* * * Artificial Intelligence and Machine Learning * * *

Clustering Algorithms

Daniel Wehner, M.Sc.

SAP SE / DHBW Mannheim

Summer term 2025

Find all slides on GitHub (DaWe1992/Applied_ML_Fundamentals)

Lecture Overview

Agenda for this Unit

1. KMeans
2. Hierarchical Clustering
3. DBSCAN
4. Mean-Shift Clustering
5. Wrap-Up

Section:

KMeans

Introduction to Clustering and Clustering Strategies
Introduction to KMeans
KMeans Algorithm
Use Case: Image Compression
Problems of KMeans and Solutions

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**, i. e. 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.: KMeans
2. **Hierarchical clustering**, e. g.: agglomerative clustering, divisive clustering
3. **Affinity-based/Density-based clustering**,
   e. g.: DBSCAN, mean-shift clustering, spectral clustering

KMeans: 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 (i. e. clusters) $\boldsymbol{X}_k$ $(1 \leqslant k \leqslant K)$ of the dataset $\boldsymbol{X}$, such that the sum of squared deviations from the cluster centroids is **minimal**:

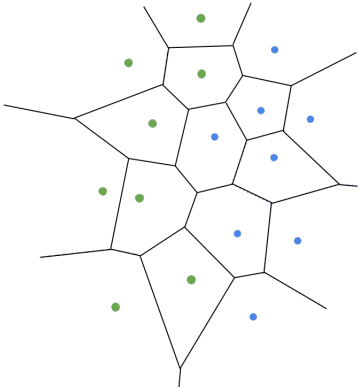
$$\min_{\boldsymbol{\mu}^k, \boldsymbol{X}_k} \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in \boldsymbol{X}_k} \left\| \boldsymbol{x} - \boldsymbol{\mu}^k \right\|^2 \tag{1}$$

- $\boldsymbol{X}_k$ is the k-th cluster / partition and $\boldsymbol{\mu}^k$ its centroid

Result: Voronoi Diagram

Voronoi diagram:

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



KMeans Algorithm

Input: $\boldsymbol{X} = \left\{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\right\}$, number of clusters K

1 Set $t \longleftarrow 0$ and randomly choose K means (cluster centroids) $\boldsymbol{\mu}^{1,t}, \boldsymbol{\mu}^{2,t}, \ldots, \boldsymbol{\mu}^{K,t}$

2 **while** *not converged* **do**

3 **E-step:** Assign each $\boldsymbol{x} \in \boldsymbol{X}$ to the closest cluster:

$$\boldsymbol{X}_{k,t} \longleftarrow \left\{\boldsymbol{x} \in \boldsymbol{X} : \left\|\boldsymbol{x} - \boldsymbol{\mu}^{k,t}\right\|^2 \leqslant \left\|\boldsymbol{x} - \boldsymbol{\mu}^{j,t}\right\|^2 ; \forall j \in \left\{1, 2, \ldots, K\right\} \backslash \left\{k\right\}\right\}$$

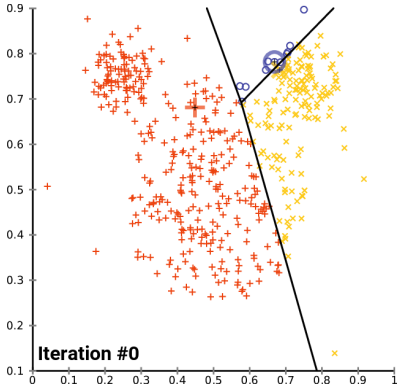
4 **M-step:** Update all cluster centroids $\boldsymbol{\mu}^{k,t}$:

$$\boldsymbol{\mu}^{k,t+1} \longleftarrow \frac{1}{|\boldsymbol{X}_{k,t}|} \sum_{\boldsymbol{x} \in \boldsymbol{X}_{k,t}} \boldsymbol{x}$$

5 $t \longleftarrow t + 1$

6 **end**

KMeans Algorithm (Ctd.)

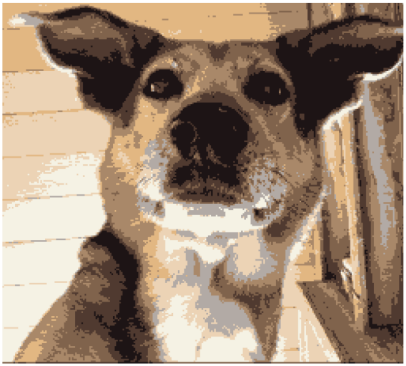


- The algorithm **might need some iterations** until the result is satisfactory
- **Caveat:** The algorithm **might get stuck in local optima**, i. e. several restarts might be required

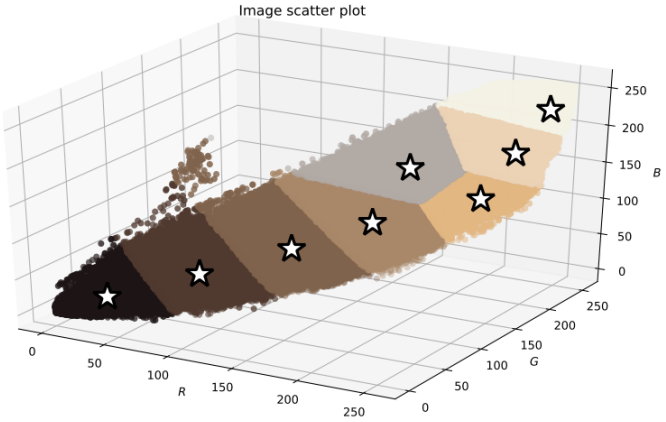
Use Case: Image Compression

Original image

Compressed image



Use Case: Image Compression (Ctd.)



KMeans Issues

The algorithm has some downsides:

1. Several restarts might be required as the algorithm **can get stuck in local optima** *(computation can turn out to be expensive)*
2. The algorithm **assumes all clusters to be spherical** *(remedies?)*
3. It does **not have a notion of outliers** *(unlike DBSCAN)*
4. **Choice of K:** What is the correct value for K?

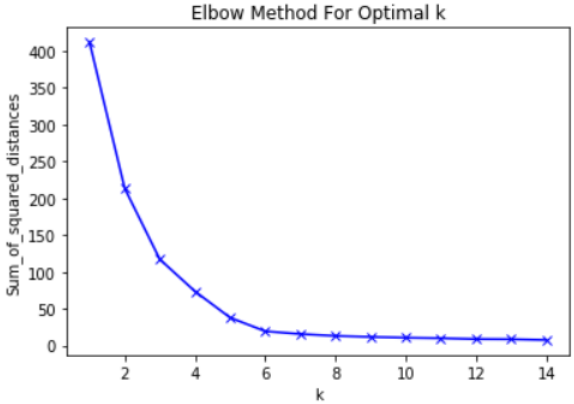
But: The algorithm is simple and intuitive

Choice of K

What is the correct value for K?

1. We can use the **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**
2. Remember the **BAYESian Information Criterion (BIC)**?
   - Try out several values for K and record the BIC values
   - Choose K which minimizes the BIC

Elbow Method



Section:

Hierarchical Clustering

Agglomerative Clustering Algorithm

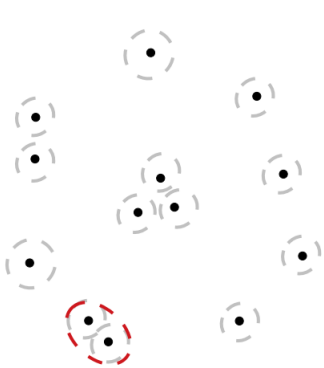
Input: $\boldsymbol{X} = \left\{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\right\}$, distance metric d, number of clusters K

1 Start with one cluster for each instance: $\mathfrak{C} := \left\{\{\boldsymbol{x}\} : \boldsymbol{x} \in \boldsymbol{X}\right\}$
2 **while** $|\mathfrak{C}| > K$ **do**
3 Compute the distance $d(C_i, C_j)$ between all pairs of clusters $C_i, C_j \in \mathfrak{C}$
4 Merge the clusters C_i and C_j with minimum distance into a new cluster $\widetilde{C}$:

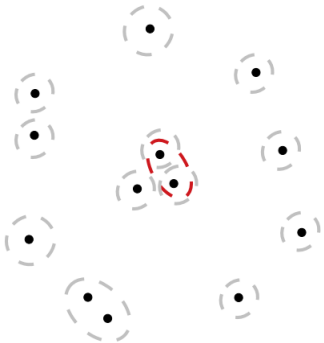
$$\widetilde{C} := \left\{C_i, C_j\right\}$$
$$\mathfrak{C} \longleftarrow \left(\mathfrak{C} \setminus \left\{\{C_i\}, \{C_j\}\right\}\right) \cup \left\{\widetilde{C}\right\}$$

5 **end**
6 **return** *the set of clusters* $\mathfrak{C}$

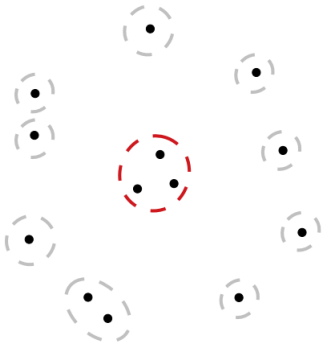
Agglomerative Clustering: Example



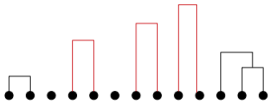
Agglomerative Clustering: Example (Ctd.)



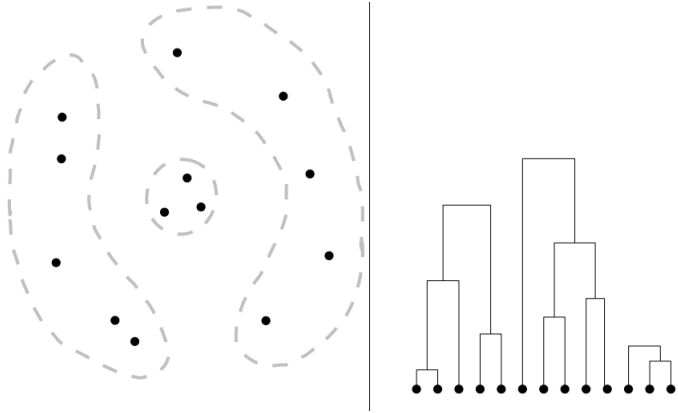
Agglomerative Clustering: Example (Ctd.)



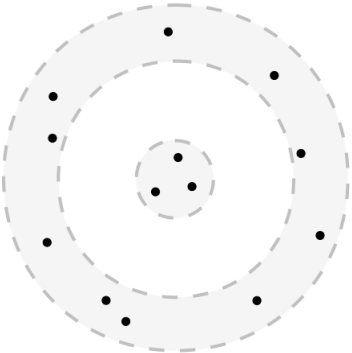
Agglomerative Clustering: Example (Ctd.)

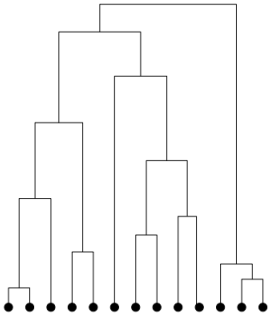


Agglomerative Clustering: Example (Ctd.)



Agglomerative Clustering: Example (Ctd.)



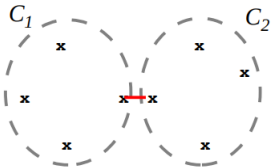


This is a
dendrogram

Single Linkage

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

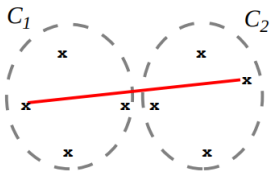
$$d(C_1, C_2) := \min\left\{ d(\boldsymbol{x}, \boldsymbol{y}) : \boldsymbol{x} \in C_1, \boldsymbol{y} \in C_2 \right\}$$



Complete Linkage

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

$$d(C_1, C_2) := \max\Big\{ d\big(\boldsymbol{x}, \boldsymbol{y}\big) : \boldsymbol{x} \in C_1, \boldsymbol{y} \in C_2 \Big\}$$



Section:

DBSCAN

Introduction to DBSCAN
Comparison of KMeans and DBSCAN
An illustrative Example

General Idea of DBSCAN

- **DBSCAN** is short for **D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise
- The algorithm is a density-based method
- **General idea:**
  - Group together points that are closely packed in high-density regions
  - Mark points as outliers which lie alone in low-density regions

Advantages:

- The algorithm does **not assume the clusters to be of a certain shape**
- **Concept of outliers:** Not all data points have to be assigned to a cluster

DBSCAN Hyperparameters

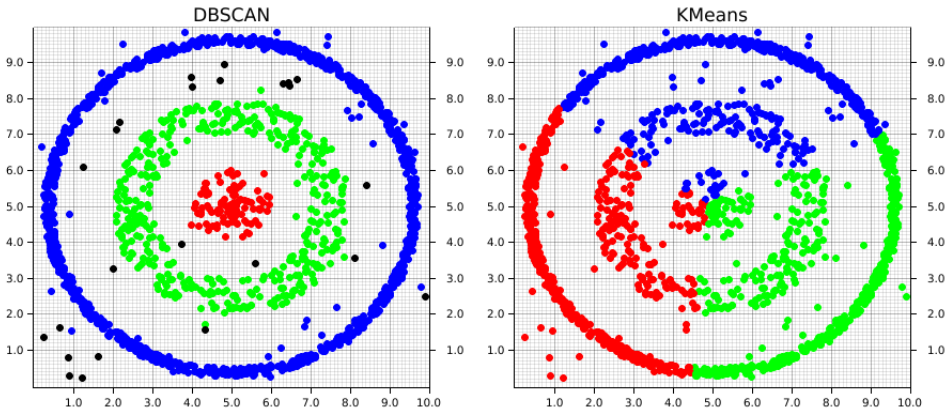
- DBSCAN has two hyperparameters, `eps` and `minPts`
- DBSCAN distinguishes three different types of points:

Core point: A point ***p*** is a **core point**, if at least `minPts` points are within distance `eps` of ***p*** *(including **p** itself)*

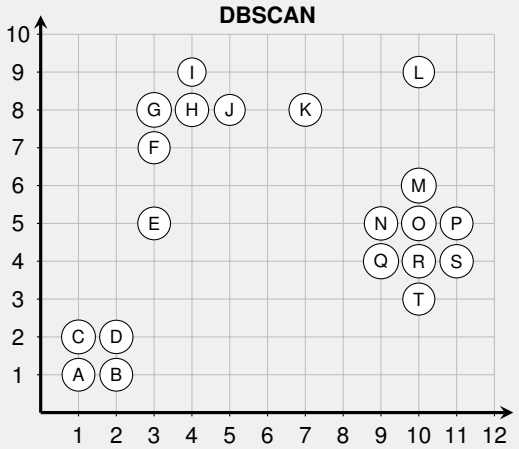
Border point: A non-core point ***q*** is a **border point**, if there is a core point ***p*** within distance `eps` from ***q***

Outlier: Points which are neither core points nor border points are said to be **outliers**

KMeans vs. DBSCAN



Example: DBSCAN



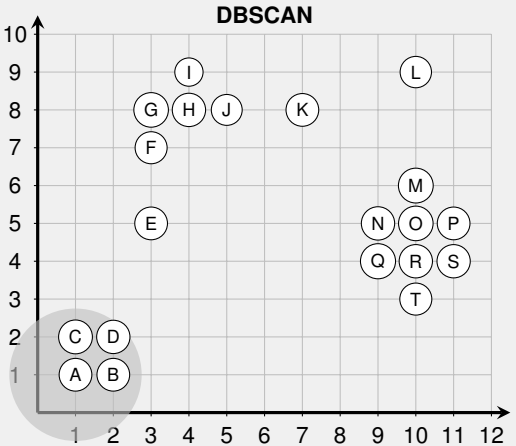
Hyperparameters:

$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

Example: DBSCAN



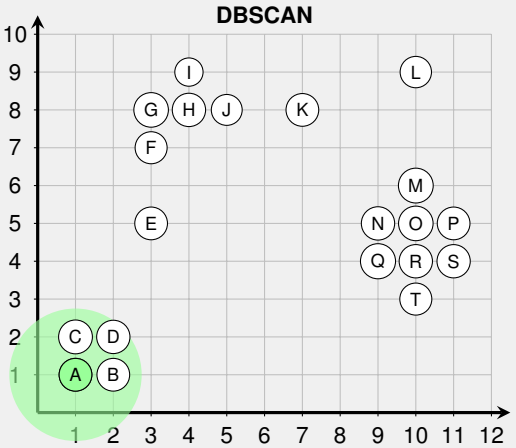
Hyperparameters:

$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

Example: DBSCAN



Hyperparameters:

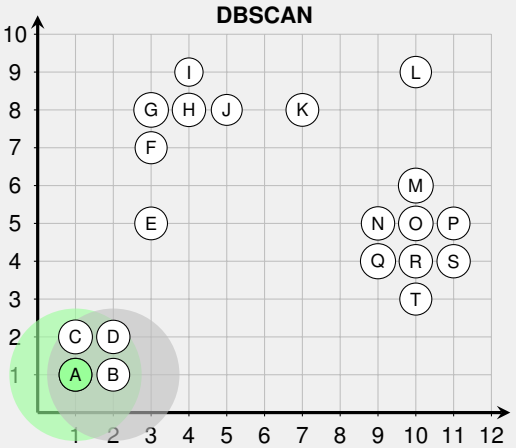
$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

A B C D

Example: DBSCAN



Hyperparameters:

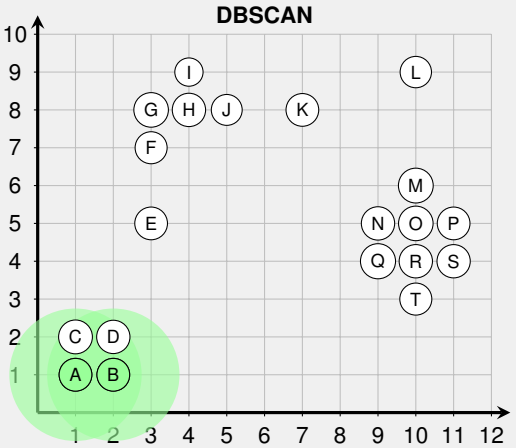
$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

A B C D

Example: DBSCAN



Hyperparameters:

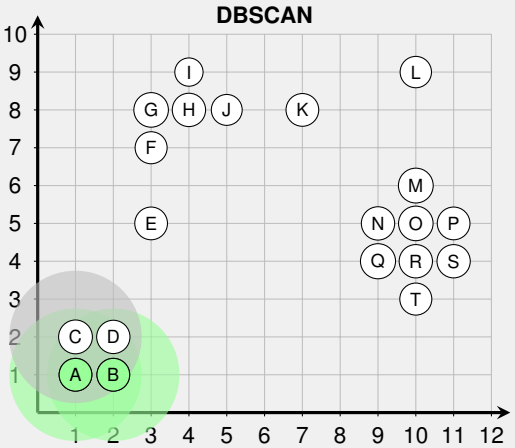
$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

A B C D

Example: DBSCAN



Hyperparameters:

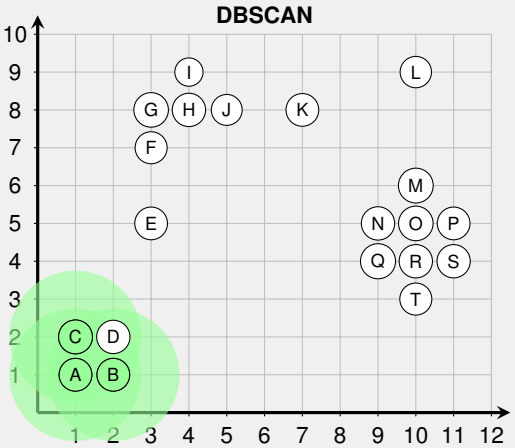
$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

A B C D

Example: DBSCAN



Hyperparameters:

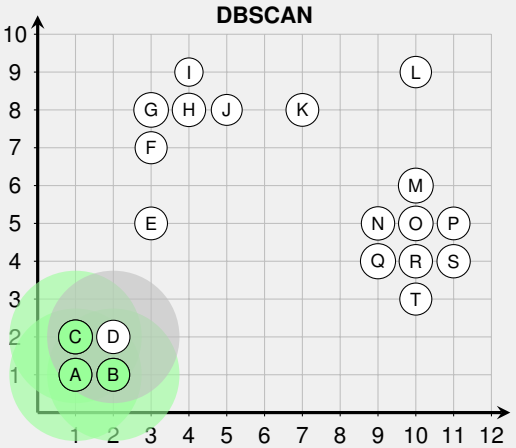
$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

A B C D

Example: DBSCAN



Hyperparameters:

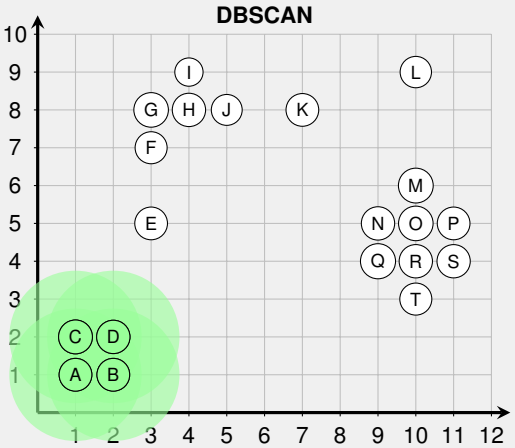
$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

A B C D

Example: DBSCAN



Hyperparameters:

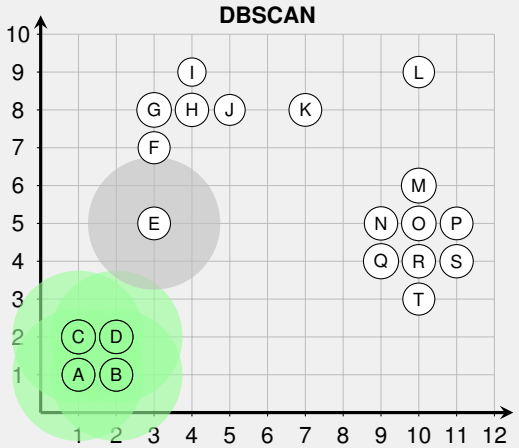
$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

A B C D

Example: DBSCAN



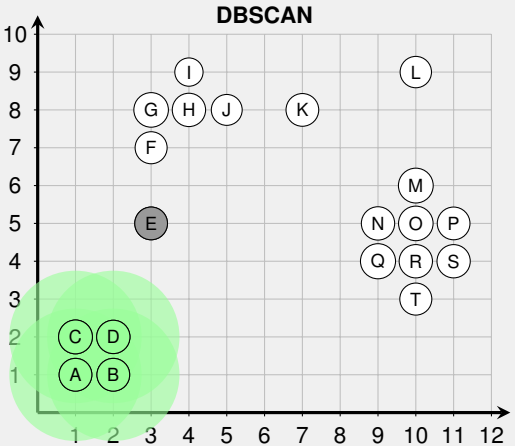
Hyperparameters:

$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

Example: DBSCAN



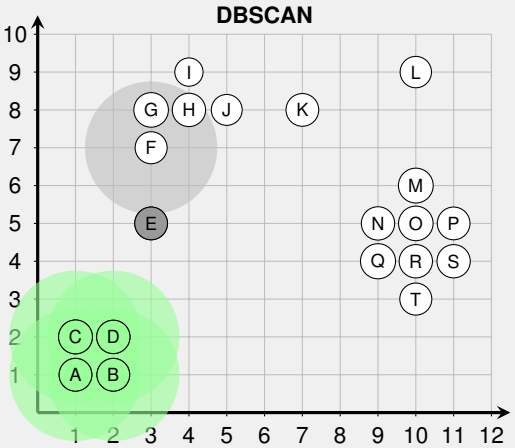
Hyperparameters:

$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

Example: DBSCAN



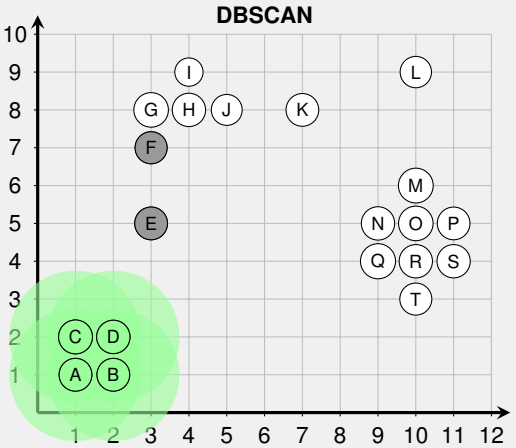
Hyperparameters:

$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

Example: DBSCAN



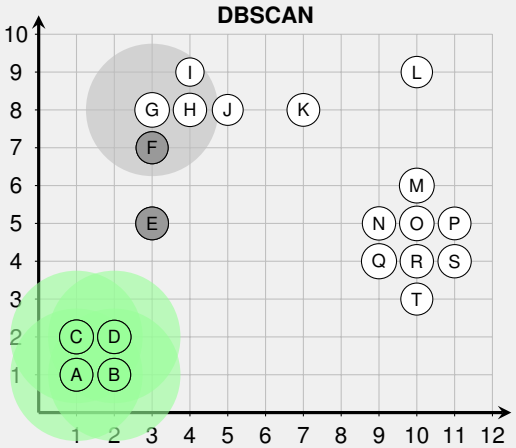
Hyperparameters:

$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

Example: DBSCAN



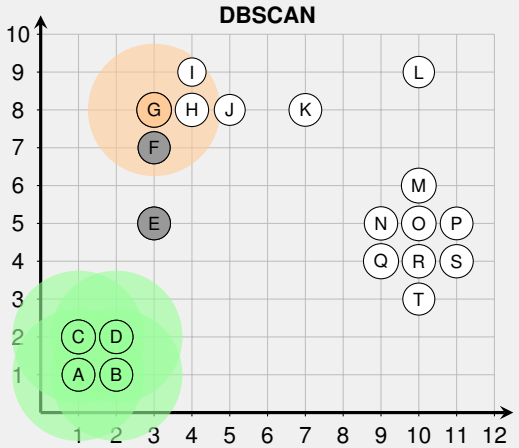
Hyperparameters:

$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

Example: DBSCAN



Hyperparameters:

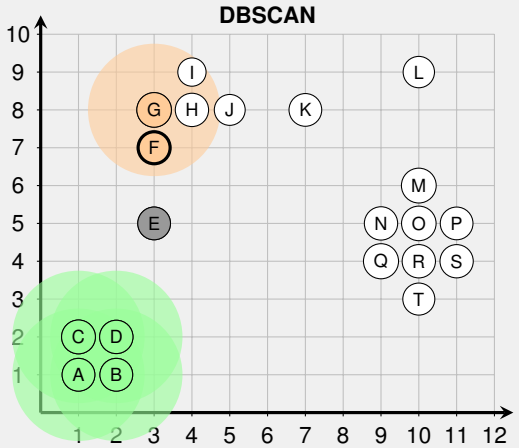
$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

F G H I

Example: DBSCAN



Hyperparameters:

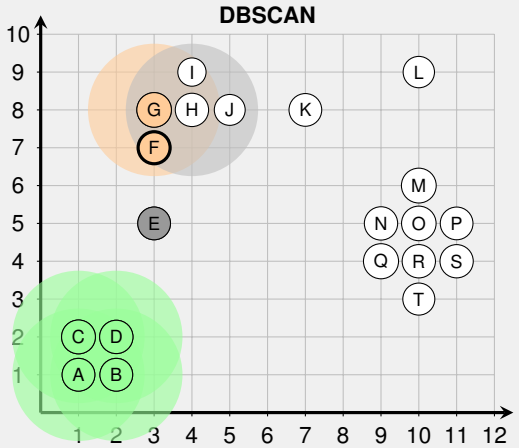
$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

F G H I

Example: DBSCAN



Hyperparameters:

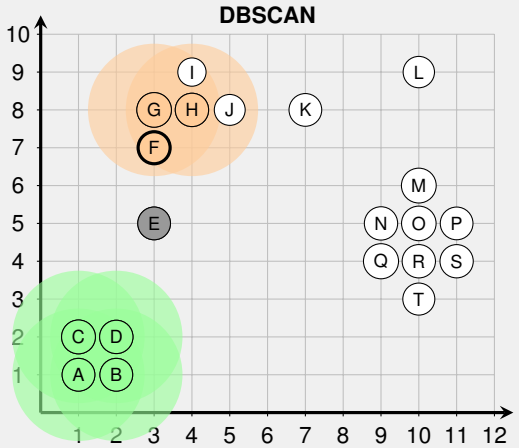
$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

F G H I

Example: DBSCAN



Hyperparameters:

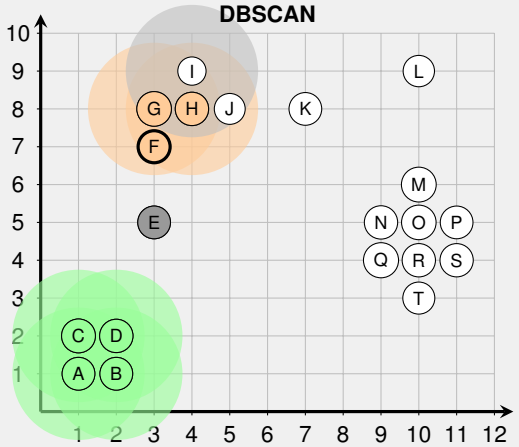
$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

F G H I J

Example: DBSCAN



Hyperparameters:

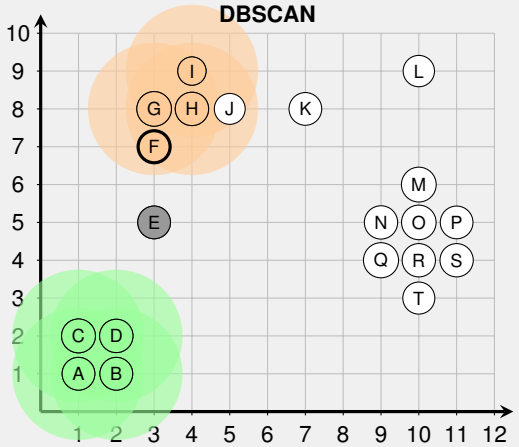
$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

F G H I J

Example: DBSCAN



Hyperparameters:

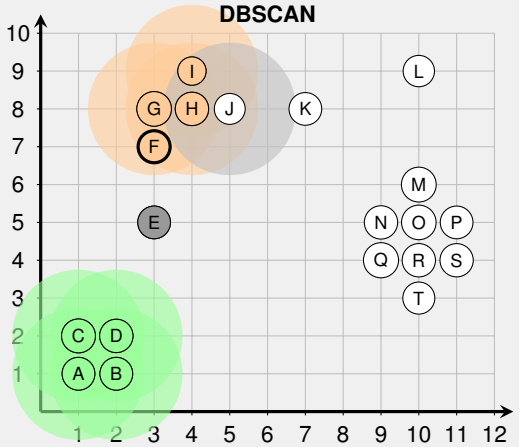
$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

F G H I J

Example: DBSCAN



Hyperparameters:

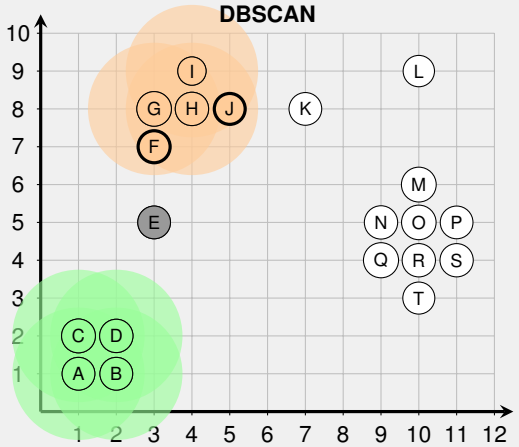
$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

F G H I J

Example: DBSCAN



Hyperparameters:

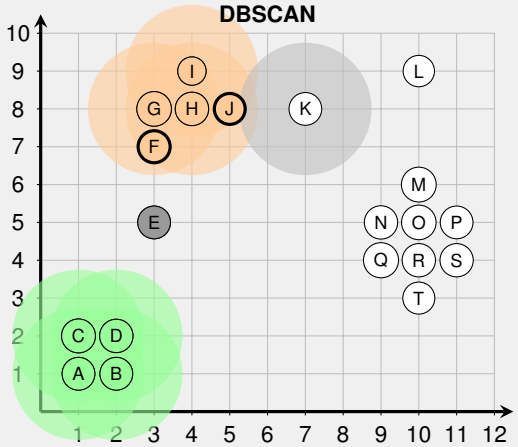
$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

F G H I J

Example: DBSCAN



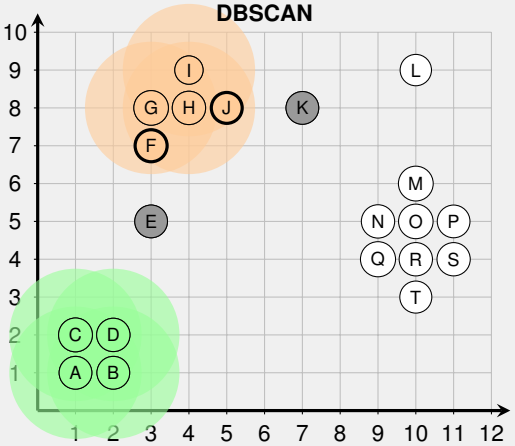
Hyperparameters:

$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

Example: DBSCAN



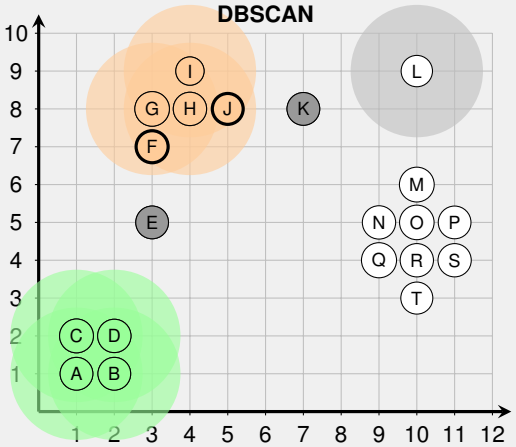
Hyperparameters:

$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

Example: DBSCAN



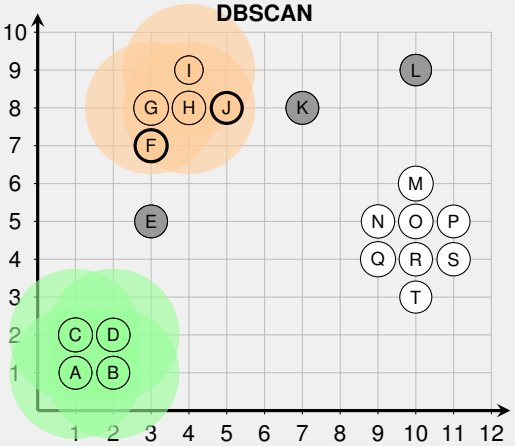
Hyperparameters:

$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

Example: DBSCAN



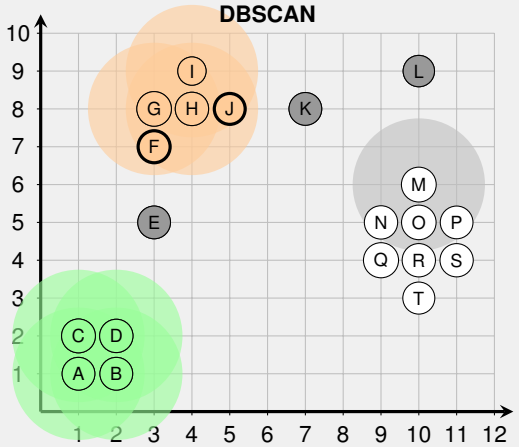
Hyperparameters:

$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

Example: DBSCAN



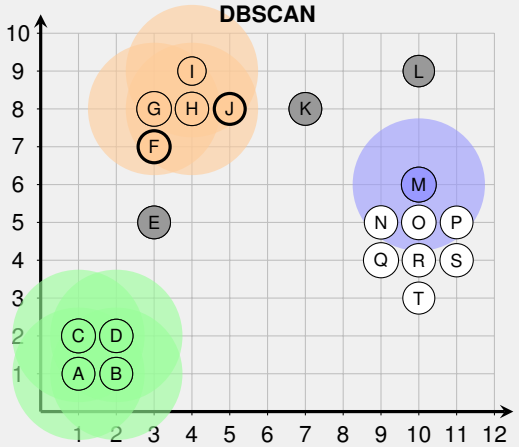
Hyperparameters:

$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

Example: DBSCAN



Hyperparameters:

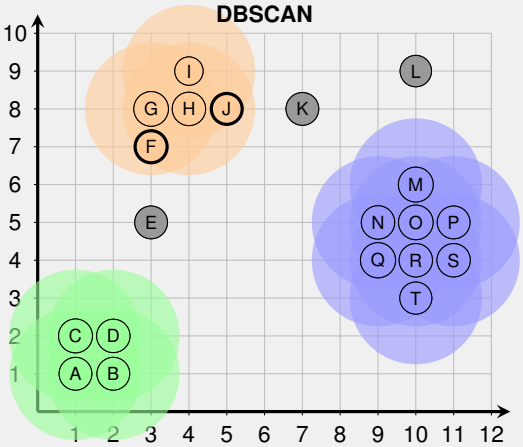
$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

M N O P

Example: DBSCAN



Hyperparameters:

$$\texttt{eps} = 1.75$$

$$\texttt{minPts} = 4$$

Seed list:

M N O P Q R S T

Section:

Mean-Shift Clustering

General Idea of the Algorithm
Recap: Kernel Density Estimation
Derivation of the Mean-Shift Vector
Final Algorithm

Introduction to Mean-Shift Clustering

- The **mean-shift** algorithm is a **non-parametric** clustering technique
- **Basic idea of the algorithm:**
  - Compute the density at each data point using a **kernel density estimator**
  - Move (a copy of) each data point into the direction of the maximum increase of the density
  - All data points ending up in (roughly) the same spot form a cluster

Advantages:

- We do **not need prior knowledge of the number of clusters**
- The algorithm does **not constrain the shape of the clusters**

Recap: Kernel Density Estimation

- **Recall:** The non-parametric density is given by:

$$p(\boldsymbol{x}) = \frac{1}{Nh^D} \sum_{n=1}^{N} K\left(\frac{\boldsymbol{x} - \boldsymbol{x}^n}{h}\right) \tag{2}$$

- K is the kernel function, e. g. the GAUSSian kernel:

$$K(\boldsymbol{u}) := \frac{1}{\left(\sqrt{2\pi}\right)^D} \exp\left(-\frac{\|\boldsymbol{u}\|^2}{2}\right) \tag{3}$$

Profile of a Kernel

- A kernel K can be written in terms of its **profile** k:
  (c_D is a normalization constant which ensures that the kernel integrates to 1)

$$K(\boldsymbol{u}) := c_D k\left(\|\boldsymbol{u}\|^2\right) \tag{4}$$

- **Example:** GAUSSian kernel – equation (3):

$$k(x) := \exp\left(-\frac{1}{2}x\right) \tag{5}$$

$$c_D := \frac{1}{\left(\sqrt{2\pi}\right)^D} \tag{6}$$

Kernel Density Estimator with Kernel Profile

The kernel density estimation becomes:

$$p(\boldsymbol{x}) = \frac{1}{Nh^D} \sum_{n=1}^{N} K\left(\frac{\boldsymbol{x} - \boldsymbol{x}^n}{h}\right)$$

$$= \frac{c_D}{Nh^D} \sum_{n=1}^{N} k\left(\left\|\frac{\boldsymbol{x} - \boldsymbol{x}^n}{h}\right\|^2\right) \tag{7}$$

We have to compute the gradient of $p(\boldsymbol{x})$ to know in which direction the density increases the most!

Gradient of the Kernel Density

- The gradient of $p(\boldsymbol{x})$ is

$$\nabla p(\boldsymbol{x}) = \frac{c_D}{Nh^D} \sum_{n=1}^{N} k' \left(\left\| \frac{\boldsymbol{x} - \boldsymbol{x}^n}{h} \right\|^2 \right) \cdot 2 \cdot \frac{\boldsymbol{x} - \boldsymbol{x}^n}{h} \cdot \frac{1}{h}$$

$$= \frac{2c_D}{Nh^{D+2}} \sum_{n=1}^{N} \left(\boldsymbol{x} - \boldsymbol{x}^n \right) k' \left(\left\| \frac{\boldsymbol{x} - \boldsymbol{x}^n}{h} \right\|^2 \right) \tag{8}$$

- As a next step, we introduce $g(\boldsymbol{x}) := -k'(\boldsymbol{x})$ (the profile of the **shadow kernel** G) and further rewrite the gradient

Gradient of the Kernel Density (Ctd.)

$$\begin{aligned}\nabla p(\boldsymbol{x}) &= \frac{2c_D}{Nh^{D+2}} \sum_{n=1}^{N} \left(\boldsymbol{x}^n - \boldsymbol{x}\right) g\left(\left\| \frac{\boldsymbol{x} - \boldsymbol{x}^n}{h} \right\|^2\right) \\ &= \frac{2c_D}{Nh^{D+2}} \left[\sum_{n=1}^{N} g\left(\left\| \frac{\boldsymbol{x} - \boldsymbol{x}^n}{h} \right\|^2\right)\right] \underbrace{\left[\frac{\sum_{n=1}^{N} \boldsymbol{x}^n g\left(\left\| \frac{\boldsymbol{x}-\boldsymbol{x}^n}{h} \right\|^2\right)}{\sum_{n=1}^{N} g\left(\left\| \frac{\boldsymbol{x}-\boldsymbol{x}^n}{h} \right\|^2\right)} - \boldsymbol{x}\right]}_{=: \boldsymbol{m}(\boldsymbol{x})} \end{aligned} \tag{9}$$

We call $\boldsymbol{m}(\boldsymbol{x})$ the **mean-shift vector** for $\boldsymbol{x}$ *(it points into the direction of maximum increase of the density at $\boldsymbol{x}$)*

Mean-Shift Algorithm

Input: $\boldsymbol{X} = \left\{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\right\}$, bandwidth h, kernel g

Set $t \longleftarrow 0$ and $\boldsymbol{x}^{1,t} := \boldsymbol{x}^1, \ldots, \boldsymbol{x}^{N,t} := \boldsymbol{x}^N$

Remark: $\boldsymbol{x}^{n,t}$ refers to the n-th data point at timestep t

while *not converged* **do**

  for $n \longleftarrow 1$ **to** N **do**

    Compute the mean-shift vector $\boldsymbol{m}\left(\boldsymbol{x}^{n,t}\right) \longleftarrow \frac{\sum_{i=1}^{N} \boldsymbol{x}^i g\left(\left\|\frac{\boldsymbol{x}^{n,t} - \boldsymbol{x}^i}{h}\right\|^2\right)}{\sum_{i=1}^{N} g\left(\left\|\frac{\boldsymbol{x}^{n,t} - \boldsymbol{x}^i}{h}\right\|^2\right)} - \boldsymbol{x}^{n,t}$

    Update the data point: $\boldsymbol{x}^{n,t+1} \longleftarrow \boldsymbol{x}^{n,t} + \boldsymbol{m}\left(\boldsymbol{x}^{n,t}\right)$

  end

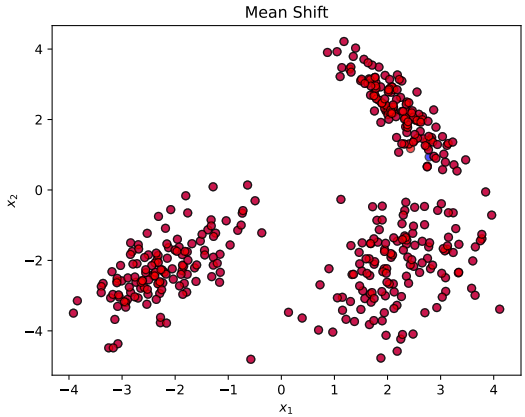
  $t \longleftarrow t + 1$

end

Create a cluster comprising all data points shifted to the same local maximum *(with some tolerance)*

return *the set of clusters* $\mathfrak{C}$

Visualization: Mean-Shift Clustering



Section:

Wrap-Up

Summary
Recommended Literature
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. KMeans)
  - Hierarchical clustering (agglomerative, divisive)
  - Affinity-based clustering (e. g. DBSCAN, mean-shift clustering)

Recommended Literature

1. **KMeans**
   - [BISHOP.2006], chapter 9.1
2. **Mean-Shift Clustering**
   - [DEMIROVIC.2015]

(For free PDF versions, see list in GitHub readme!)

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 KMeans. What are disadvantages?
4. What is a dendrogram?
5. Describe what DBSCAN works!
6. What is affinity-based clustering? How does it differ from KMeans?
7. What is the main idea of mean-shift clustering?

What's next...?

I	Machine Learning Introduction
II	Optimization Techniques
III	Bayesian Decision Theory
IV	Non-parametric Density Estimation
V	Probabilistic Graphical Models
VI	Linear Regression
VII	Logistic Regression
VIII	Deep Learning
IX	Evaluation
X	Decision Trees
XI	Support Vector Machines
XII	Clustering
• **XIII**	Principal Component Analysis
XIV	Reinforcement Learning
XV	Advanced Regression

Thank you very much for the attention!

*** * * Artificial Intelligence and Machine Learning * * ***

Topic: Clustering Algorithms

Term: Summer term 2025

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

Do you have any questions?