DBSCAN

ML2: AI Concepts and Algorithms (SS2025)

Faculty of Computer Science and Applied Mathematics
University of Applied Sciences Technikum Wien

Lecturer: Rosana de Oliveira Gomes
Authors: B. Knapp, M. Blaickner, S. Rezagholi, R.O. Gomes

Classification
KNN classification
Classification trees
Ensembles & boosting
Random Forest
Logistic regression
Naive Bayes
Support vector machines
Neural networks

Regression
KNN regression
Regression trees
Linear regression
Multiple regression
Ridge and Lasso regression
Neural networks

Supervised learning

Data handling
EDA
Data cleaning
Feature selection
Class balancing
etc

Clustering
k-means
Hierachical clustering
DB-scan

Non-supervised learning

AI

Dimensionality reduction
PCA / SVD
tSNE
MDS

Reinforcement learning
Covered in a separate lecture.

FH University of Applied Sciences TECHNIKUM WIEN

Clustering Recap

- Unsupervised learning method that aggregates data into different groups (clusters) according to their similarities.
- Given the unlabeled data, clustering does not have a metric to measure algorithmic performance – often such methods rely on domain interpretation.
- Common Algorithms: **K-Means Clustering**, **DBScan**, Hierarchical clustering, Mean-Shift Clustering, **Gaussian Mixture Model (GMM)**.

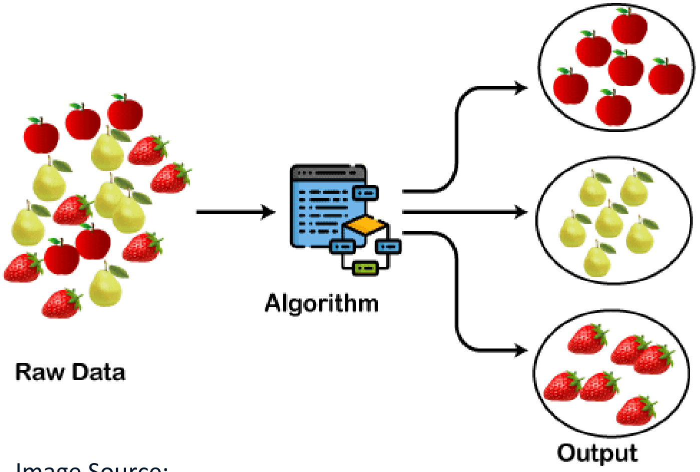


Image Source:
https://static.javatpoint.com/tutorial/machine-learning/images/clustering-in-machine-learning.png

Types of clustering algorithms

Exclusive clustering: A single data point exclusively belongs to one cluster.
Overlapping clustering: A single data point may belong to multiple clusters.
[Hierarchical clustering]: Groups are created according to hierarchical similarities.
Probalistic clustering: Clusters are created using probability distribution.

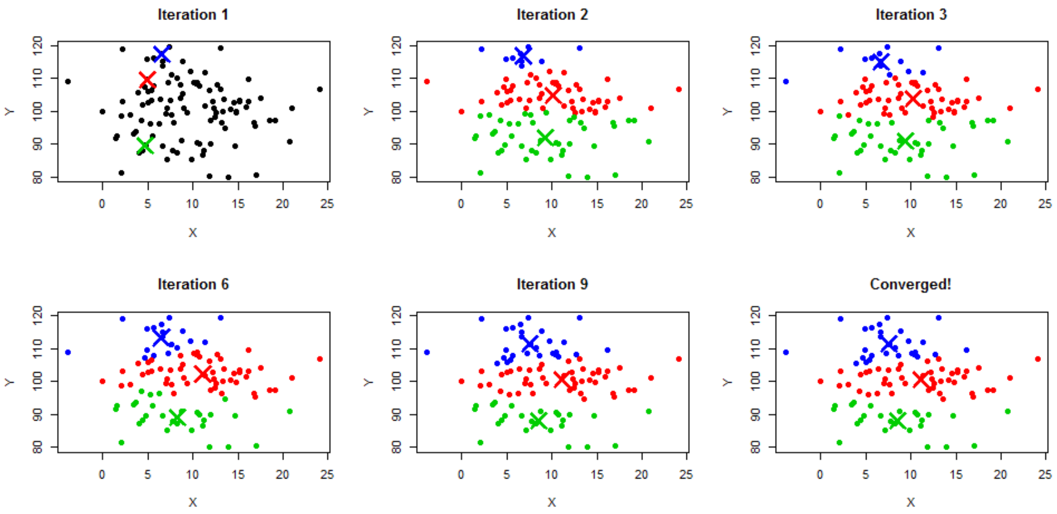
FH University of Applied Sciences TECHNIKUM WIEN

K-Means Recap

K-Means is the most popular clustering method.

It is a method that divides the dataset into k-clusters (defined by the user).

Each data point belong to only one cluster.



Algorithm:

1. Specify number of clusters *k.*
2. Randomly initialize the cluster‘s centroids (see Iteration 1).
3. Calculate distance between each data points and centroids and assign data points to the nearest centroid.
4. Recalculate mean of the centroid based on all assigned data points.
5. Repeat until convergence.

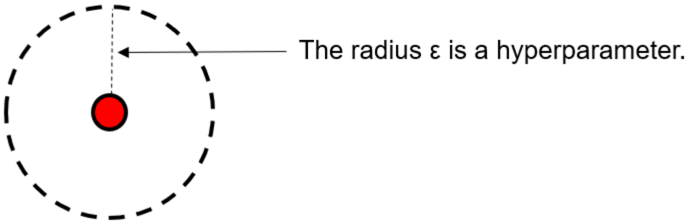
DBSCAN: **D**ensity-**B**ased **S**patial **C**lustering for **A**pplications with **N**oise

Clusters are dense spaces in the region separated by lower-density regions.

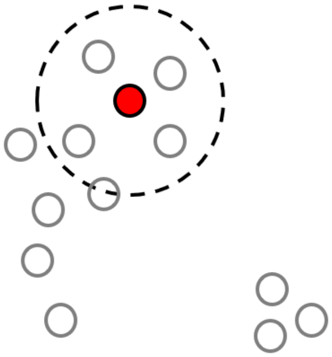
- DBSCAN is a clustering algorithm sensitive to density which explicitly allows *noise points* (points that are not in any cluster).
- In density-based clustering points are subdivided into **dense regions** separated by **low density** regions.
    1. How do we measure density?
    2. What is a dense region?

DBSCAN: Epsilon

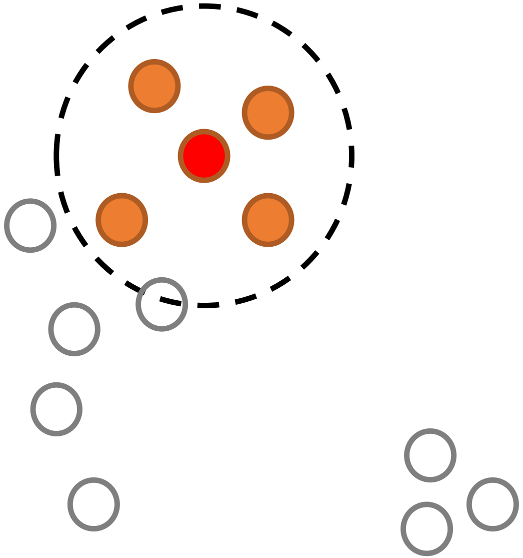
ε-k-Dense Region: A ball of radius **ε** that contains at least **k** points.



The ε-ball around the red point includes 4 other points.



DBSCAN: Hyperparameter k



k tells us how many points we need (within ε) to start a new cluster.

Example: k=4.

Step 1:

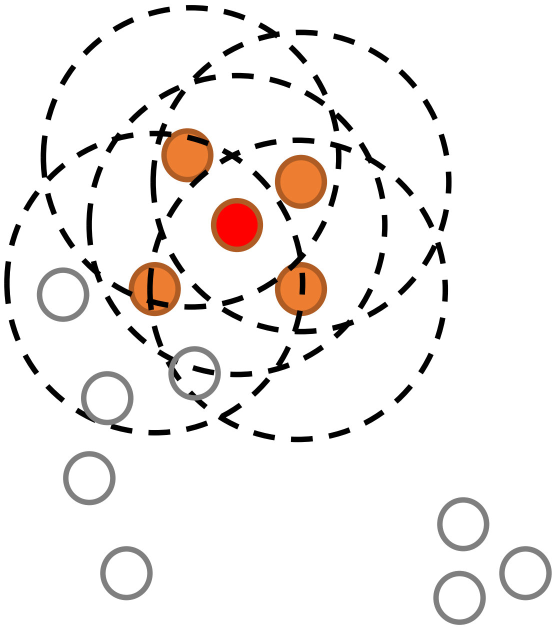
The **red** point is a **randomly** selected starting point.

The **orange** points are in the ε-ball around the red point.
They are called **core points**.

Since k <= 5 we start a new cluster around the red point.

FH University of Applied Sciences TECHNIKUM WIEN

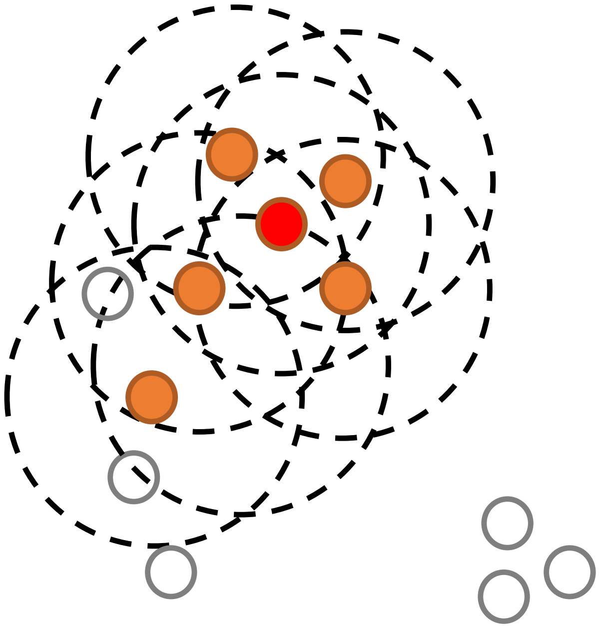
DBSCAN: Growing a Cluster



Step 2:

We consider the **ε-balls around all core points**.

DBSCAN: Growing a Cluster



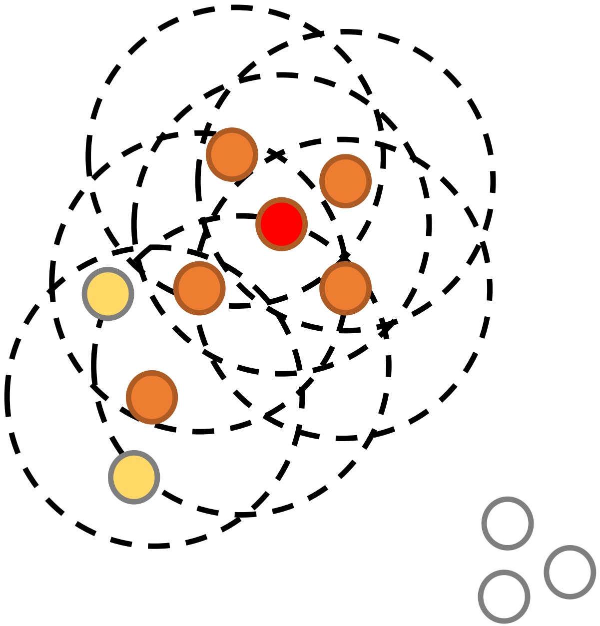
Step 3:

Points within ε-reach of core points are **added** to the cluster.

If the newly added points are **core points** (their ε-ball contains more than k points) then we consider their ε-balls (as in step 2) in an iterative procedure.

This procedure is **repeated** until no more points can be reached.

DBSCAN: Growing a Cluster



Border points:

Points that can be reached from a cluster but that do not have enough ε-neighbours to be **core points** are called **border points**.

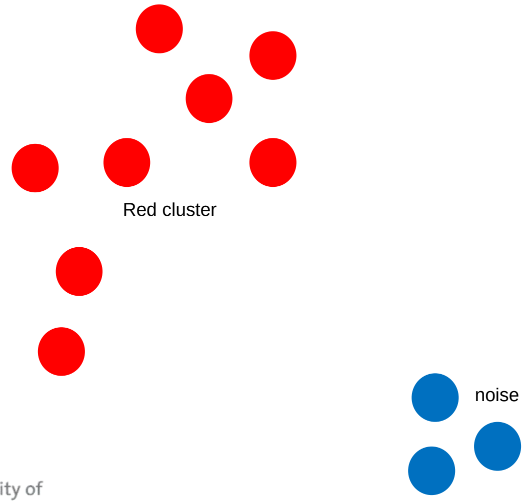
Border points are cluster members but they are no longer used in the iterative cluster-growing procedure.

Points that cannot be reached at all are called **noise points**.

DBSCAN: Finished Cluster

The red cluster is finalized.

DBSCAN: Final Result

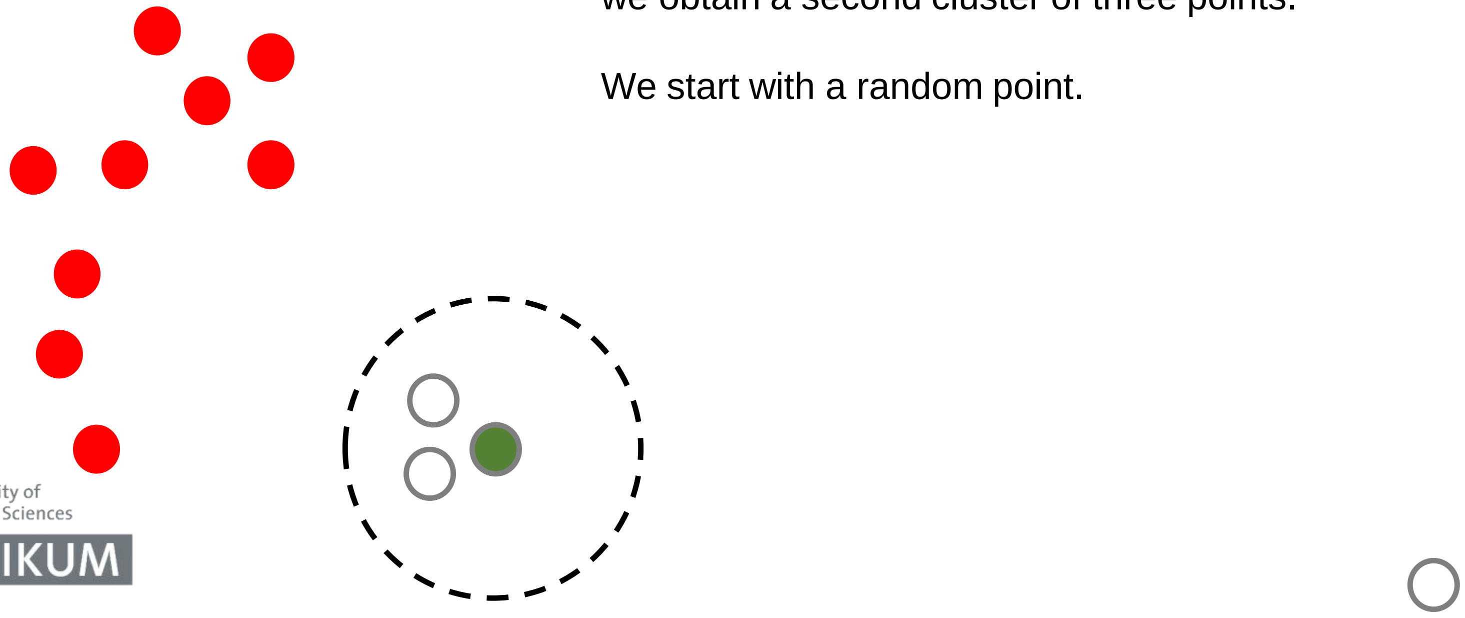


We may start another cluster with the remaining points (depends on k and ε).

Since k=4 and none of the remaining points have more than 4 ε-neighbors all remaining points are classified as noise points.

This is the final result for k=4: One cluster and the rest is noise.

DBSCAN: Results for Different Hyperparameters

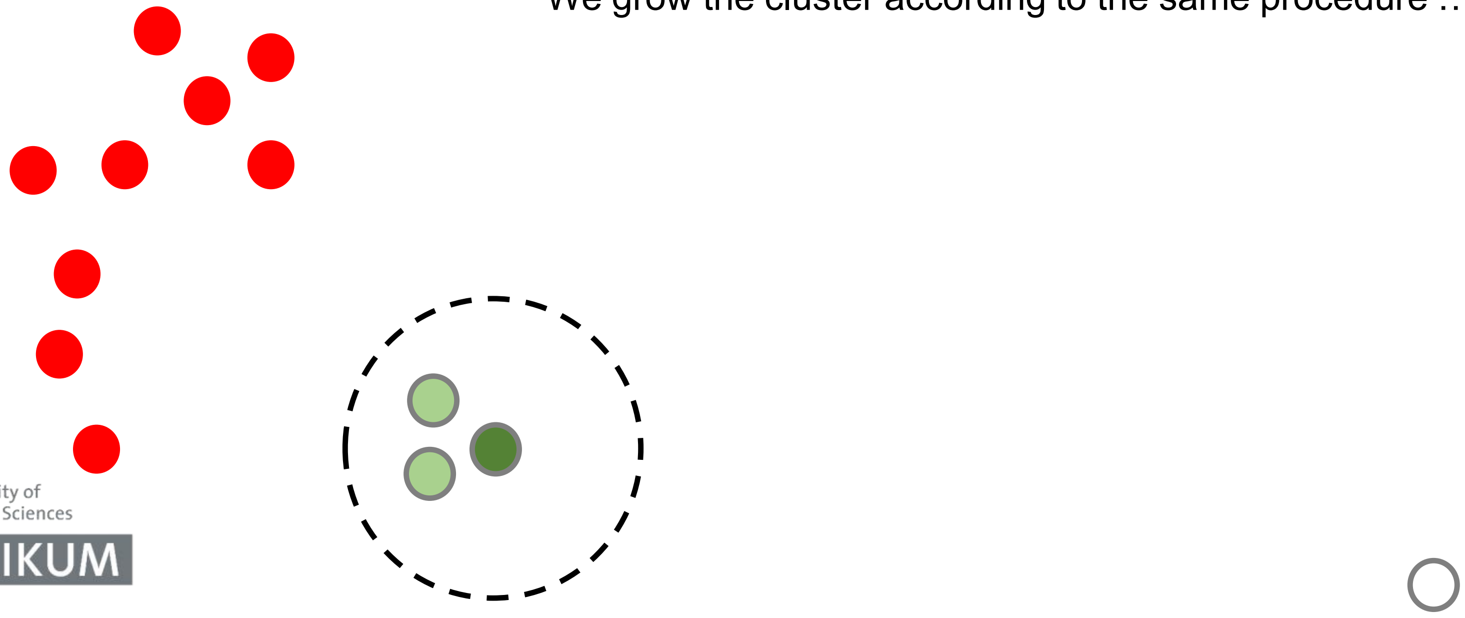


If we use the same value for ε but **change to k=3** then we obtain a second cluster of three points.

We start with a random point.

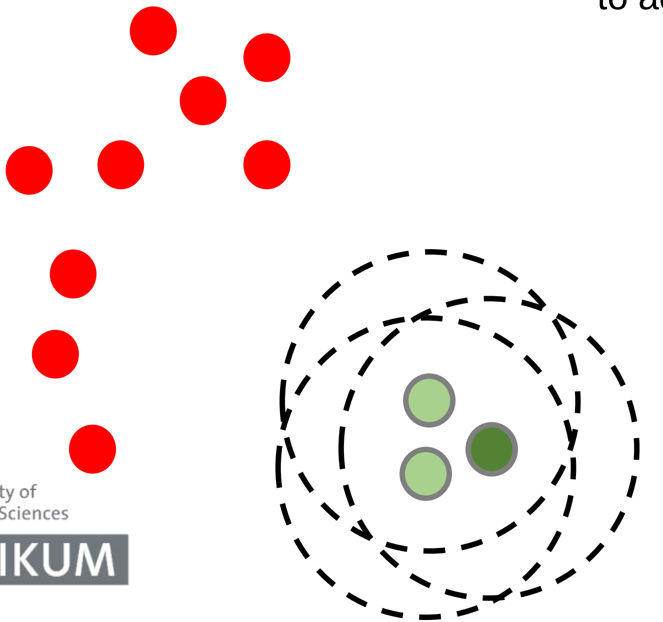
DBSCAN: Results for Different Hyperparameters

We grow the cluster according to the same procedure …



DBSCAN: Results for Different Hyperparameters

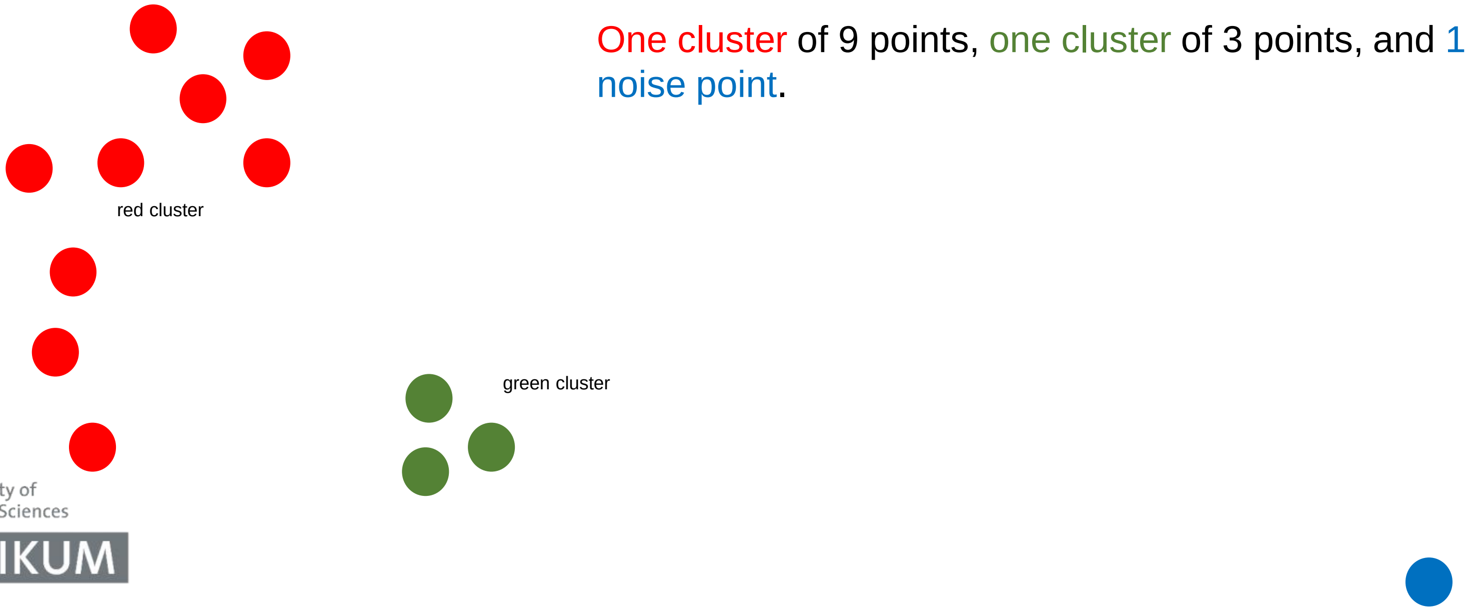
All 3 **green** points are core points but they do not lead to additional cluster members.



DBSCAN: Results for Different Hyperparameters

The final result for k=3:

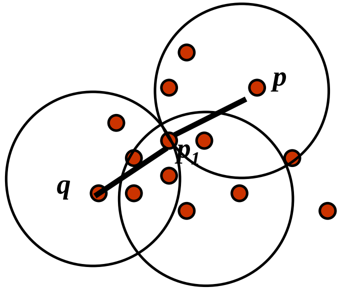
One cluster of 9 points, one cluster of 3 points, and 1 noise point.



Density-Connected Points

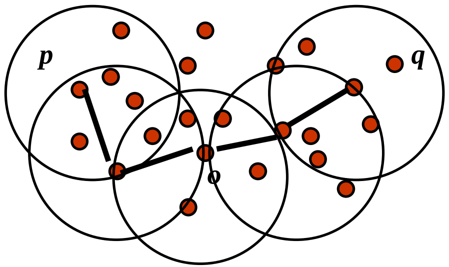
Density edge

- A density edge exists between two core points q and p if there is a point p_1 such that both p and q are within distance ε of p_1.



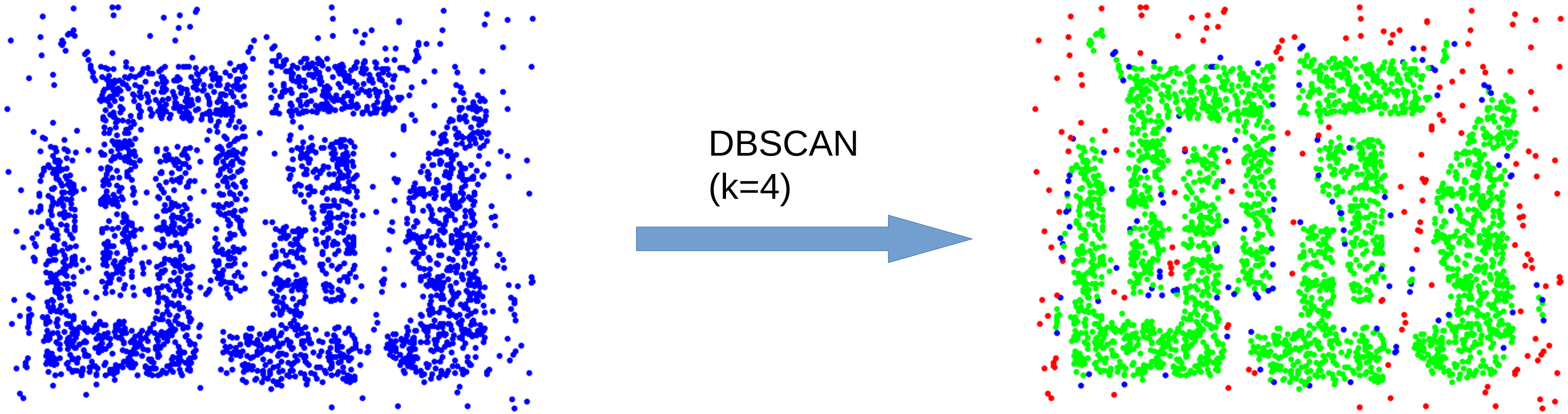
Density-connection

- A point p is **density-connected** to a point q if there is a **path of edges** from p to q.



FH University of Applied Sciences TECHNIKUM WIEN

DBSCAN: Core, Border, and Noise Points



Point types:

- Core points
- Border points
- Noise points

FH University of Applied Sciences
TECHNIKUM
WIEN

DBSCAN: Summary

- A point is a **core point** if it has more than a specified number of points (hyperparameter k) within its ε-radius.
- The number of ε-neighbors of a **border point** is less than k but the point itself is in the ε-neighborhood of a core point.
- A **noise point** is any point that is neither a core point nor a border point.

FH University of Applied Sciences TECHNIKUM WIEN

Quiz

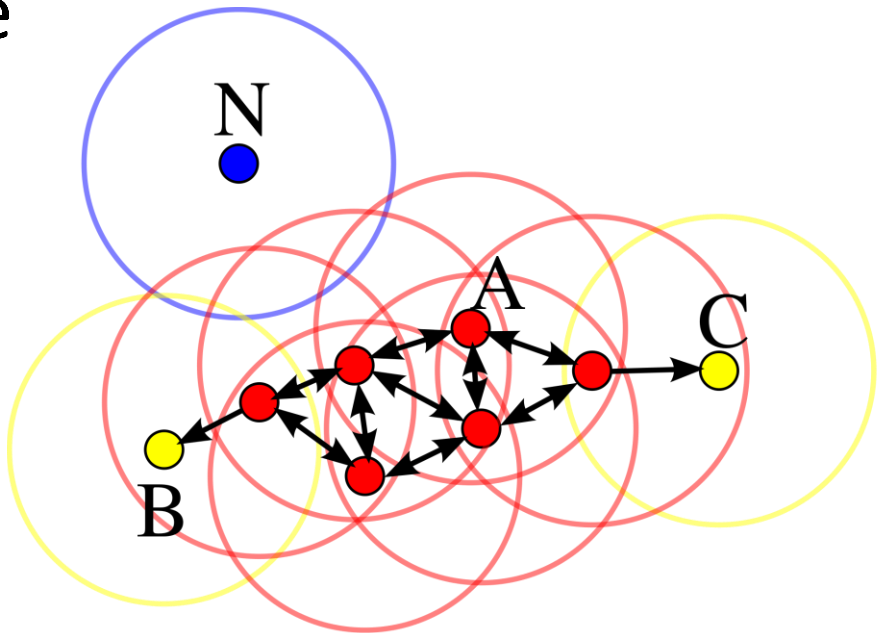
In DBSCAN, which of the following best describes the role of the minPts (k) parameter?

A. Minimum distance between two core points
B. Minimum number of points required to form a core point
C. Minimum variance within a cluster
D. Minimum number of clusters the algorithm should produce

Consider a dataset with two clusters of very different densities.
What is the most likely result of applying DBSCAN with a fixed ε?

A. Both clusters will be identified correctly
B. Only the denser cluster will be identified correctly; the sparser one may be split or treated as noise
C. DBSCAN will automatically adjust ε for each region
D. DBSCAN will merge both clusters into one

DBSCAN: Example



- **k=4**
- **Point A** and the other red points are **core points**, because each of the respective *ε-balls* contains at least 4 points **(including the point itself)**. They form a single cluster because they are all reachable from one another.
- **Points B** and **C** are **border points** since they are reachable from A (via other core points) and thus belong to A's cluster as well.
- **Point N** is a **noise point** since it is neither a core point nor directly reachable.

DBSCAN: Stochasticity of the Algorithm

- DBSCAN is **not deterministic**. Even with the same parameters you do not necessarily obtain identical results.
- **Border points** that are reachable from multiple clusters can be part of any of these clusters, **depending on the order of cluster processing** (determined by random choice of starting points).
- For the majority of points (the core points) the results will be identical.

FH University of Applied Sciences TECHNIKUM WIEN

DBSCAN: Determining ε and k

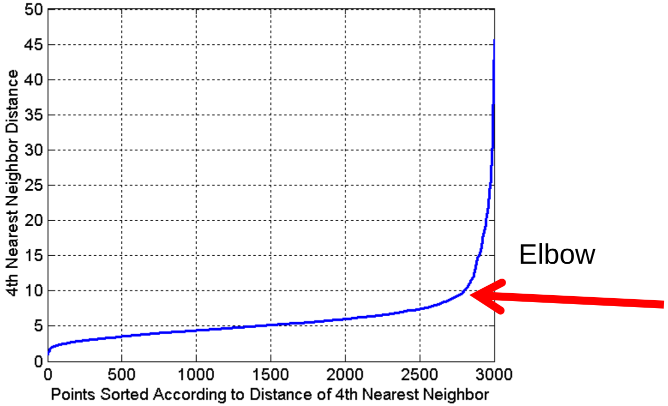
Ideally ε and k are determined by domain knowledge (example: a physical distance for ε).

Choosing minPts (k): common values of k are at least D+1, where D refers to the number of dimensions (features).
- In the case of noisy data, increase the value of k to avoid false clusters.

Choosing ε:

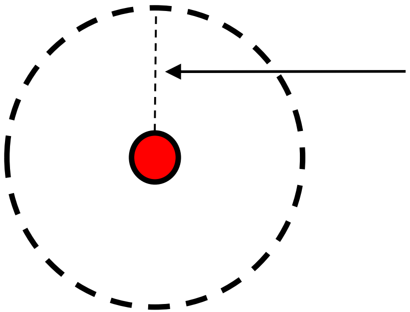
Elbow method: For points in a cluster, their k^{th} nearest neighbors are at roughly the same distance.

1. Compute the distance to each point's **k-th nearest neighbor**
2. Sort the distances in ascending order
3. Plot the sorted distances
4. Look for the **"elbow" point** in the plot — this is your candidate for ε.



FH University of Applied Sciences TECHNIKUM WIEN

DBSCAN: Alternatives to the Euclidean Distance



The radius ε is a hyperparameter.

Any metric (distance function satisfying the mathematical properties of a metric) can be used.

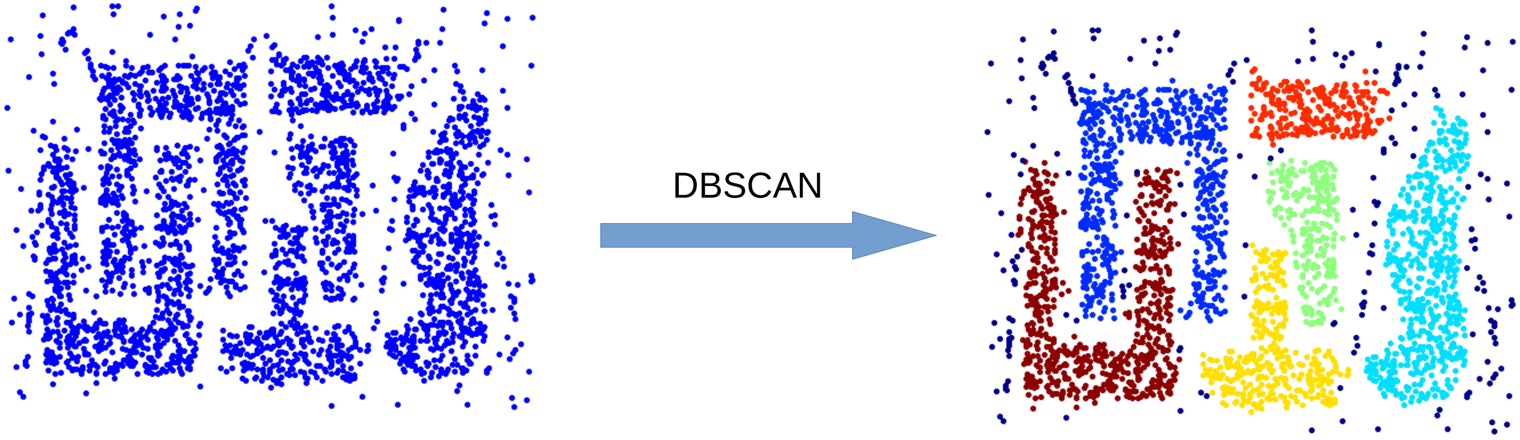
- <u>Euclidean metric:</u> default (works well in a **continuous vector space** where features are equally scaled and meaningful.

- <u>Non-Euclidean metric:</u> dependent of use case.
  - Categorical data → e.g. Jaccard distance
  - Text/Sequence data → e.g. cosine distance
  - GPS data → e.g. haversine distance
  - Graph-structured data → geodesic distance

Several metrics are implemented in SK-Learn:
https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise_distances.html#sklearn.metrics.pairwise_distances

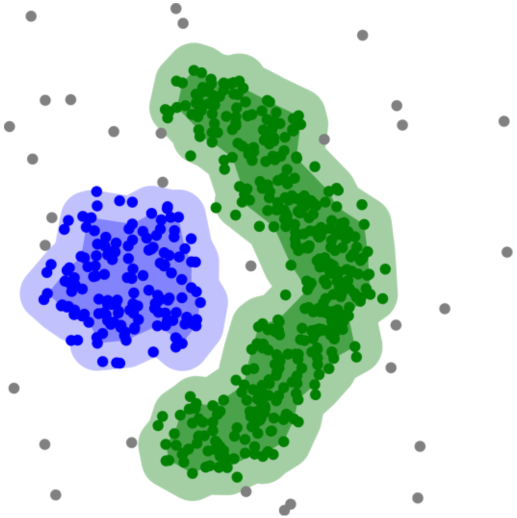
FH University of Applied Sciences TECHNIKUM WIEN

When does DBSCAN work well?

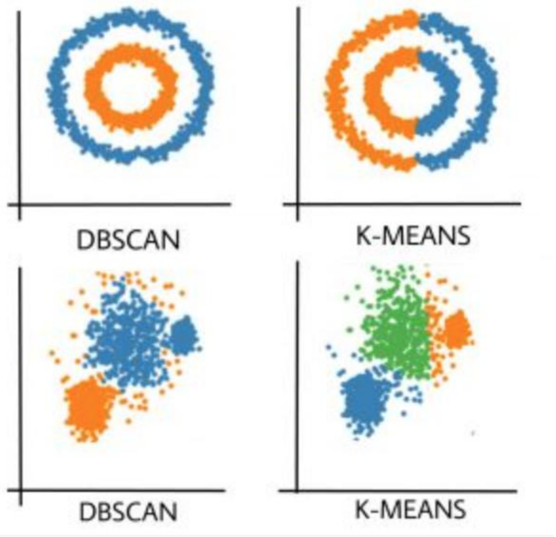


- **Clusters of arbitrary shape** – DBSCAN doesn’t assume spherical clusters.
- **Data with noise or outliers** – Naturally handles and labels noise points.
- **No need to predefine number of clusters** – The number emerges from the data.

When does DBSCAN work well?



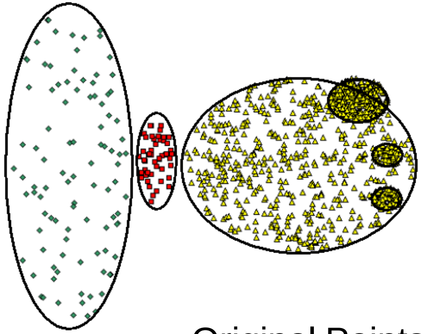
DBSCAN can find certain clusters that can not be obtained using the k-means algorithm.



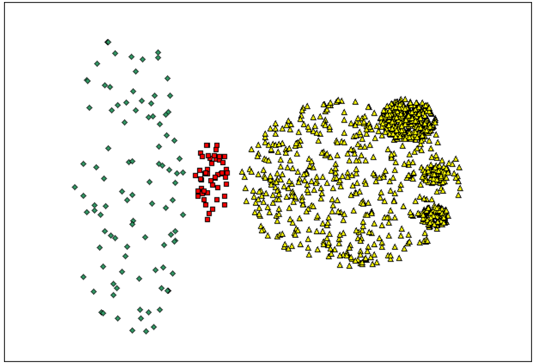
FH University of Applied Sciences TECHNIKUM WIEN

When does DBSCAN fail?

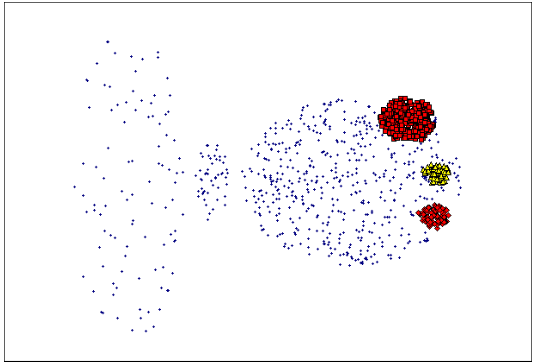
- **Varying densities** – A single global ε makes it hard to detect clusters of varying densities.
- **High-dimensional data** – Distance metrics lose meaning in high dimensions, leading to poor clustering.
- **Parameter sensitivity** – Performance can be very sensitive to ε and minPts (k); needs careful tuning.



Original Points

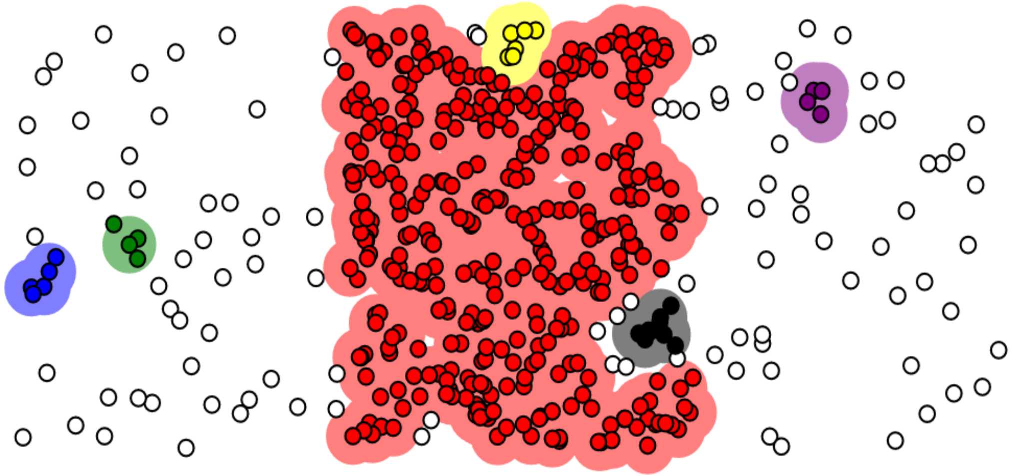


k=4, ε=9.75



k=4, ε=9.92

When does DBSCAN fail?

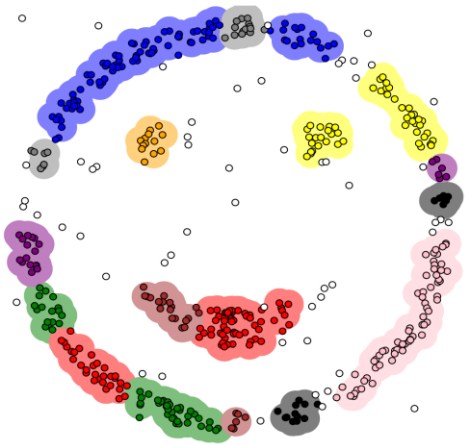


epsilon=0.5, k=4

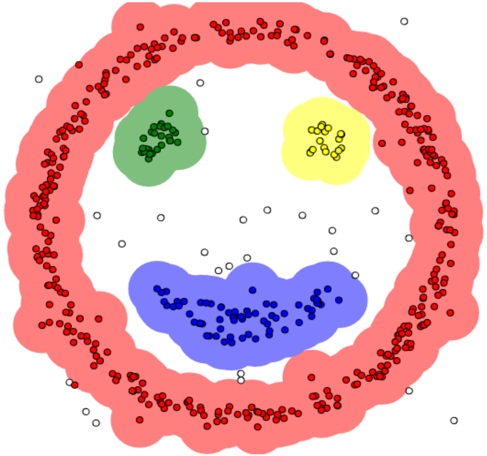
University of Applied Sciences
FH TECHNIKUM WIEN

[https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/]

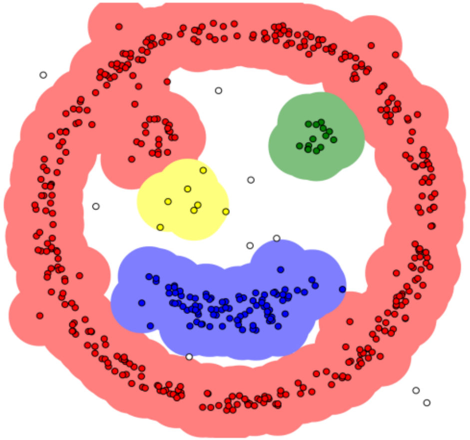
DBSCAN: More Counterexamples



epsilon=0.66, k=5



epsilon=1.32, k=5



epsilon=1.5, k=4

https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

FH University of Applied Sciences TECHNIKUM WIEN

Takeaway

- DBSCAN is **highly effective** for clustering **low-dimensional**, **noisy**, and **non-spherical** datasets where cluster shapes are complex and the number of clusters is unknown.
- DBSCAN is **less effective** for **high-dimensional** data or data with **clusters of varying density**.
- Hyperparameters:

Too small `ε` → many small clusters and noise

Too large `ε` → merges distinct clusters into one

Too small `minPts` → more noise, sensitive to outliers

Too large `minPts` → may miss small clusters

Next up: Dimensionality Reduction

FH University of Applied Sciences TECHNIKUM WIEN

Assignment: DBSCAN

a) Explain DBSCAN via visualizations.

Use self-made images or even hand drawings (of which you take a photo).

Use self-written explanations.

Do not copy from the lecture slides or the internet (neither text nor images).

b) Implement a (basic) version of DBSCAN.

Run it on 3 test sets of your choice. Run a library-version of k-means on the same test sets and compare the results with DBSCAN. Visualize the results for DBSCAN and k-means.

FH University of Applied Sciences TECHNIKUM WIEN

Literature

Giuseppe Bonaccorso. Machine Learning Algorithms - 2nd Edition. Birmingham: Packt Publishing, 2018.

FH University of Applied Sciences TECHNIKUM WIEN