

*** Applied Machine Learning Fundamentals ***

Clustering

Daniel Wehner, M.Sc.

SAP SE / DHBW Mannheim

Winter term 2023/2024



Find all slides on [GitHub](https://github.com/DaWe1992/Applied_ML_Fundamentals) (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

What is Clustering?
Clustering Strategies Overview

② *k*-Means

Introduction
k-Means Algorithm
Use Case: Image Compression
Problems and Issues

③ Hierarchical Clustering

Agglomerative Clustering Algorithm
Agglomerative Clustering: Example
Distance Metrics between Clusters

④ DBSCAN

⑤ Wrap-Up

Summary
Self-Test Questions
Lecture Outlook

Section:
Introduction



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 ($\hat{=}$ clusters) of the data set \mathcal{D} , such that the sum of squared deviations from the cluster centroids is **minimal**:

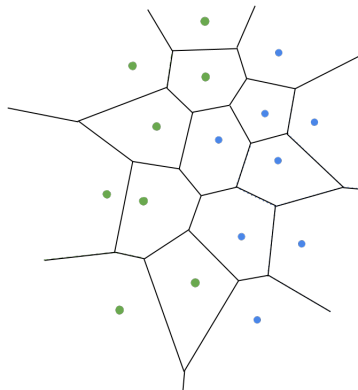
$$\min_{\mu_j} \sum_{j=1}^k \sum_{\mathbf{x}^{(i)} \in \mathcal{D}_j} \|\mathbf{x}^{(i)} - \mu_j\|^2 \quad (1)$$

- Where $\mathcal{D}_j \equiv j$ -th cluster, $\mu_j \equiv$ 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



k-Means Algorithm

- Input: $\mathcal{D} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}\} \in \mathbb{R}^{n \times m}$, number of clusters k
- Algorithm:

① $t \leftarrow 1$

② Randomly choose k means $\mu_1^{(1)}, \mu_2^{(1)}, \dots, \mu_k^{(1)}$

③ While not converged:

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

$$\mathcal{D}_j^{(t)} = \left\{ \mathbf{x}^{(i)} : \|\mathbf{x}^{(i)} - \mu_j^{(t)}\|^2 \leq \|\mathbf{x}^{(i)} - \mu_{j^*}^{(t)}\|^2; \forall j^* = 1, 2, \dots, k; \mathbf{x}^{(i)} \in \mathcal{D} \right\}$$

3b Update cluster centroids μ_j :

$$\mu_j^{(t+1)} = \frac{1}{|\mathcal{D}_j^{(t)}|} \sum_{\mathbf{x}^{(i)} \in \mathcal{D}_j^{(t)}} \mathbf{x}^{(i)} \quad \text{then update } t: \quad t \leftarrow t + 1$$



k-Means Algorithm (Ctd.)

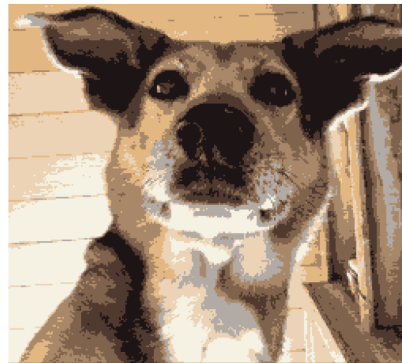
- The algorithm might need some iterations until the result is satisfactory
- **Caveat:** The algorithm might get stuck in local optima
⇒ several restarts

Use Case: Image Compression

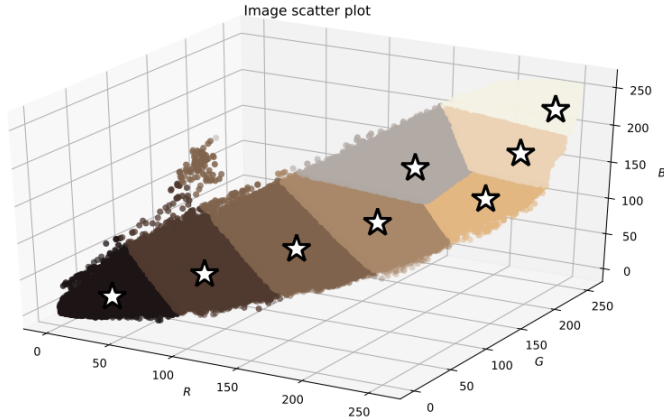
Original image



Compressed image



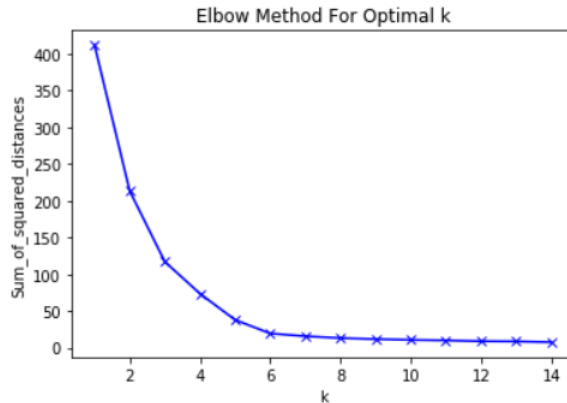
Use Case: Image Compression (Ctd.)



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

- 1 Start with one cluster for each instance: $C = \{\{\mathbf{x}^{(i)}\} : \mathbf{x}^{(i)} \in \mathcal{D}\}$
- 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 :

$$C_p = \{C_i, C_j\}$$

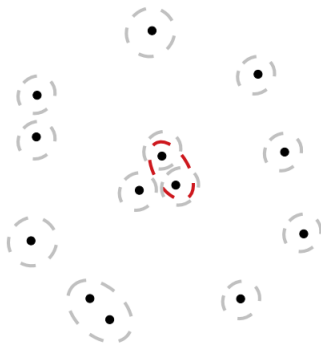
$$C = (C \setminus \{C_i, C_j\}) \cup \{C_p\}$$

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

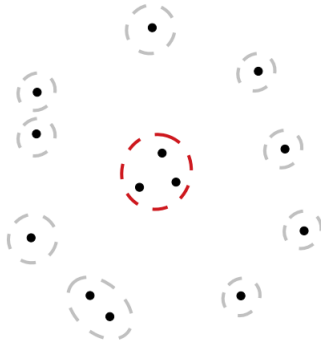
Agglomerative Clustering: Example



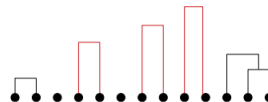
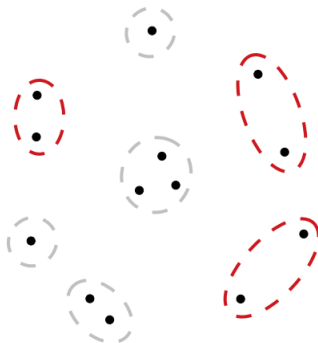
Agglomerative Clustering: Example (Ctd.)



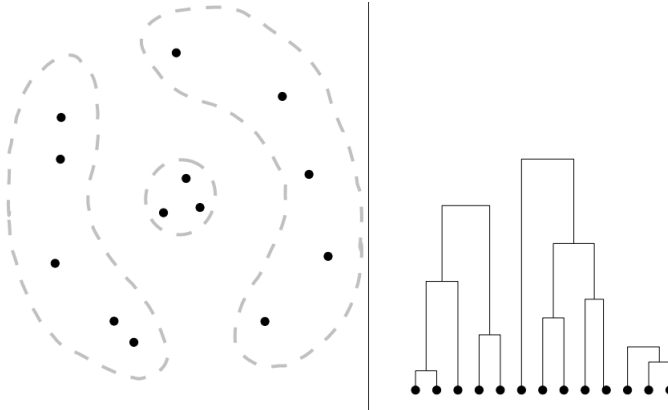
Agglomerative Clustering: Example (Ctd.)



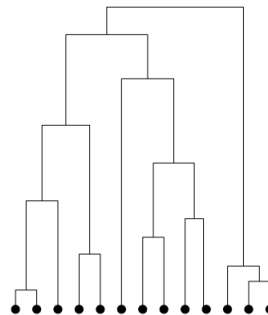
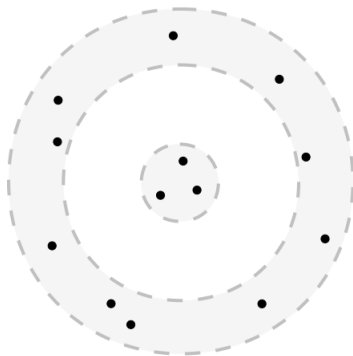
Agglomerative Clustering: Example (Ctd.)



Agglomerative Clustering: Example (Ctd.)



Agglomerative Clustering: Example (Ctd.)

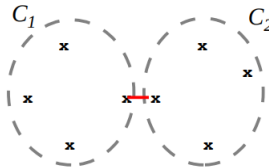


This is a
dendrogram

Single Linkage

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

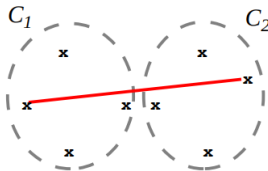
$$d(C_1, C_2) = \min\{d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) : \mathbf{x}^{(i)} \in C_1, \mathbf{x}^{(j)} \in C_2\}$$



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

- 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?