*** 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

1. Introduction
2. k-Means
3. Hierarchical Clustering
4. DBSCAN
5. Wrap-Up

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

Clustering Strategies

There are different types of clustering algorithms.

Most prominent are:

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

What is k-Means?
k-Means Algorithm
Use Case: Image Compression
Problems and Issues

Important

k-Means: Procedure

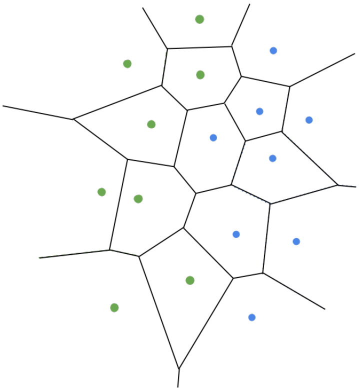
- The algorithm is an instance of **vector quantization**
  - It represents data points by a single vector (**centroid**) which is close to them
  - This is useful for **data compression**!
- **How to**: Create k partitions $\boldsymbol{X}_j$ $(1 \leqslant j \leqslant k)$ of the dataset $\boldsymbol{X}$, such that the sum of squared deviations from the cluster centroids is **minimal**:

$$\min_{\boldsymbol{\mu}_j, \boldsymbol{X}_j} \sum_{j=1}^{k} \sum_{\boldsymbol{x} \in \boldsymbol{X}_j} \|\boldsymbol{x} - \boldsymbol{\mu}_j\|^2 \tag{1}$$

- $\boldsymbol{X}_j$ is the j-th cluster and $\boldsymbol{\mu}_j$ its centroid

Result: Voronoi Diagram

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



Algorithm 1: *k*-Means Algorithm

Input: $\boldsymbol{X} = \left\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)}\right\}$, number of clusters k

$t \longleftarrow 1$

Randomly choose k means $\boldsymbol{\mu}_1^{\langle t\rangle}, \boldsymbol{\mu}_2^{\langle t\rangle}, \ldots, \boldsymbol{\mu}_k^{\langle t\rangle}$

while *not converged* **do**

Assign each $\boldsymbol{x} \in \boldsymbol{X}$ to the closest cluster:

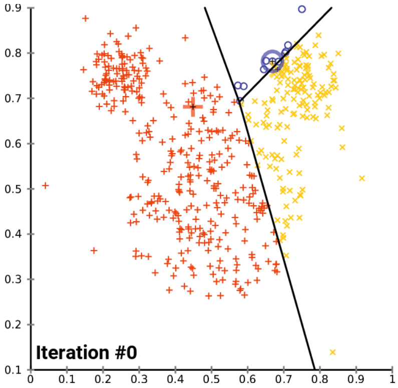
$$\boldsymbol{X}_j^{\langle t\rangle} \longleftarrow \left\{\boldsymbol{x} \in \boldsymbol{X} : \|\boldsymbol{x} - \boldsymbol{\mu}_j^{\langle t\rangle}\|^2 \leqslant \|\boldsymbol{x} - \boldsymbol{\mu}_{j^*}^{\langle t\rangle}\|^2 ; j^* = 1, 2, \ldots, k\right\}$$

Update all cluster centroids $\boldsymbol{\mu}_j^{\langle t\rangle}$:

$$\boldsymbol{\mu}_j^{\langle t+1\rangle} \longleftarrow \frac{1}{|\boldsymbol{X}_j^{\langle t\rangle}|} \sum_{\boldsymbol{x} \in \boldsymbol{X}_j^{\langle t\rangle}} \boldsymbol{x}$$

$t \longleftarrow t + 1$

k-Means Algorithm (Ctd.)

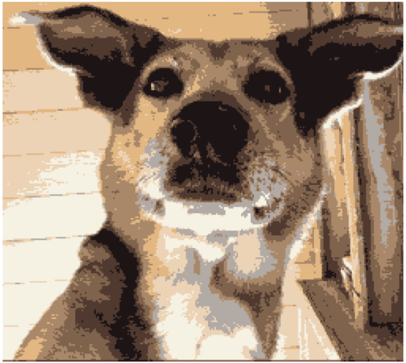


- The algorithm might need some iterations until the result is satisfactory
- **Caveat:** The algorithm might get stuck in local optima
  $\Rightarrow$ several restarts might be required

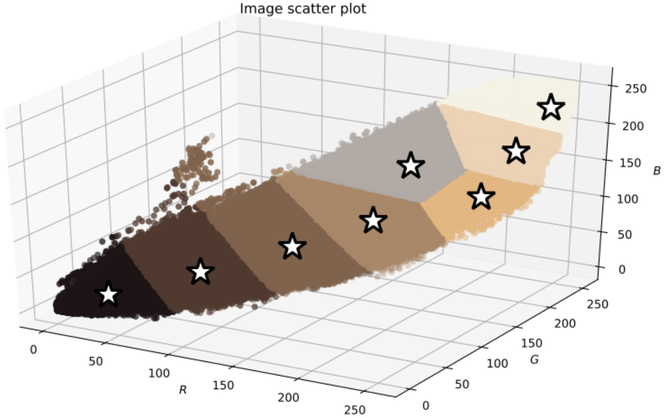
Use Case: Image Compression

Original image

Compressed image



Use Case: Image Compression (Ctd.)

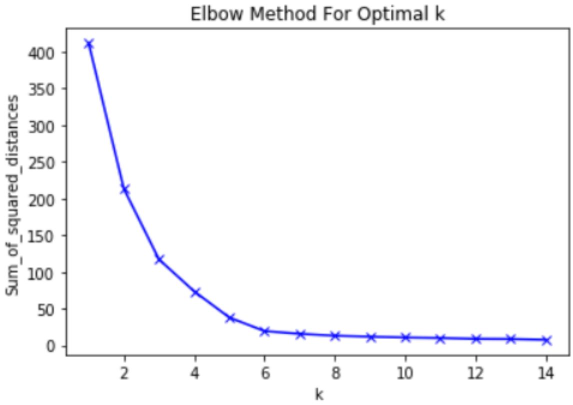


k-Means Issues

The algorithm has some downsides. Some of them are:

- The algorithm assumes that all clusters are **spherical** (in contrast to affinity-based clustering)
- It does not have a notion of **outliers** (unlike DBSCAN)
- What is the correct value for k? $\Rightarrow$ Elbow-method:
  - Compute the average of the sum of squared distances from all data points to their cluster centers for different values of k
  - Create a line plot based on the result
  - Search for the elbow point

Elbow Method



Section:

Hierarchical Clustering

Agglomerative Clustering Algorithm
Agglomerative Clustering: Example
Distance Metrics between Clusters

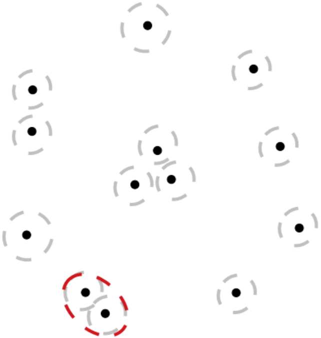
Agglomerative Clustering Algorithm

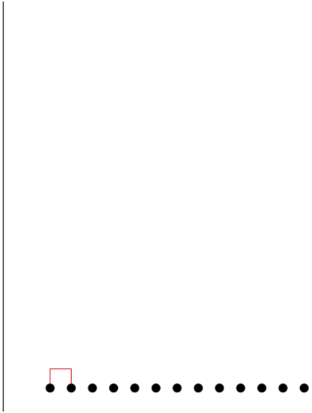
1. Start with one cluster for each instance: $C = \{\{x\} : x \in \boldsymbol{X}\}$
2. Compute the distance $d(C_i, C_j)$ between all pairs of clusters C_i, C_j
3. Join the clusters C_i and C_j with minimum distance into a new cluster C_p:

$$
\begin{aligned}
C_p &= \{C_i, C_j\} \\
C &= \left(C \setminus \{C_i, C_j\}\right) \cup \{C_p\}
\end{aligned}
$$

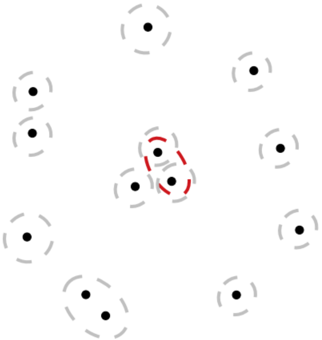
4. Compute the distances between C_p and all other clusters in C
5. If $|C| > 1$, goto 3

Agglomerative Clustering: Example

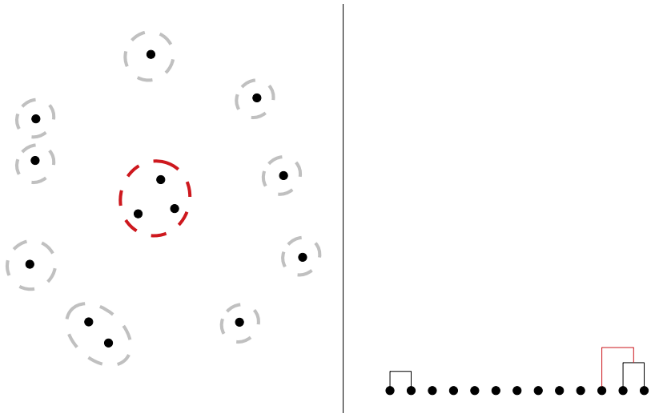




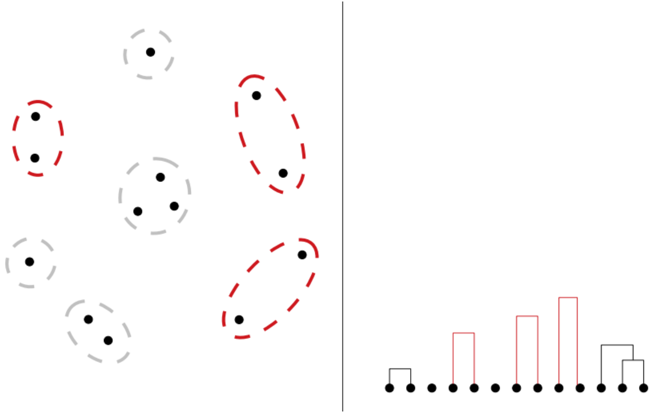
Agglomerative Clustering: Example (Ctd.)



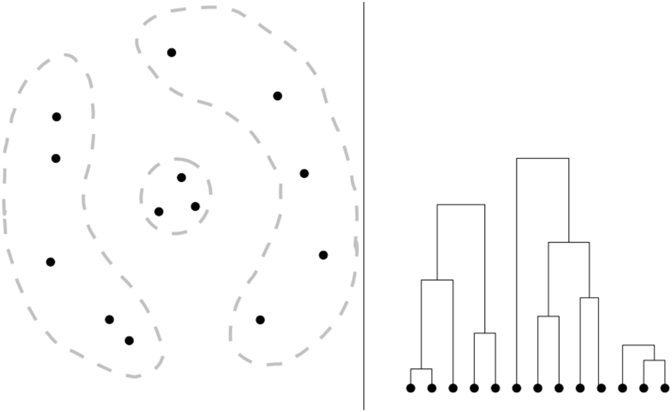
Agglomerative Clustering: Example (Ctd.)



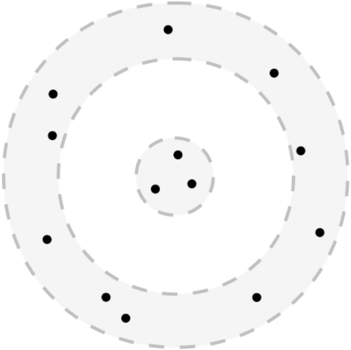
Agglomerative Clustering: Example (Ctd.)

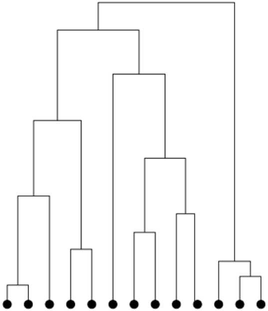


Agglomerative Clustering: Example (Ctd.)



Agglomerative Clustering: Example (Ctd.)



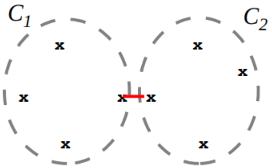


This is a **dendrogram**

Single Linkage

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

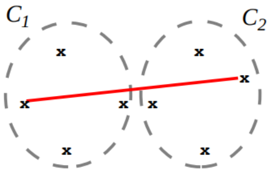
$$d(C_1, C_2) = \min\left\{d\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) : \mathbf{x}^{(i)} \in C_1, \mathbf{x}^{(j)} \in C_2\right\}$$



Complete Linkage

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

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



Section:

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)

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?