Life Sciences and
Facility Management

Institute of
Computational Life Sciences

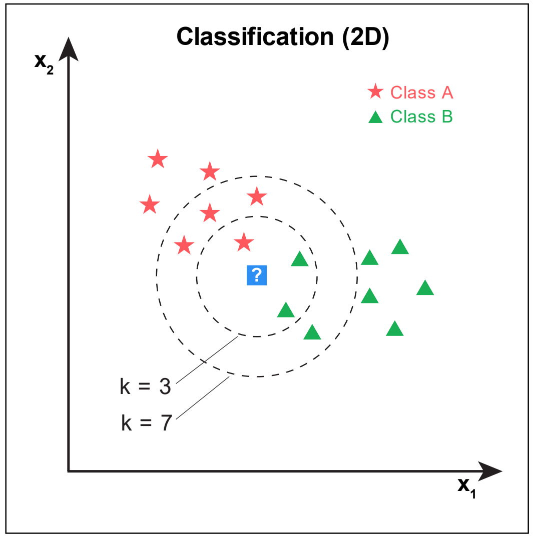
Clustering

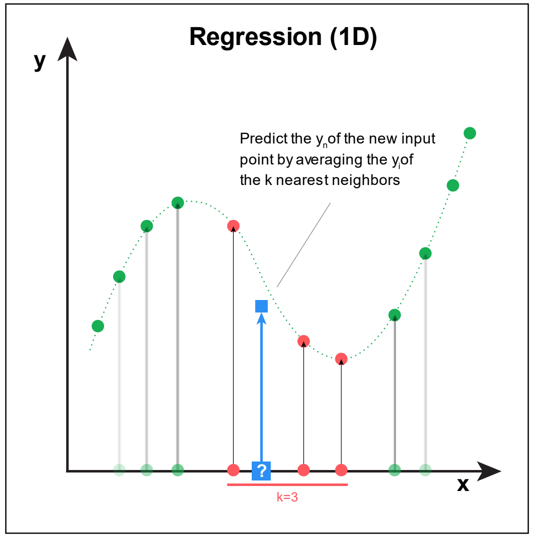
Machine Learning

Norman Juchler

Recap: k-nearest neighbors (kNN)

- The kNN method is an intuitive method to assign a class to a new data point based on the majority class of its k nearest neighbors.
- ~~It can be used for classification (supervised) and clustering (unsupervised) and even regression~~.
- It can be used for supervised (classification, regression) and even unsupervised tasks (anomaly detection, dimensionality reduction, ...)





Today

- Clustering
- Methods:
  - k-means
  - DBSCAN
  - and many more...
- Evaluation
  - What is the best clustering?

Introduction

Unsupervised learning

- In unsupervised learning the training data does **not** contain corresponding output values but only the input features $X \in \mathbb{R}^{n \times m}$
  - n: Number of samples
  - m: Number of features

- **Goal**: Model the underlying distribution of the data to
  - Discover hidden patterns that explain the data
  - Apply the model to new data

- **Challenges**:
  - Problem statement more open than in the supervised setting
  - The model evaluation is more difficult without expected output values

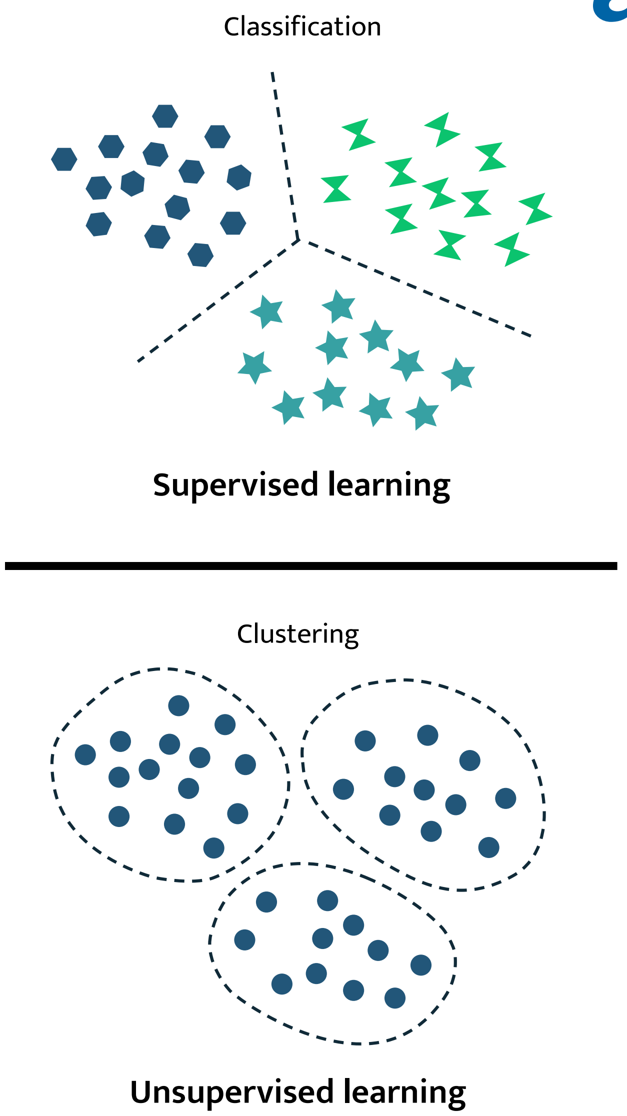


Image source: Link

Unsupervised learning

- Common tasks in unsupervised learning
  - Clustering: Grouping of similar data points based on features
  - Dimensionality reduction: Reducing the number of features
  - Anomaly detection: Identifying samples that deviate significantly

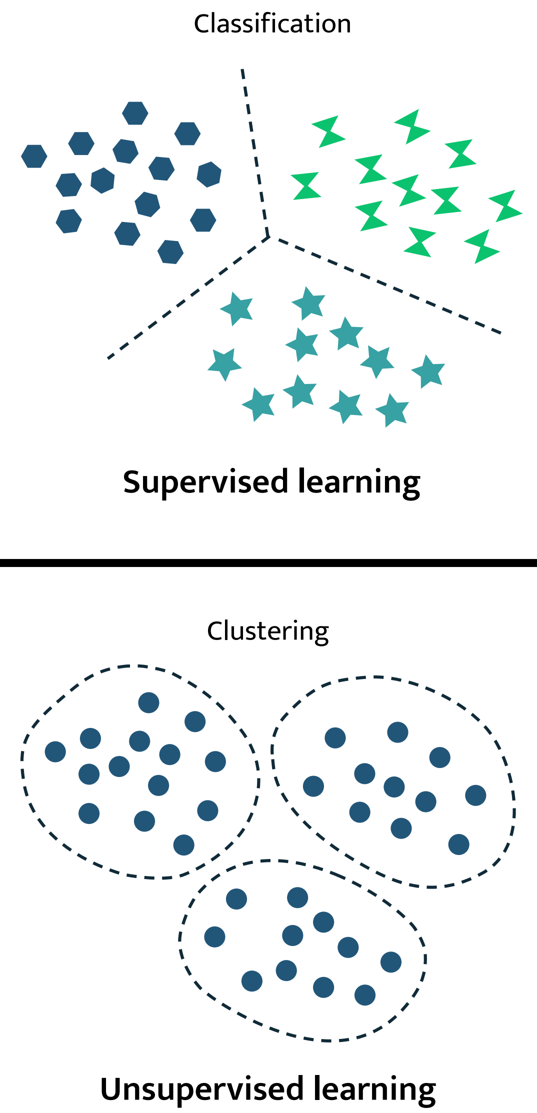


Image source: Link

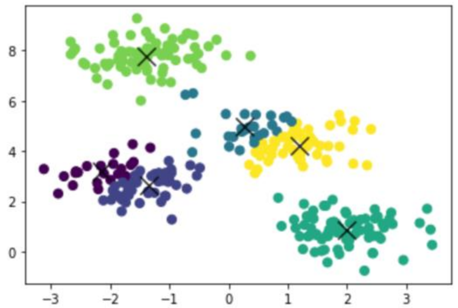
Clustering

- **Goal**: Find subgroups of data points that are similar based on their features

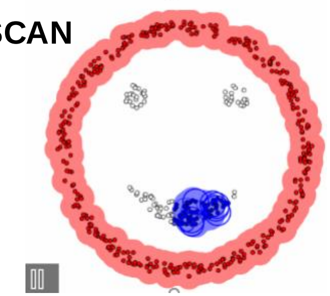
- **Problem**: Given *N* data points, separate them into *K* (K ≪ N) clusters

- **Variants**:
  - Hard clustering: Each data point is assigned a unique cluster (belongs to only one cluster)
  - Soft clustering: Each data point x_i is assigned a probability p_{ki} that it belongs to cluster k such that $\sum_k p_{ki} = 1$ (typically associated with probabilistic model like Gaussian Mixture Models-GMMs)

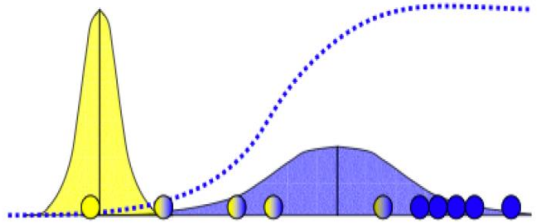
K-Means



DBSCAN



GMM



Clustering: Why?

Why would we want to cluster the data?

- **Data understanding**: find „natural“ clusters and describe their properties
- **Data class identification**: find useful and suitable groupings
- **Data reduction**: find representatives for homogenous groups
- **Outlier / anomaly detection**: find unusual data objects

Clustering: Examples

- **Text analysis**: Group the articles in a large corpus based on similar topics
- **Disease bioinformatics**: Cluster patients based on gene expression or protein markers to identify genes related to certain diseases or to discover drug targets
- **Sports science**: Find players with similar behavior based on characteristics of their play

Clustering: Method overview

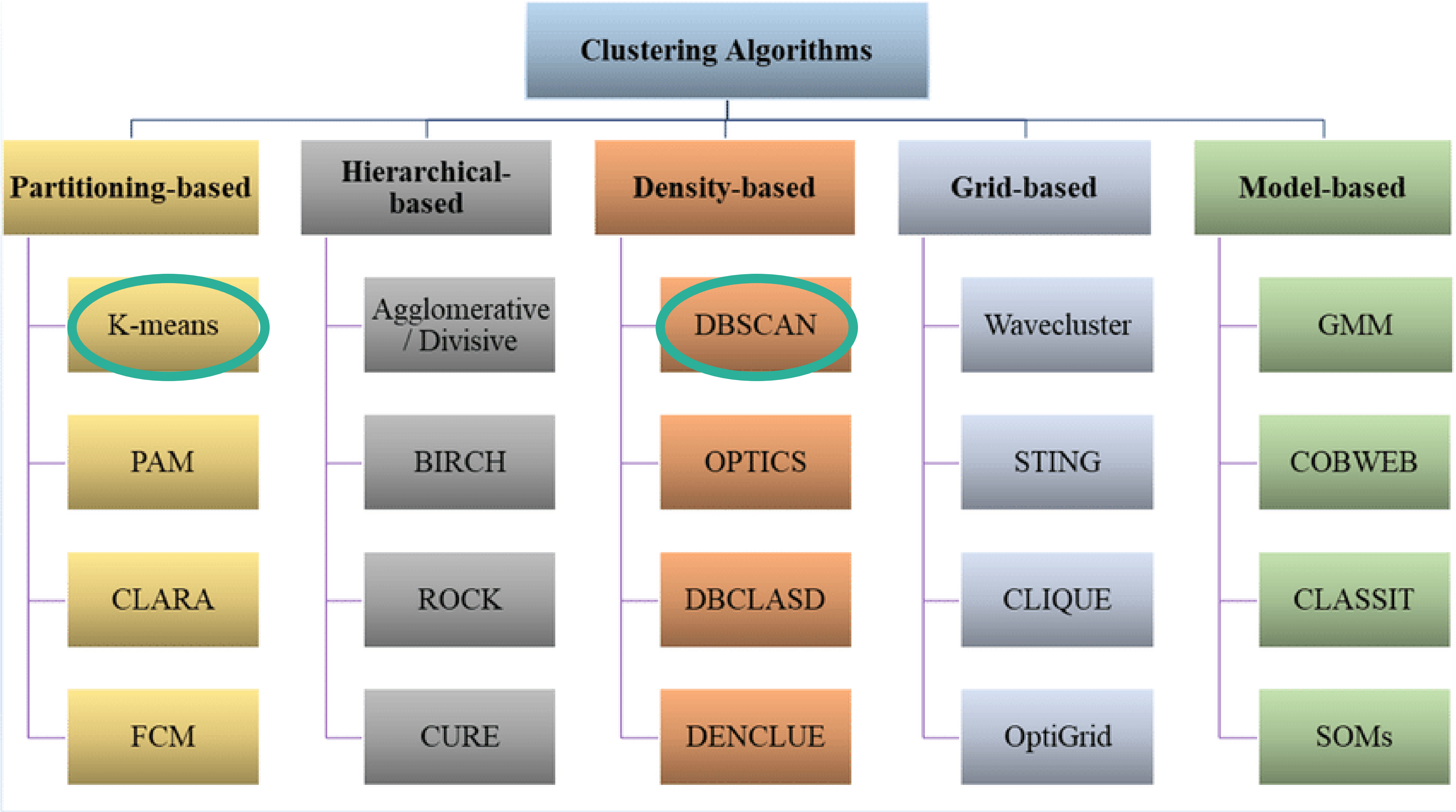


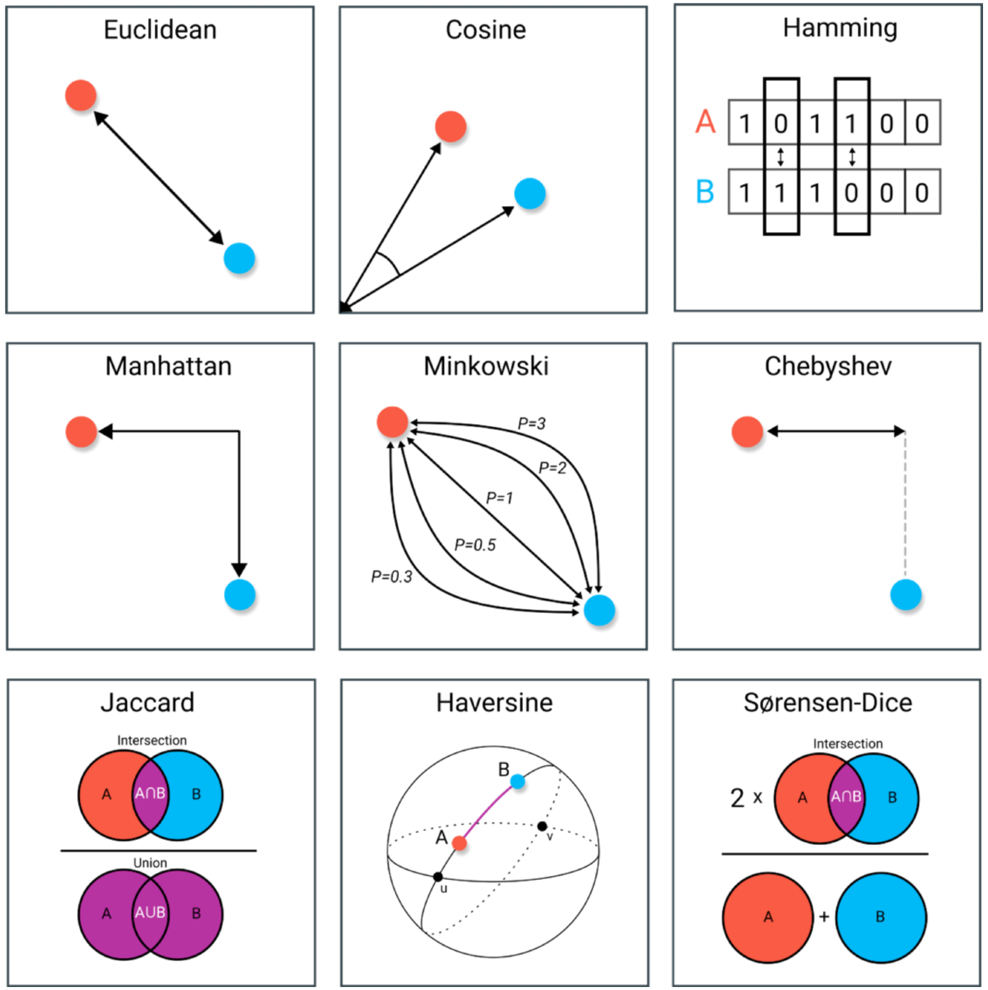
Illustration: Mehta et al, 2020. Link

Distance metrics

Recap

How to quantify similarity / proximity?

- Euclidean $\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
- Manhattan $\sum_{i=1}^{n}|p_i - q_i|$
- Minkowski $\left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{\frac{1}{p}}$
- Cosine
- Hamming
- ...
- and any application specific distances...



Similarity vs. distance

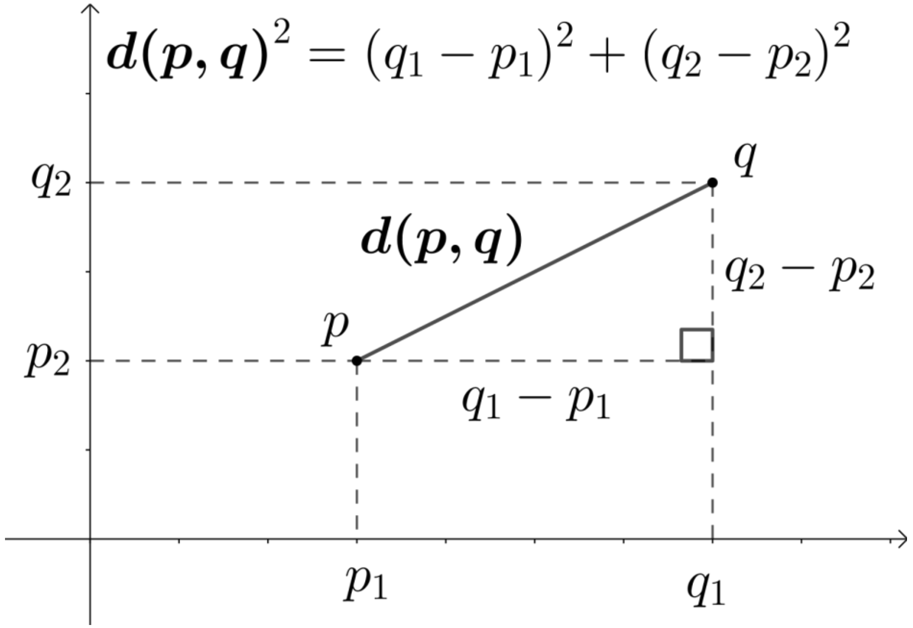
- **Similarity**:
  - Measure how **alike** two objects are.
  - A higher similarity value indicates that the objects share more common features.
  - Values often range between [0, 1], or [-1, 1]

- **Distance** (or dissimilarity):
  - Measure how **different** or far apart two objects are.
  - A higher distance indicates that the objects are more dissimilar or farther apart.
  - Values often range between $[0, \infty]$

- **Conversion**:
  - To convert a similarity metric S to a dissimilarity metric D,
    the following conversion rule is often applied: $D = 1 - S$

Euclidian distance

- Between two points **p=**(p_1, p_2) and **q=**(q_1,q_2)**:**



Between two n-dimensional points**:**

p = (p_1, p_2,..., p_n) **q** = (q_1, q_2,..., q_n)

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1-p_1)^2 + (q_2-p_2)^2 + \cdots + (q_n-p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}.$$

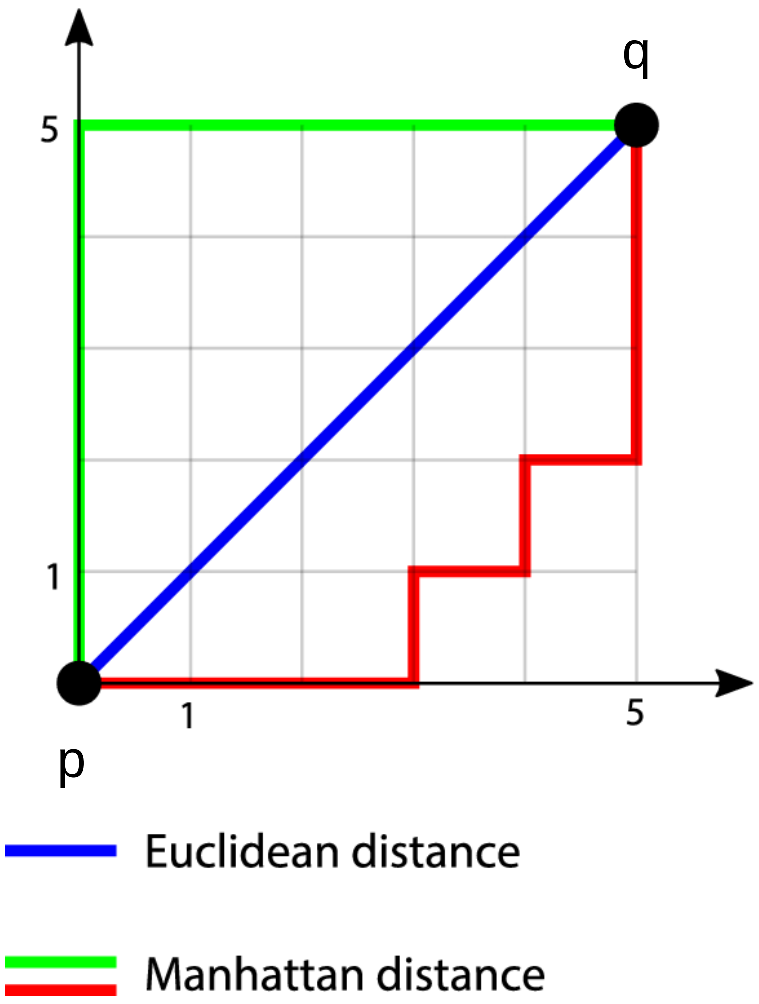
$$\|\mathbf{q}-\mathbf{p}\| = \sqrt{(\mathbf{q}-\mathbf{p})\cdot(\mathbf{q}-\mathbf{p})}.$$

Manhattan distance

- Between two n-dimensional points:

$\mathbf{p} = (p_1, p_2, \dots, p_n) \qquad \mathbf{q} = (q_1, q_2, \dots, q_n)$

$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|,$$

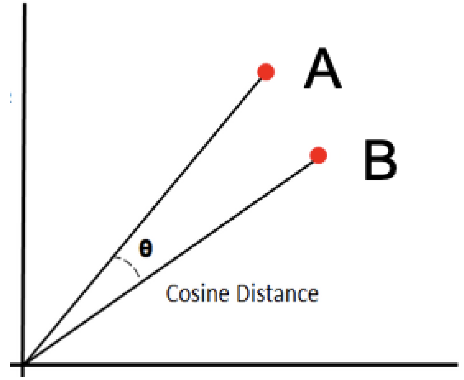


Cosine similarity

- Given two vectors $A = [a_1, a_2, \dots, a_n]$ and $B = [b_1, b_2, \dots, b_n]$ with n attributes each, the cosine similarity, $\cos(\theta)$ is represented as

$$\cos(\theta) = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2} \times \sqrt{\sum_{i=1}^{n} b_i^2}} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

It ONLY considers the angle between the two vectors, but NOT their lengths



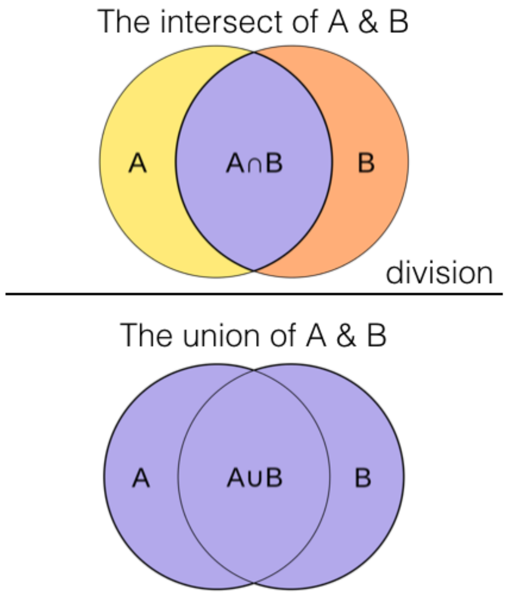
- **Interpretation**:
  - Similar points have a value close to 1
  - Points with no similarity have a value close to 0
  - Samples with an opposite meaning have a value close to -1

- **Note**:
  - To yield a distance / dissimilarity metric: $\text{Cosine Distance} = 1 - \text{Cosine Similarity}$

Jaccard similarity index

- A similarity metric often used for categorical or binary data

J(A,B) =



$$= \frac{|A \cap B|}{|A \cup B|}$$

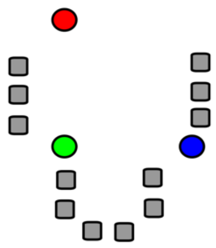
- Interpretation:
  - J=1: The two datasets are identical, meaning they have the same elements.
  - J=0: The two datasets are completely dissimilar, meaning they have no common elements.

k-means clustering

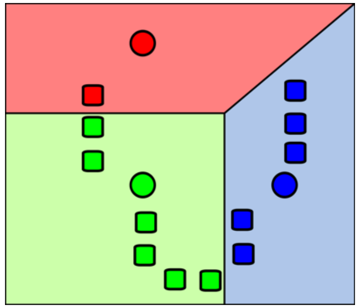
k-means clustering: Overview

- A popular unsupervised learning algorithm
- Used for clustering data into distinct groups or clusters

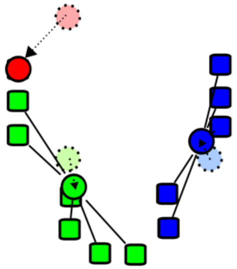
Steps:



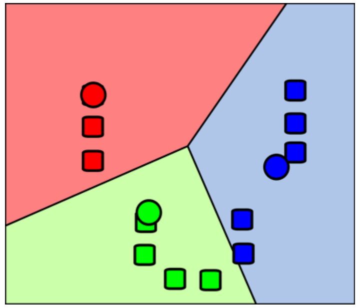
1. K initial "means" (in this case $K = 3$) are randomly generated within the data domain (shown in color)



2. K clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means



3. The centroid of each of the K clusters becomes the new mean



4. Steps 2 and 3 are repeated until a stop-criterion is met

Initialization

1. K initial "means" (in this case $K = 3$) are randomly generated within the data domain (shown in color)

2. K clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means

3. The centroid of each of the K clusters becomes the new mean

4. Steps 2 and 3 are repeated until a stop-criterion is met

Selection of initial centroids: Three alternatives

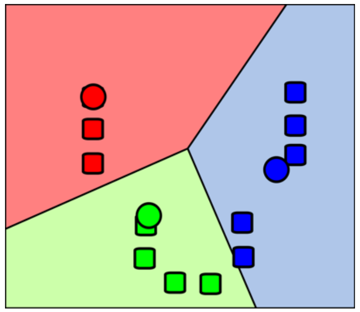
- Random points: K points from $\mathbb{R}^N$
- Forgy method: Randomly chosen K points from the training set
- Random partition: Randomly assign a cluster to each point and go to step 3

Stopping criteria

1. K initial "means" (in this case $K = 3$) are randomly generated within the data domain (shown in color)

2. K clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means

3. The centroid of each of the K clusters becomes the new mean

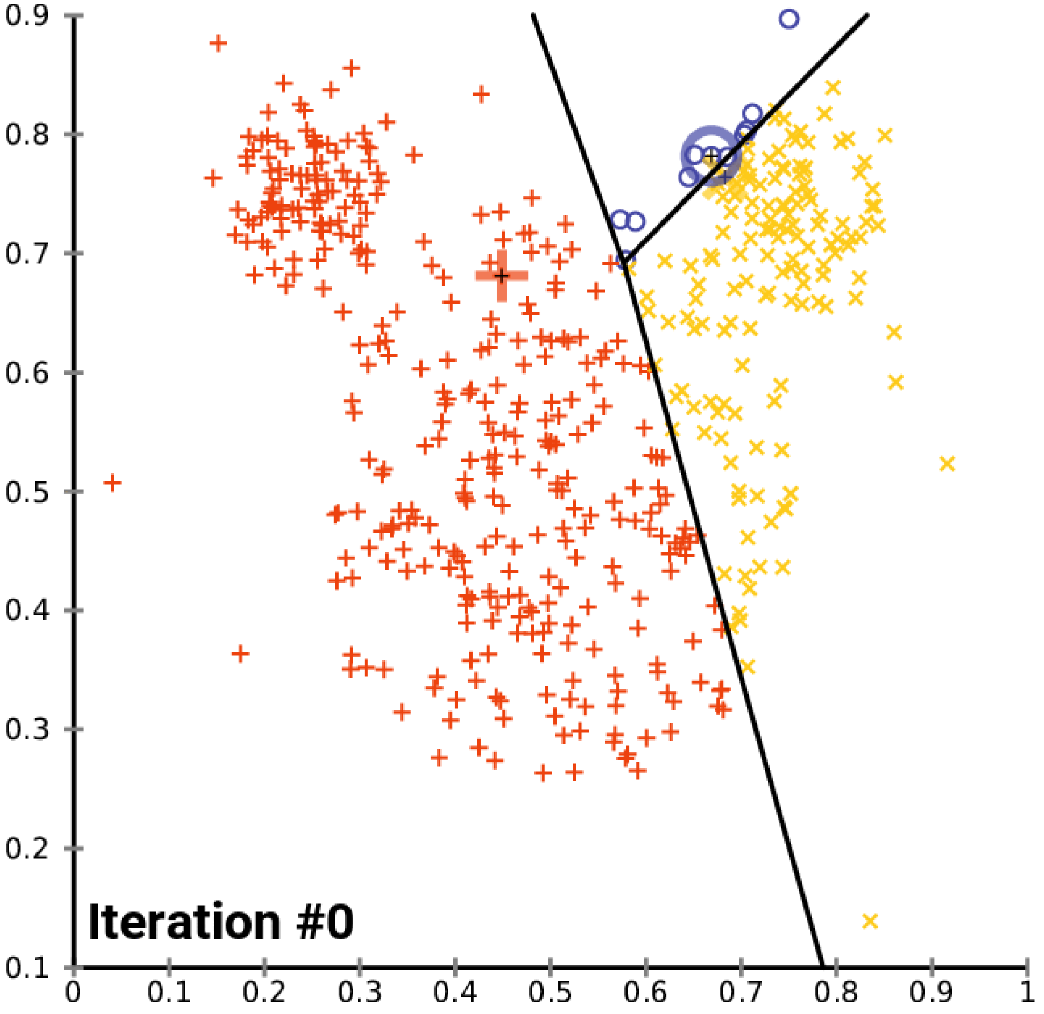


4. Steps 2 and 3 are repeated until a stop-criterion is met

Stopping criteria: (Alternatives can be used simultaneously)

- Stop when the coordinates of the centroids change only very little from one iteration to the next
- Stop when only very few data points are re-assigned to a different centroid in a new iteration
- Stop after a fixed number of iterations (e.g. after 20-50 iterations)
- Stop after a certain time for the entire computation has elapsed (e.g. after 1 minute)

k-means in action

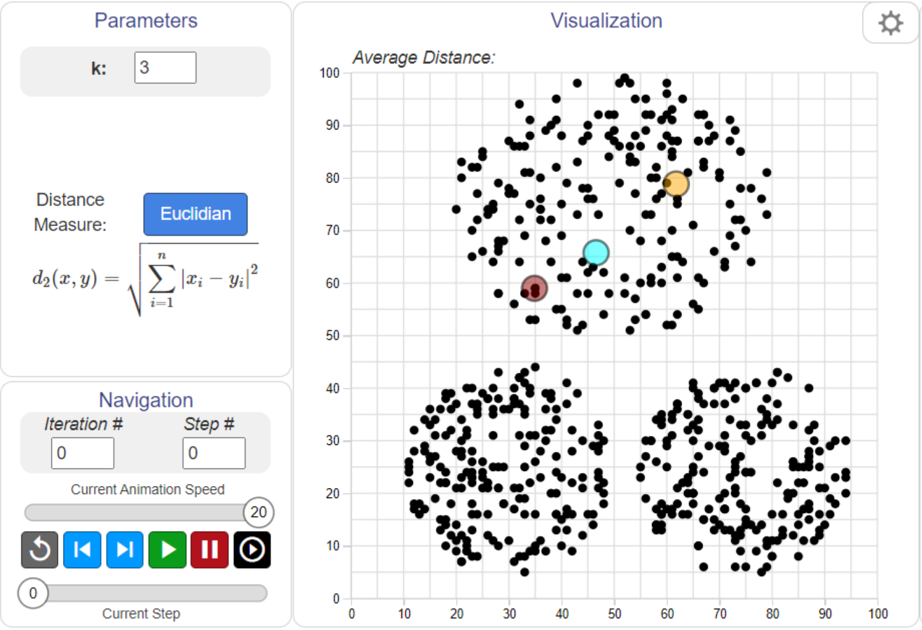


Source: WikiMedia

k-means in action

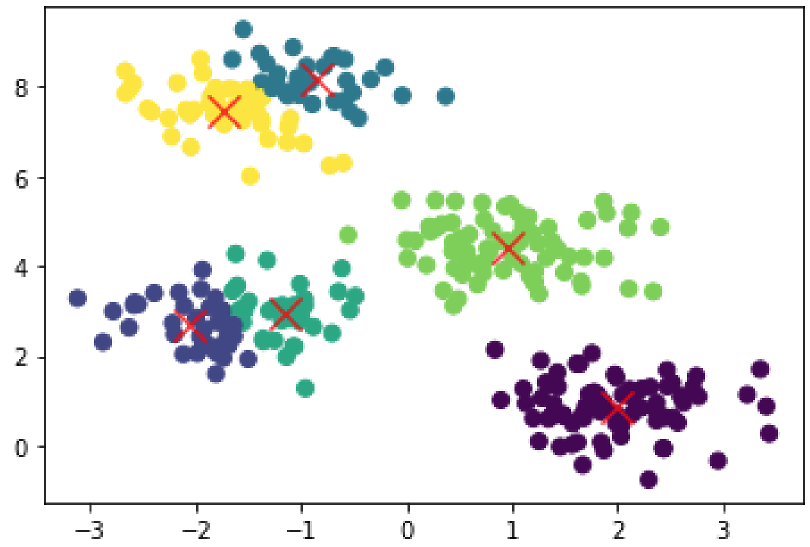
Go to:
https://educlust.dbvis.de/

- Select “k-means” algorithm, “Three Equal Circles” data and k=3:
  - How many iterations does it take
  - Play with initial conditions. Does it always converge to the same solution?
- Select “k-means”, “Outlier”, k=4
  - Does it always converge to the same solution?
  - Play with initial conditions. Can you manage to create a stable solution with a centroid in a small cluster?
- Select “k-means”, “Small Eyebrow”, k=2
  - Explain the solutions and the shortcoming of k-means



Multiple runs of k-means

- Assume you run K-means several times, each time with a different set of initial centroids.
- Will the different runs always stop with the same clustering?



Quality of k-means clustering

- Because of the random initialization, the clusters found by k-means can vary from one run to another and different initial centroids might yield very different clustering results

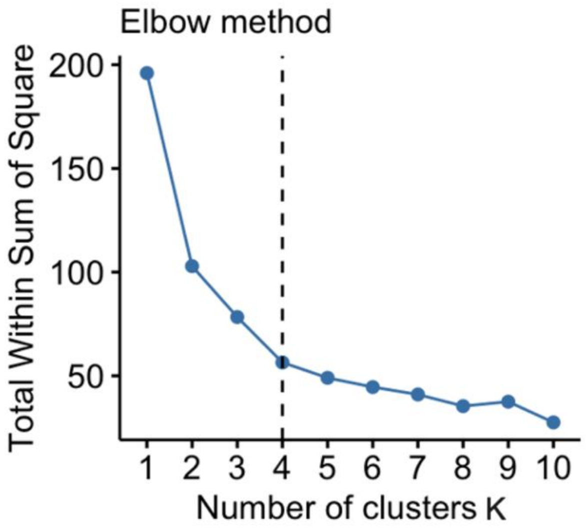
- An optimal value of the hyperparameter k also needs to be selected

- Potential function (within-cluster inertia) measures the squared distance between each data point and its closest centroid and can be used to quantify the clustering quality

$$\Phi(C, X) = \sum_{m=1}^{M} min_{c \in C}(d(x^{(m)}, c)^2)$$

Choosing k with the elbow method

- The elbow method:
  - Run k-means with different values of k and plot the value of the potential function for each k
  - Select the k at the point where the slope of the curve significantly decreases (at the elbow)



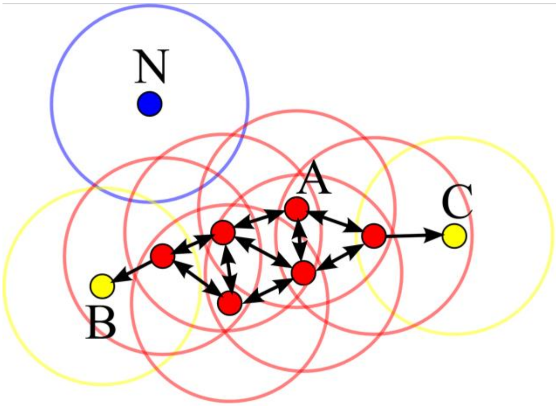
Summary

- Method: Choose k initial centroids, and successively improve positions of these centroids
- Runtime of k-means: O(L*K*N*M)
  - L: number of iterations
  - K: number of clusters
  - N: number of samples
  - M: number of features
- Advanced methods exist:
  - k-Means++ tries to find better initial centroids to reduce the runtime

Density-based clustering: DBSCAN

Definitions

- **minPts**: minimum number of points (a threshold) clustered together for a region to be considered dense. (Will be a hyperparameter of DBSCAN)
- **ε**: distance measure (to define the neighborhood of any point)
- **Point x is reachable from point y** if and only if distance(x,y) < ε
- **Core point**: has at least minPts points within distance ε from itself (e.g. point A)
- **Border point** (B,C): has at least one core point within distance ε (e.g. point B, C)
- **Noise point**: a point that is neither a core nor a border point (e.g. point N)



- **DBSCAN**: Density-based Spatial Clustering of Applications with Noise

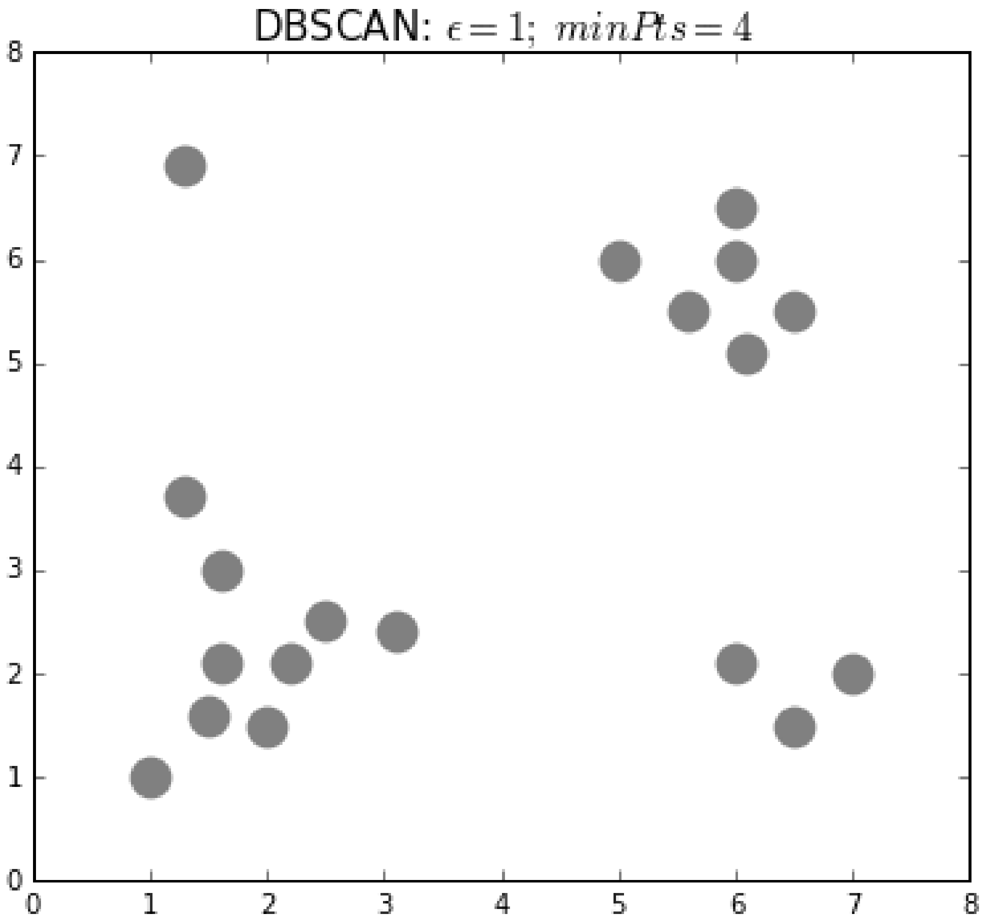
DBSCAN algorithm

1. Select an unprocessed point P
2. If P is not a Core Point (i.e. there are less than *minPts* points within range ϵ), then classify P as noise and go back to Step 1
3. Otherwise, if P is a core point, a new cluster is formed as follows:

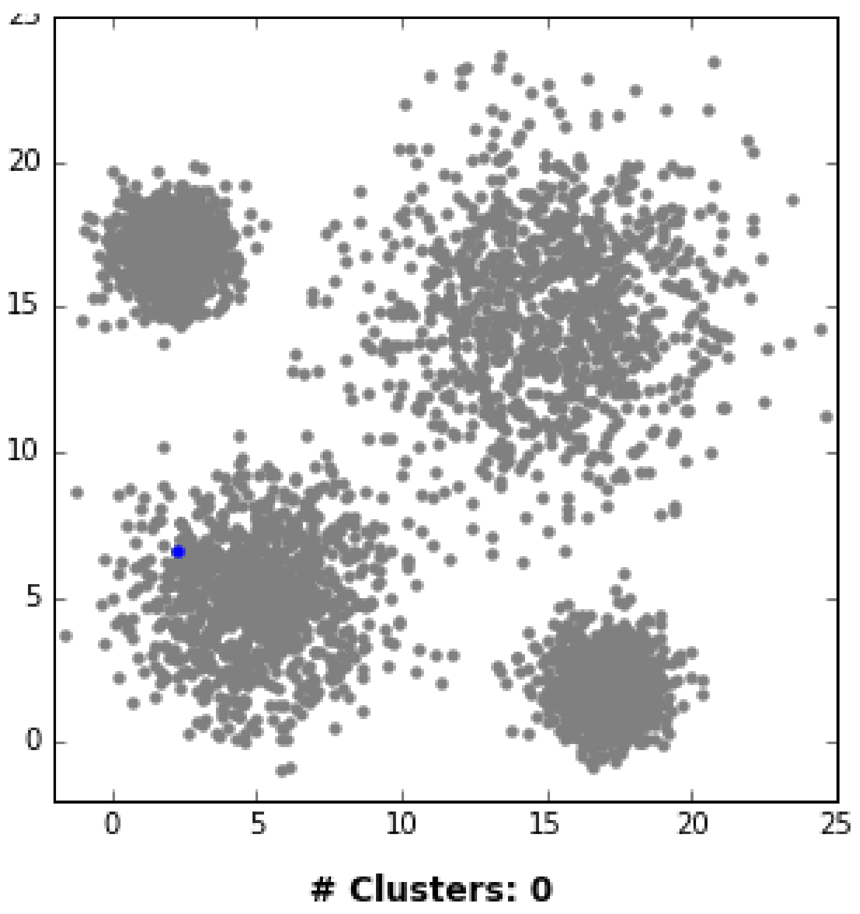
- Assign all neighbors of P (i.e. all points within distance ϵ from P) to the new cluster
- Repeat previous assignment step for all newly-assigned neighbors that are core points

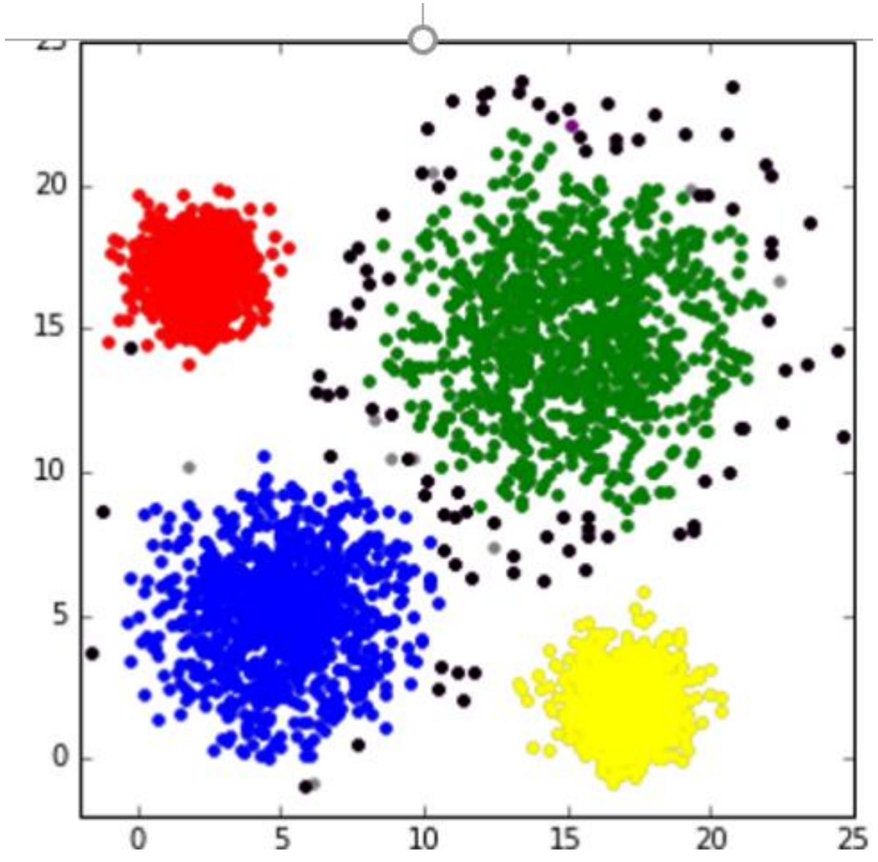
4. Go back to Step 1
5. Continue algorithm until all data points have been processed.

DBSCAN algorithm: Step by step



Example

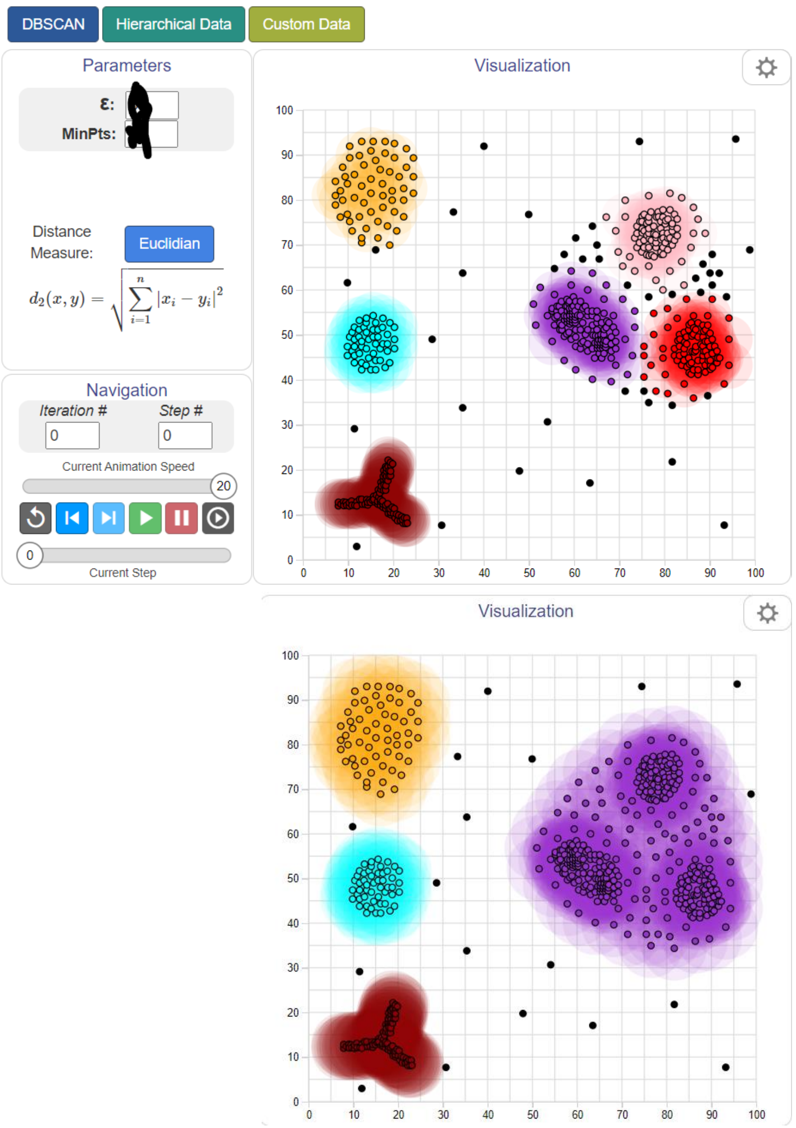




DBSCAN in action

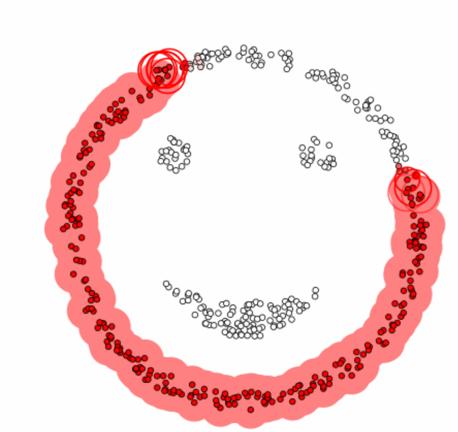
Go to:
https://educlust.dbvis.de/

- Select “DBSCAN”, “Hierarchical Data”
  - Which eps/MinPts do you need to choose to get upper visualization?
  - Which eps/MinPts do you need to choose to get lower visualization?



Properties of DBSCAN

- **Complexity** (running time):
  - $O(N^2)$, with N being the number of samples
  - With more efficient indexing data structure and for non-degenerated data: O(M * log M)



- **Advantages**
  - no need to specify the number of clusters in advance
  - able to find arbitrarily shaped clusters
  - able to detect noise

- **Disadvantages**
  - cannot cluster data sets well with large differences in densities (solved with OPTICS, https://de.wikipedia.org/wiki/OPTICS)

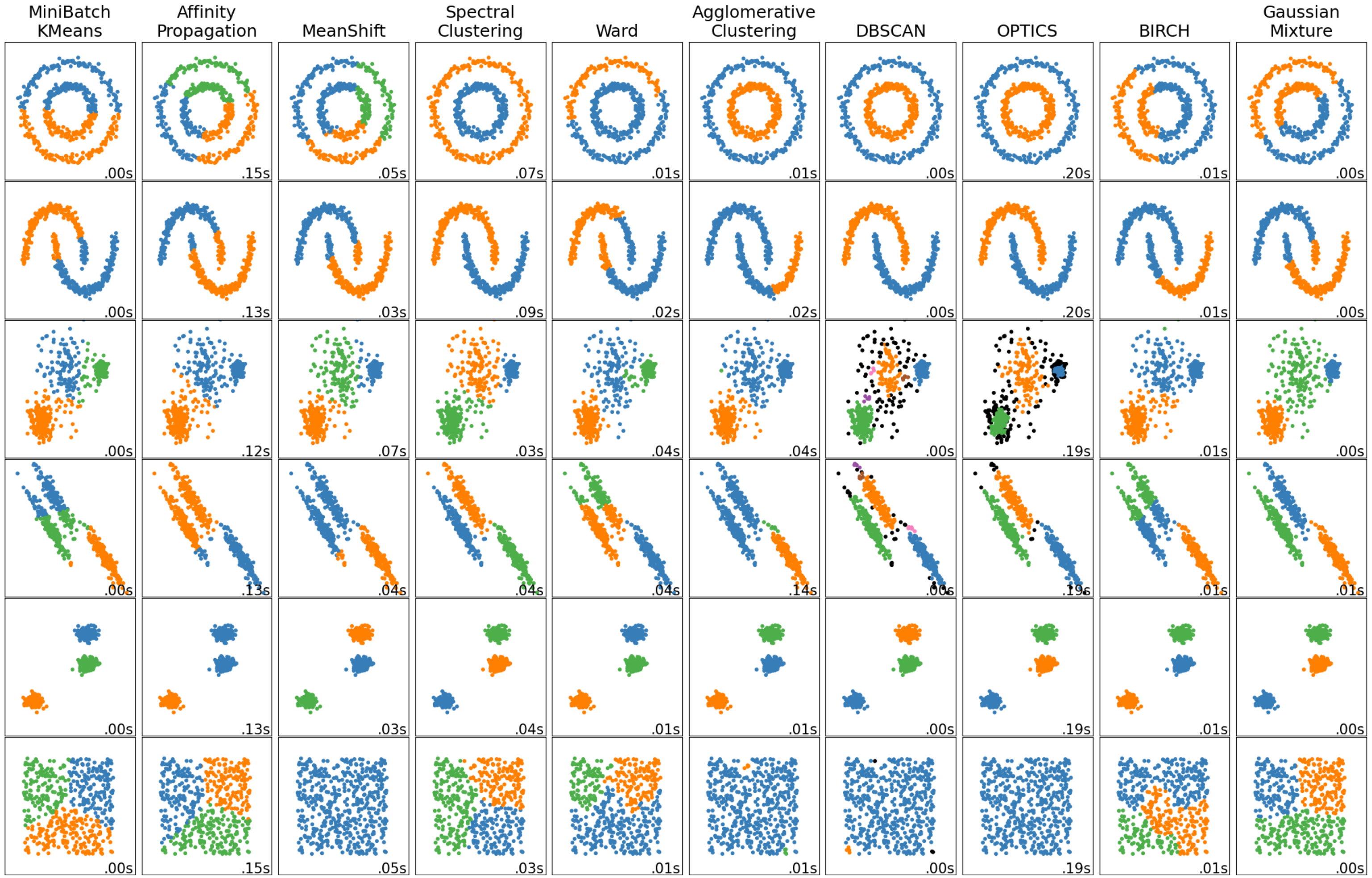
Summary

- Based on density, i.e. closeness of datapoints
- Parameters ε and minPts determine neighborhood of points
- Can handle arbitrary shapes of clusters
- Running time O(M * log M) with smart data structures

Clustering evaluation

Observation

- Different clustering algorithms yield very different clusterings! Which clustering is "the best"?



Evaluation method: Silhouette score

Given K clusters, M data points, and given any data point $x^{(m)}$, let:

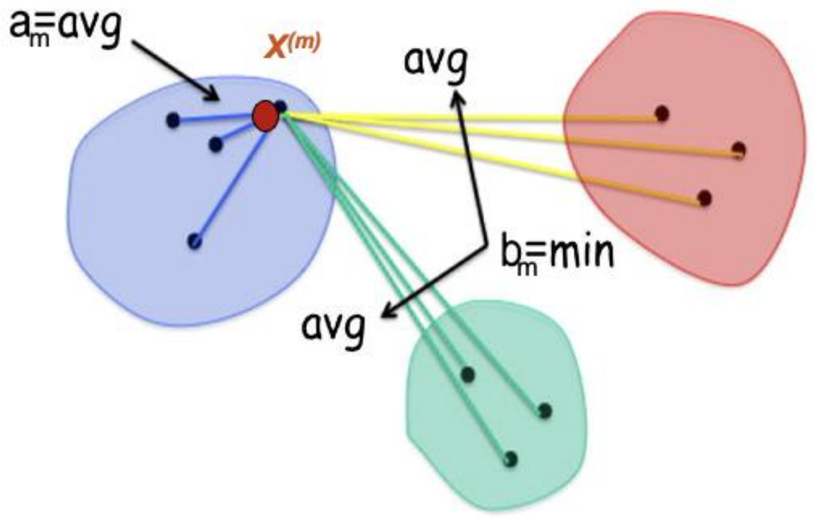
- a_m be the average distance of $x^{(m)}$ with all other points in the *same* cluster. a_m measures how well $x^{(m)}$ fits *into its own* cluster.
- b_m is the smallest average distance of $x^{(m)}$ to all points in any other cluster, $x^{(m)}$ is not a member of.

The **Silhouette Score** $s_m \in [-1,1]$ is then defined as:

$$s_m = \frac{b_m - a_m}{\max(b_m, a_m)}$$

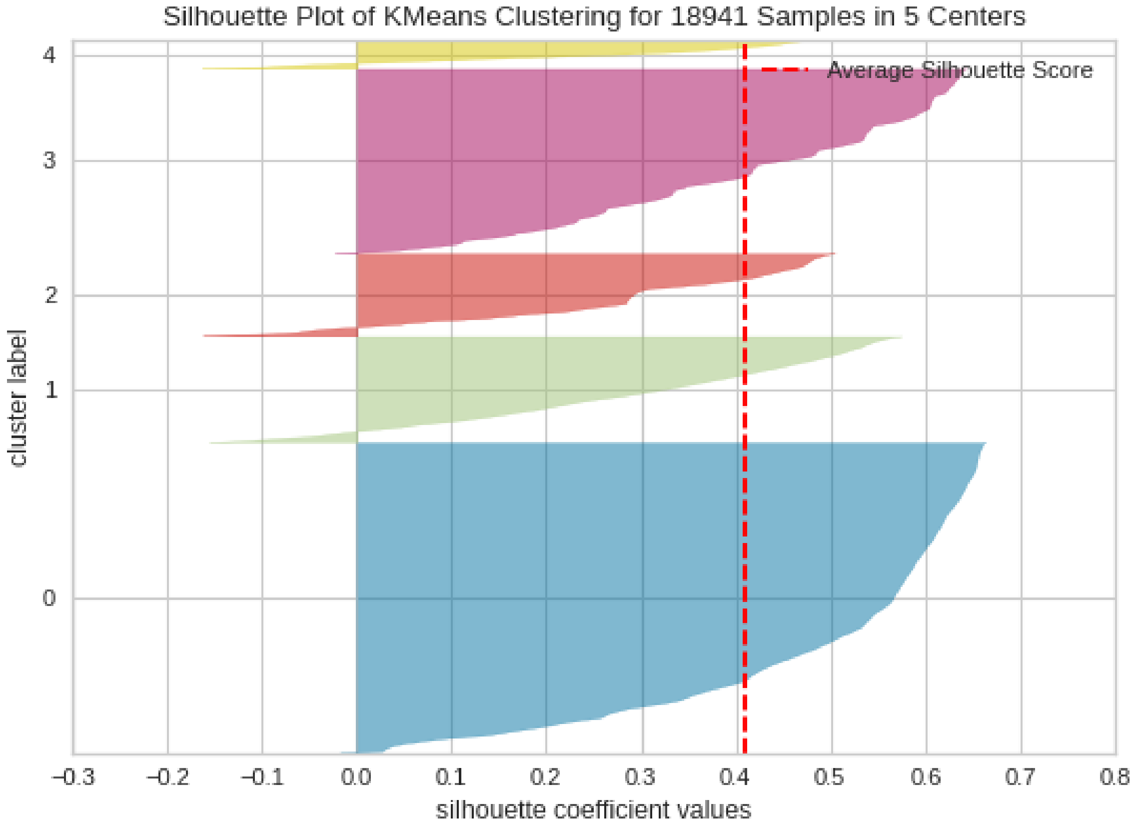
The **Average Silhouette Score** is given by:

$$\frac{1}{M}\sum_{m=1}^{M} s_m$$

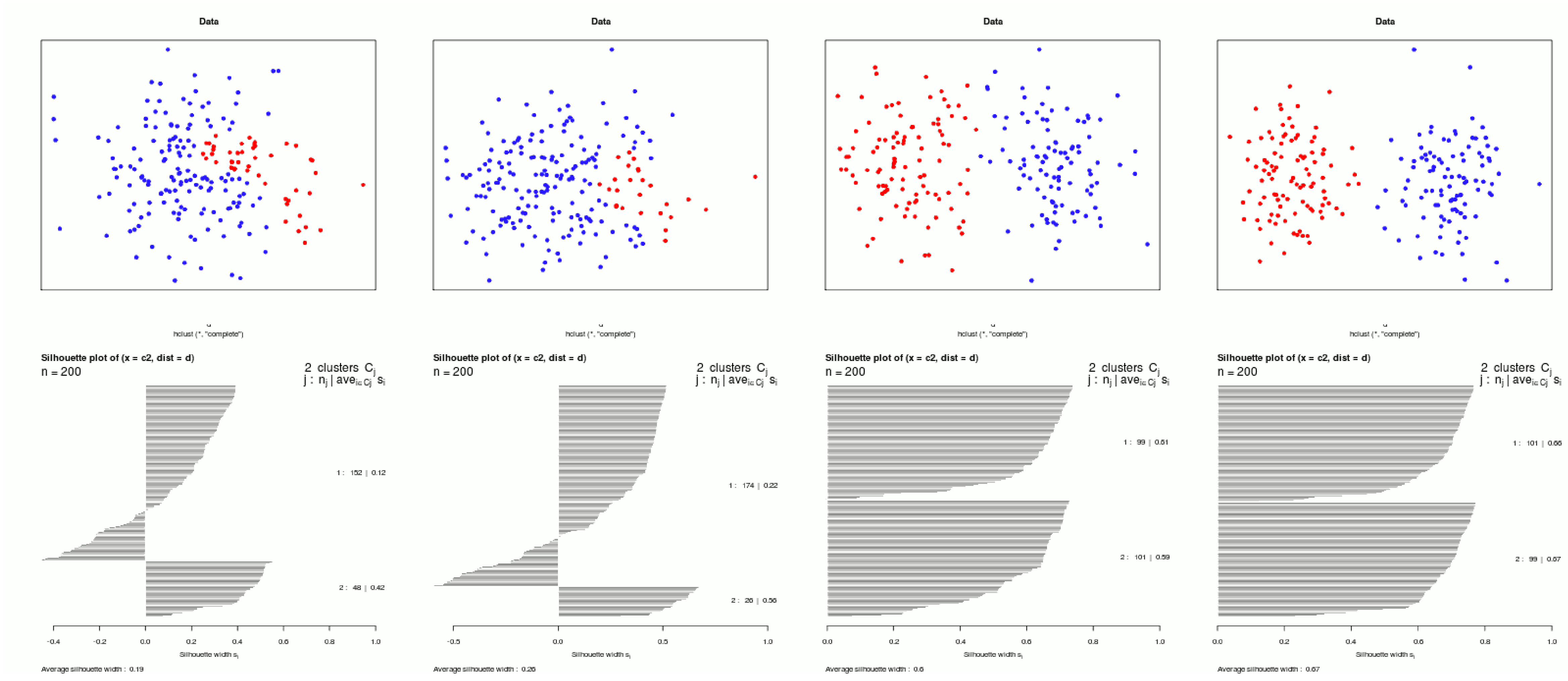


Interpreation of the silhouette score

- Value near +1: the sample is far away from the neighboring clusters
- Value close to 0: the sample is on or very close to the decision boundary between two neighboring clusters
- Negative value: the sample might have been assigned to the wrong cluster



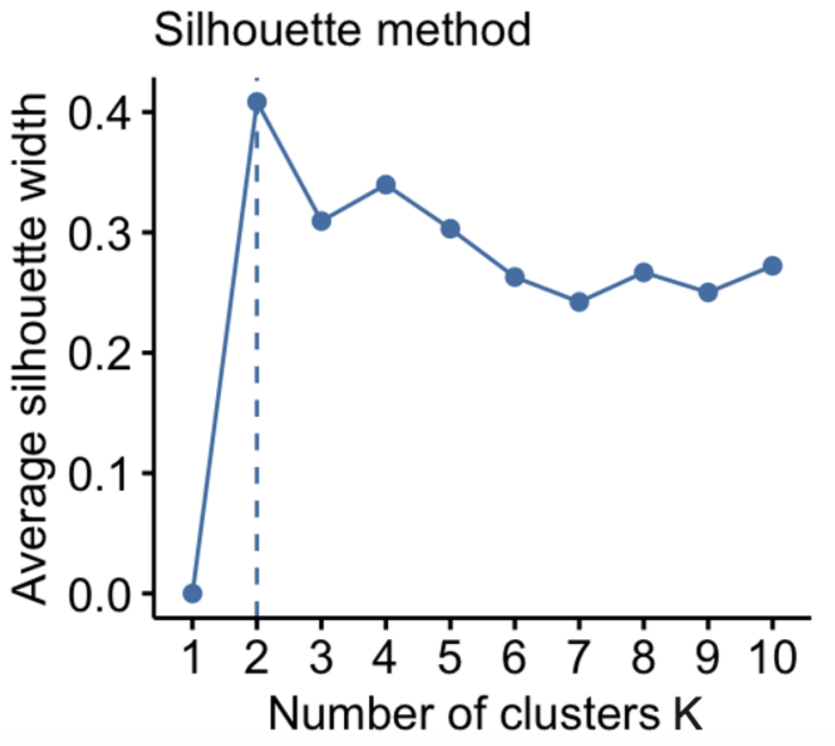
Silhouette score for 4 datasets



Silhouette method for selecting best k

- Compute the average silhouette scores for different values of k (e.g. vary k from 1 to 10 clusters)

- Select the k corresponding to the maximum Average Silhouette Score



Summary

Comparison

Algorithm	K-Means	DBSCAN
Advantages	• relatively easy and efficient to implement • works for very large datasets • computationally faster then other clusterings	• no need to define number of clusters • can discover arbitrarily shaped clusters • robust to outliers and noise
Disadvantages	• sensitive to initialization • number of clusters is hard to choose • unable to handle outliers or noisy data	• not partitionable for multiprocesses • datasets with different densities are tricky • can depend on the order of the data

Further ~~reading~~ watching

- StatQuest: K-means clustering (8 min)
- StatQuest: hierarchical clustering (11 min)
- StatQuest: clustering with DBSCAN (9 min)

- Don’t confuse kNN with k-means clustering!