n$:

$$F(x_1 \dots x_n) = P(X_1 \leq x_1 \dots X_n \leq x_n) = \int_{-\infty}^{x_n} \dots \int_{-\infty}^{x_1} f(t)dt$$

Gaussian models of the individual distribution

- Density of univariate normal distribution (1 dimensional)

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

  Where μ is the mean and σ the standard deviation

- Density of multivariate normal distribution (d-dimensional)

$$f(x) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{-\left(\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)}$$

  Where μ is an d-dimensional vector,
  Σ is the $d \times d$ covariance matrix and $|\Sigma|$ the determinant

Leads to very popular **Gaussian Mixture Models**

- Other non-Gaussian distribution P with parameters θ possible

Model estimation

1. Estimate model parameters from data
2. Maximum likelihood estimation:
   - 1 dimensional case:

$$\mu = \frac{1}{n}\sum_{x \in X} x_i \qquad \sigma^2 = \frac{1}{n-1}\sum_{x \in X}(x_i - \mu)^2$$

   - m-dimensional case:

$$\mu = \frac{1}{n}\sum_{x \in X} \boldsymbol{x_i} \qquad \Sigma = \frac{1}{n-1}\sum_{x \in X}(x_i - \mu)(x_i - \mu)^T$$

   - Maximize log likelihood function:

$$\log(L(\theta|X)) = \log(\prod_{x \in X} P(x|\theta)) = \sum_{x \in X}\log(P(x|\theta))$$

But: we don't know which objects belongs to which cluster ...

Latent variables: Cluster assignment

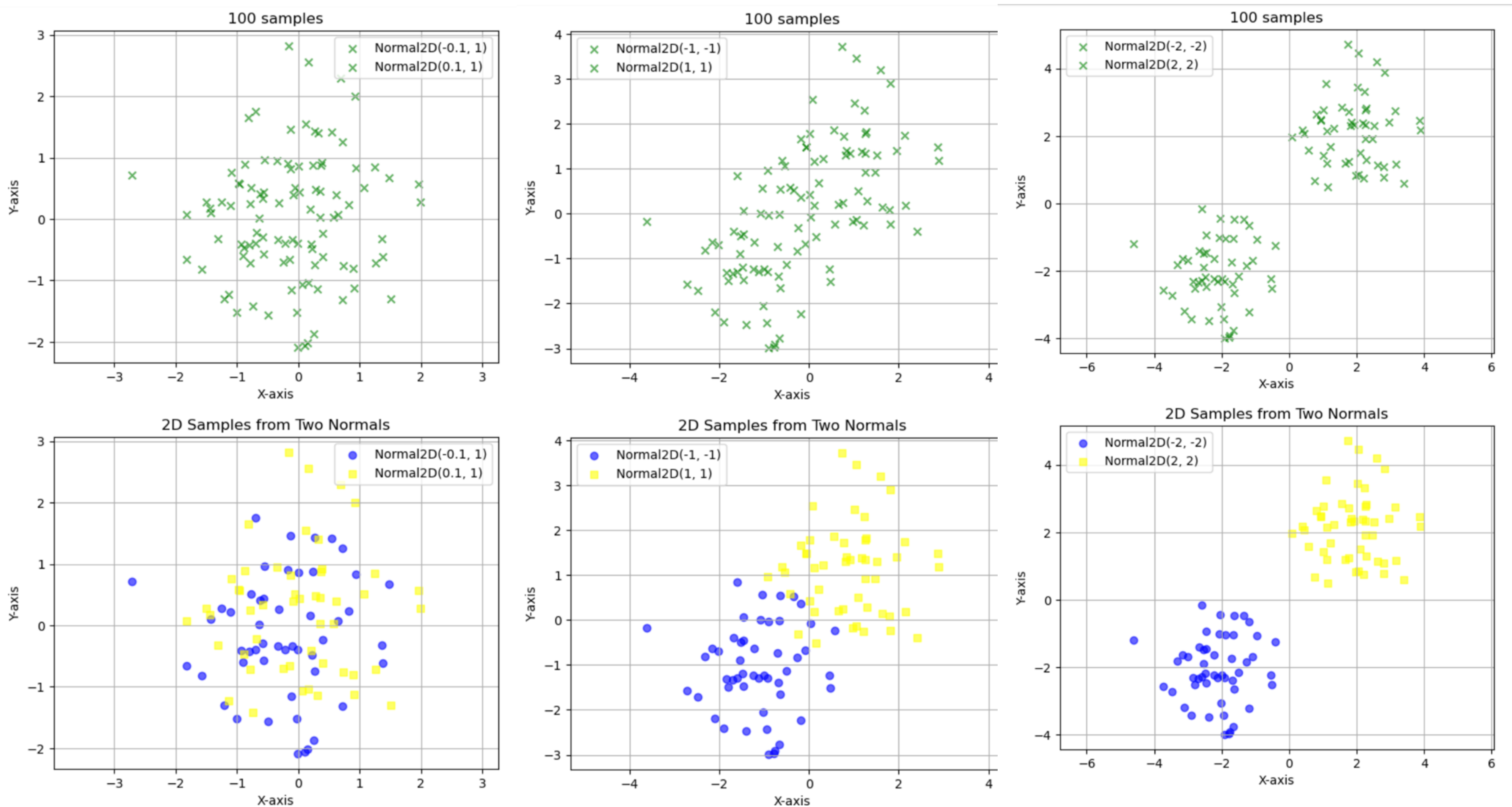
- For each object $x_i \in X$
  we model latent variables z_{ij} indicating to which cluster ω_j it belongs
- **Iterate until convergence:**
  - **Expectation step**
    - Given the current model parameters θ
      calculate the expected values for z_{ij}
      - How probable is it that object x_i belongs to cluster ω_j
        given the current model hypothesis θ
  - **Maximization step**
    - Given the current estimates for the latent variables z_{ij},
      calculate the model parameters θ
      - what are the model parameters θ when we assume that the assigment to clusters was correct

Initialization

- Problem:
  - Expectation step needs model parameters to estimate latent variables
  - Maximization step needs latent variables to estimate model parameters

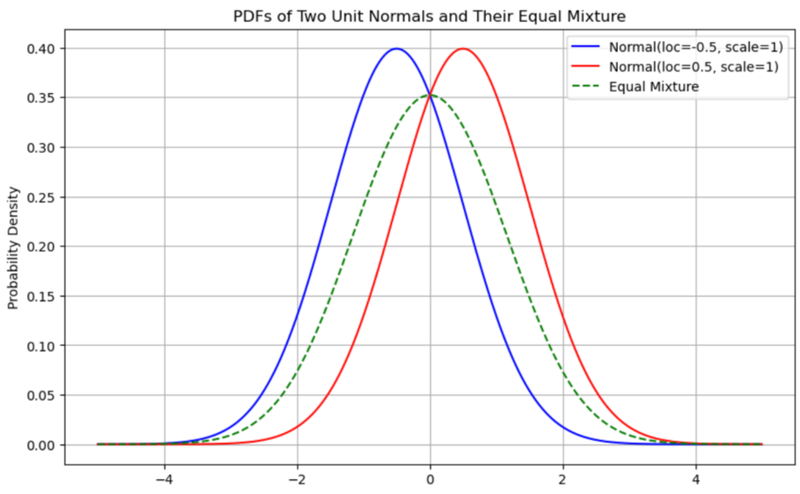
- Different options:
  - Hand selected initial model parameters
  - Random initialization
  - Choose random individuals
  - Perform k-means clustering to find initial clusters

From non-separable to well-separable distributions

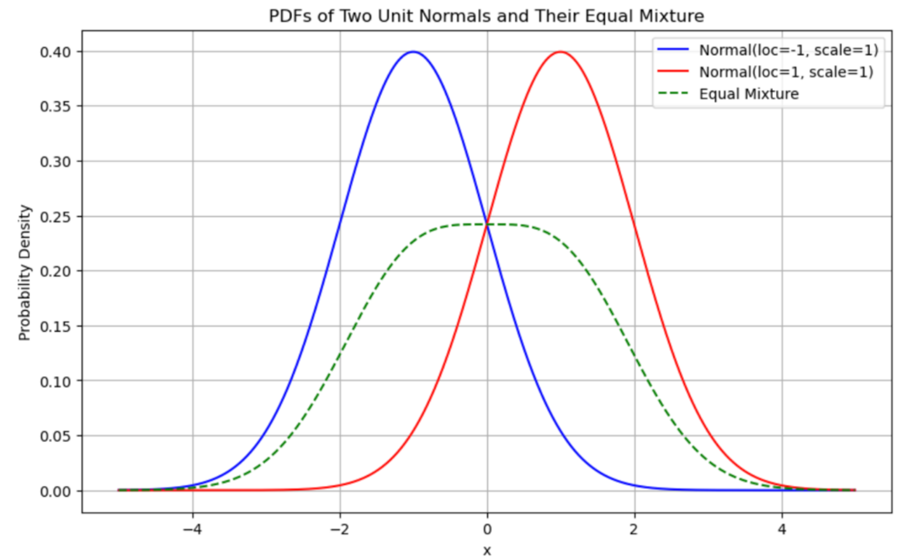


Mixture of two Gaussians

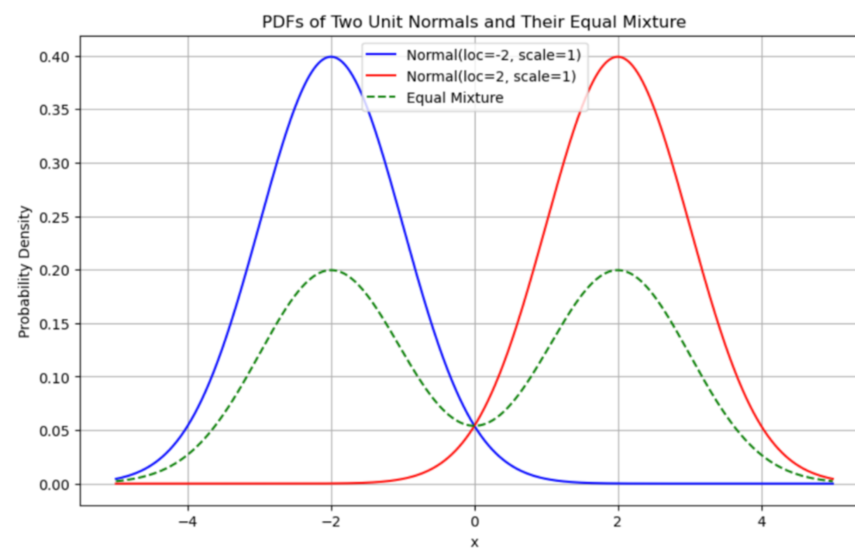
Mixture of close-by Gaussians returns unimodal Gaussian



Transitioning



Mixture of far-aways Gaussians returns bi-modal distribution



Example

Gender	height
F	124
F	115
F	121
F	139
F	98
F	135
F	131
M	170
M	166
M	155
M	167
M	158
M	175
M	143
M	163
M	160
M	145
M	176

- Cluster people by their height
  - Two classes
  - Assume initial model:
    - $\mu_1 = 110$ $\sigma_1 = 20$
    - $\mu_2 = 160$ $\sigma_2 = 20$

- Expectation:
  - $f_{\mu_1,\sigma_1}(124) = 0.0156$
  - $f_{\mu_2,\sigma_2}(124) = 0.0039$
  - Weights:
    - $z_{11} = 0.7982$
    - $z_{12} = 0.2018$
  - ...

Example

Gender	height
F	124
F	115
F	121
F	139
F	98
F	135
F	131
M	170
M	166
M	155
M	167
M	158
M	175
M	143
M	163
M	160
M	145
M	176

- Given all weights:
- New model parameters

$$\mu_j = \frac{\sum_{x \in X} z_{ij} \, x_i}{\sum_{x \in X} z_{ij}}$$

$$\sigma_j^2 = \frac{\sum_{x \in X} z_{ij} (x_i - \mu)^2}{\sum_{x \in X} z_{ij}}$$

- Values:
  - $\mu_1 = 123.72$ $\sigma_1 = 15.98$
  - $\mu_2 = 157.72$ $\sigma_2 = 14.62$

Some other clustering techniques

- Many variations of K-Means
  - K-median, hierarchical K-Means, ….
- Agglomerative clustering
- DBScan (density oriented)
- Generalizations of EM to “graphical models”
  - Latent dirichlet allocation
  - See lecture “Probabilistic Machine Learning” in winter term
- Graph clustering (on graph data)
- Self-supervised learning (as preprocessing for clustering)

Universität Stuttgart
KI

Thank you!

Steffen Staab

E-Mail Steffen.staab@ki.uni-stuttgart.de

Telefon +49 (0) 711 685-88100

www.ki.uni-stuttgart.de/

Universität Stuttgart
Analytic Computing, Institut für Künstliche Intelligenz
Universitätsstraße 32, 50569 Stuttgart