*** Applied Machine Learning Fundamentals *** Clustering

Daniel Wehner, M.Sc.

SAP SE / DHBW Mannheim

Winter term 2023/2024

Find all slides on GitHub (DaWe1992/Applied_ML_Fundamentals)

Lecture Overview

Unit I	Machine Learning Introduction
Unit II	Mathematical Foundations
Unit III	Bayesian Decision Theory
Unit IV	Regression
Unit V	Classification I
Unit VI	Evaluation
Unit VII	Classification II
Unit VIII	**Clustering**
Unit IX	Dimensionality Reduction

Agenda for this Unit

Section:

Introduction

What is Clustering?
Clustering Strategies Overview

Clustering Introduction

- Clustering belongs to the category of **unsupervised learning**
- A clustering algorithm tries to **find structure** in the data
- Once the clusters are found, they first have to be interpreted...
- ...and can then be used for prediction purposes

A cluster should be **internally homogeneous**, but simultaneously **externally heterogeneous**. (Elements of one cluster should be similar to each other, but should differ significantly from elements belonging to other clusters.)

Example Use Cases for Clustering

- **Behavioral segmentation**
  - Customer segmentation (e. g. **sinus milieus**)
  - Creating profiles based on activity monitoring
- **Sorting sensor measurements**
  - Image grouping
  - Detection of activity types in motion sensors
- **Inventory categorization**
  - Grouping inventory by sales activity
  - Grouping inventory by manufacturing metrics
- Many, many more, ...

Clustering Strategies

1. **EM-based clustering**, e. g.: k-Means
2. **Hierarchical clustering**, e. g.:
   - Agglomerative clustering
   - Divisive clustering
3. **Affinity-based clustering**, e. g.:
   - DBSCAN
   - Spectral clustering

Section:

k-Means

k-Means: Procedure

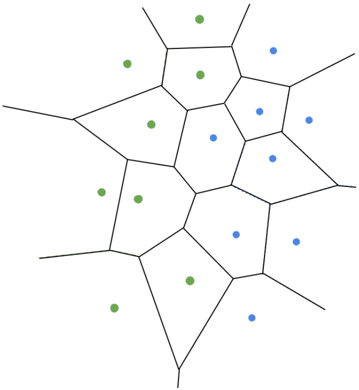
- The algorithm is an instance of vector quantization
  - It represents data points by a single vector (centroid) which is close to them
  - This is useful for **data compression**!
- **How to**: Create k partitions ($\widehat{=}$ clusters) of the data set $\mathcal{D}$, such that the sum of squared deviations from the cluster centroids is **minimal**:

$$\min_{\boldsymbol{\mu}_j} \sum_{j=1}^{k} \sum_{\boldsymbol{x}^{(i)} \in \mathcal{D}_j} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j \right\|^2 \tag{1}$$

- Where $\mathcal{D}_j \equiv j$-th cluster, $\boldsymbol{\mu}_j \equiv$ centroid of j-th cluster

Result: Voronoi Diagram

- The dots represent cluster centroids
- The lines visualize the **cluster boundaries**
- For a new point we can easily determine to which cluster it has to be assigned



Important

k-Means Algorithm

- Input: $\mathcal{D} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(n)}\} \in \mathbb{R}^{n \times m}$, number of clusters k
- Algorithm:
  1. $t \longleftarrow 1$
  2. Randomly choose k means $\boldsymbol{\mu}_1^{\langle 1 \rangle}, \boldsymbol{\mu}_2^{\langle 1 \rangle}, \dots, \boldsymbol{\mu}_k^{\langle 1 \rangle}$
  3. While not converged:

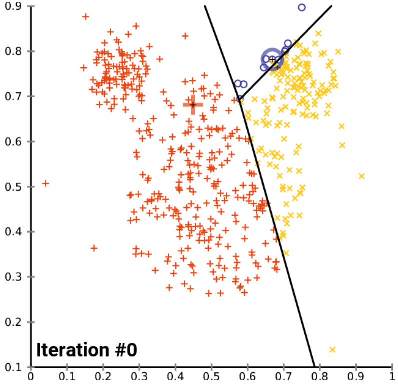
     3a Assign each $\boldsymbol{x}^{(i)} \in \mathcal{D}$ to the closest cluster:

$$\mathcal{D}_j^{\langle t \rangle} = \left\{ \boldsymbol{x}^{(i)} : \|\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j^{\langle t \rangle}\|^2 \leqslant \|\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_{j^*}^{\langle t \rangle}\|^2;\ \forall j^* = 1, 2, \dots, k; \boldsymbol{x}^{(i)} \in \mathcal{D} \right\}$$

     3b Update cluster centroids $\boldsymbol{\mu}_j$:

$$\boldsymbol{\mu}_j^{\langle t+1 \rangle} = \frac{1}{|\mathcal{D}_j^{\langle t \rangle}|} \sum_{\boldsymbol{x}^{(i)} \in \mathcal{D}_j^{\langle t \rangle}} \boldsymbol{x}^{(i)} \qquad \text{then update } t\text{:} \quad t \longleftarrow t + 1$$

k-Means Algorithm (Ctd.)

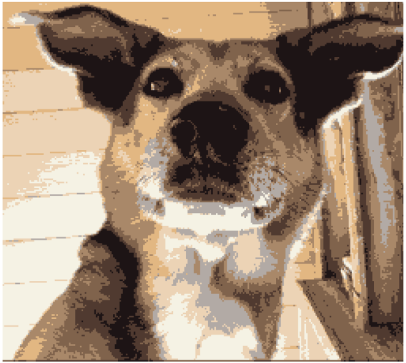


- The algorithm might need some iterations until the result is satisfactory
- **Caveat:** The algorithm might get stuck in local optima
  $\Rightarrow$ several restarts

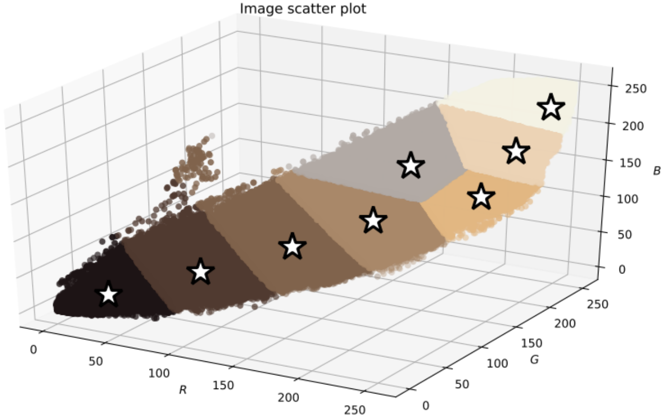
Use Case: Image Compression

Original image

Compressed image



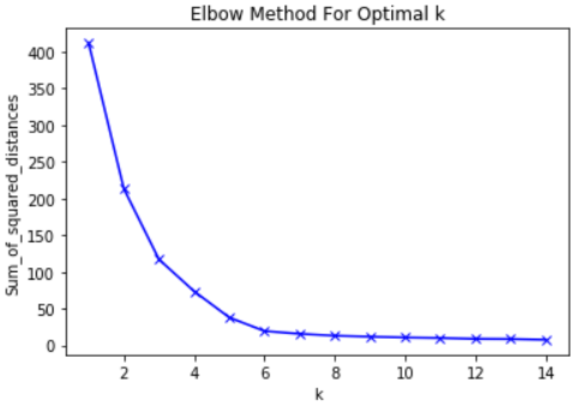
Use Case: Image Compression (Ctd.)



k-Means Issues

- The algorithm assumes that all clusters are **spherical** ($\neq$ **affinity-based clustering**)
- It does not have a notion of **outliers** (unlike DBSCAN)
- What is the correct value for k? $\Rightarrow$ **Elbow-method:**
  - Measure sum of squared distances from data points to cluster centers (inertia)
  - Record results for different values for k and plot them
  - Search for the 'elbow point'

Elbow Method



Section:

Hierarchical Clustering

Agglomerative Clustering Algorithm
Agglomerative Clustering: Example
Distance Metrics between Clusters

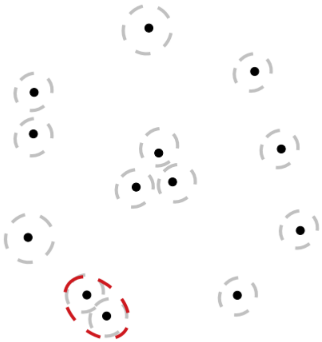
Agglomerative Clustering Algorithm

1. Start with one cluster for each instance: $C = \{\{\boldsymbol{x}^{(i)}\} : \boldsymbol{x}^{(i)} \in \mathcal{D}\}$
2. Compute distance $d(C_i, C_j)$ between all pairs of clusters C_i, C_j
3. Join clusters C_i and C_j with minimum distance into a new cluster C_p:

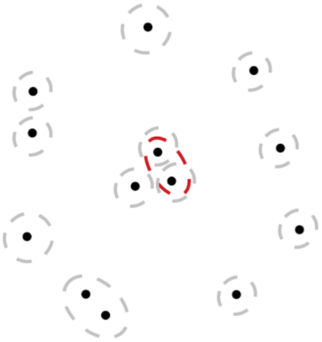
$$C_p = \{C_i, C_j\}$$
$$C = (C \backslash \{C_i, C_j\}) \cup \{C_p\}$$

4. Compute distances between C_p and all other clusters in C
5. If $|C| > 1$, goto 3

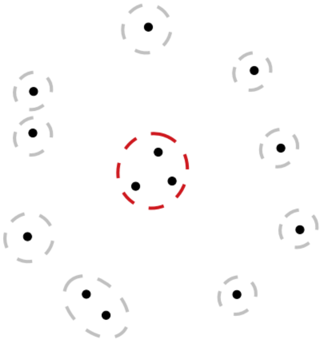
Agglomerative Clustering: Example



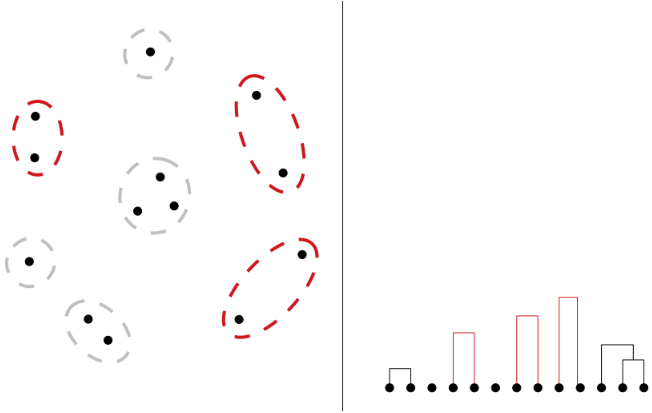
Agglomerative Clustering: Example (Ctd.)



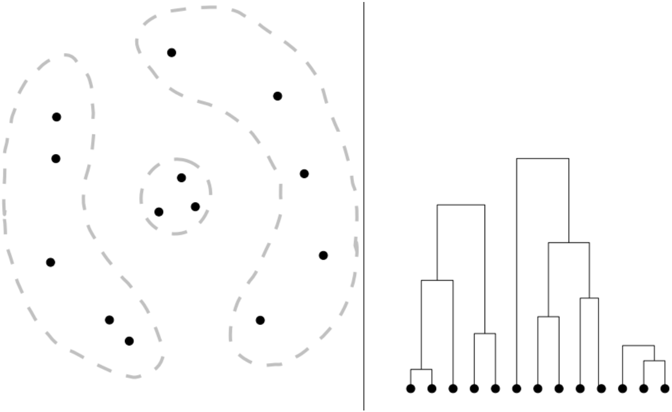
Agglomerative Clustering: Example (Ctd.)



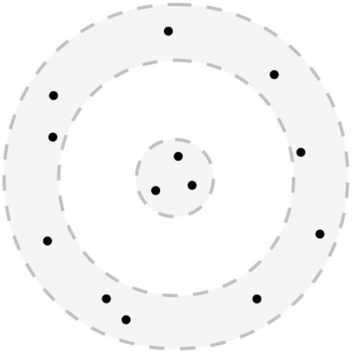
Agglomerative Clustering: Example (Ctd.)

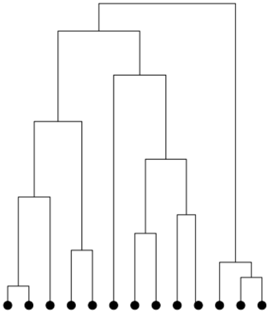


Agglomerative Clustering: Example (Ctd.)



Agglomerative Clustering: Example (Ctd.)



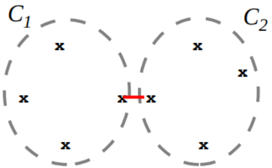


This is a **dendrogram**

Single Linkage

- How to compute the distance between two clusters C_1 and C_2?
- **Single linkage**:

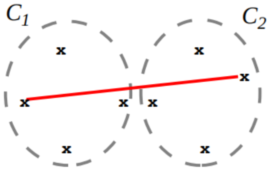
$$d(C_1, C_2) = \min\{d(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) : \boldsymbol{x}^{(i)} \in C_1, \boldsymbol{x}^{(j)} \in C_2\}$$



Complete Linkage

- How to compute the distance between two clusters C_1 and C_2?
- **Complete linkage**:

$$d(C_1, C_2) = \max\{d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) : \mathbf{x}^{(i)} \in C_1, \mathbf{x}^{(j)} \in C_2\}$$



Section:

DBSCAN

DBSCAN

Under construction...

Section:
Wrap-Up

Summary
Self-Test Questions
Lecture Outlook

Summary

- Clustering belongs to the category of **unsupervised learning**
- With clustering we try to find **structure in the data**
- Different algorithms make **different assumptions** about the resulting clusters
- **Clustering Strategies:**
  - EM-based clustering (e. g. k-Means)
  - Hierarchical clustering
  - Affinity-based clustering (e. g. DBSCAN, spectral clustering)

Important

Self-Test Questions

1. What is clustering?
2. What is the definition of a cluster. Which properties should it have?
3. Describe the general procedure of k-Means. What are disadvantages?
4. What is a dendrogram?
5. Describe what DBSCAN works!
6. What is affinity-based clustering? How does it differ from k-Means?

What's next...?

Unit I	Machine Learning Introduction
Unit II	Mathematical Foundations
Unit III	Bayesian Decision Theory
Unit IV	Regression
Unit V	Classification I
Unit VI	Evaluation
Unit VII	Classification II
Unit VIII	Clustering
Unit IX	**Dimensionality Reduction**

Thank you very much for the attention!

Topic: *** Applied Machine Learning Fundamentals *** Clustering
Term: Winter term 2023/2024

Contact:

Daniel Wehner, M.Sc.
SAP SE / DHBW Mannheim
daniel.wehner@sap.com

Do you have any questions?