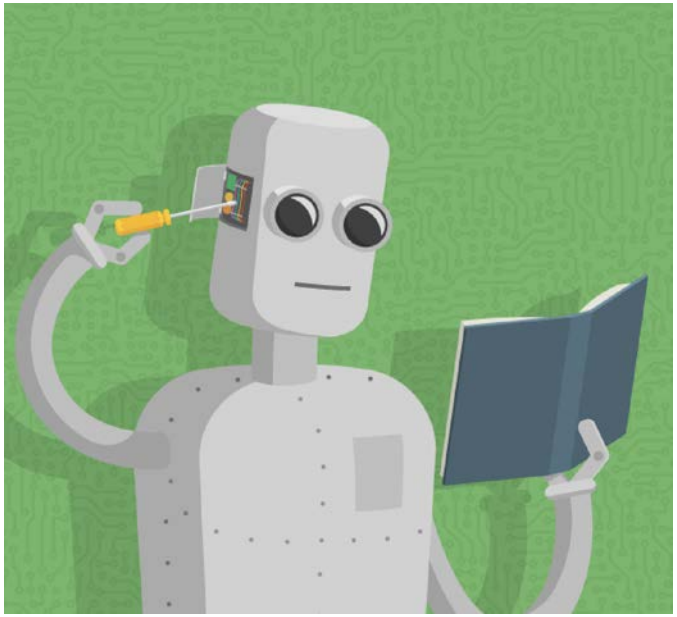
NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

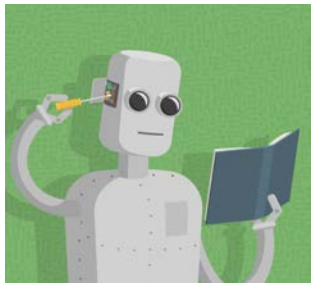


MSE MachLe
Clustering

Christoph Würsch
Institute for Computational Enineering ICE
Interstaatliche Hochschule für Technik Buchs, FHO

Unsupervised Learning
Hierarchical Clustering & K-means

Overview



Labeled Data

Indirect (reward)

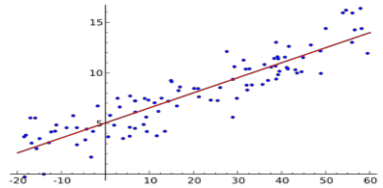
Unlabeled Data

Supervised Learning

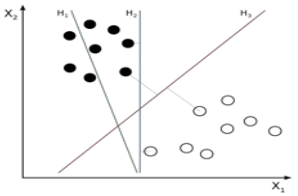
Reinforcement & Bandit Learning

Unsupervised Learning

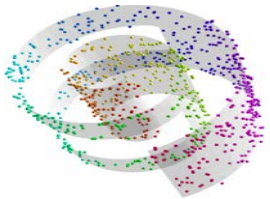
Regression



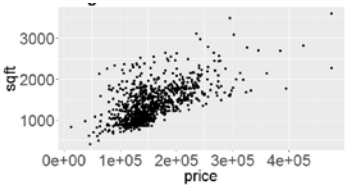
Classification



Dimensionality Reduction



Clustering



Agenda

NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

1. **Hierarchical Clustering (cost based)**
   - Bottom-up: **agglomerative**
     (Linkage algorithms (single, complete, Ward, average, maxoid, medoid,…)
   - Top-Down: **divisive**
     (Single linkage clustering using a minimum spanning tree =MST)
2. **Partitional Clustering:**
   - K-means and its variants
3. **Metrics to evaluate Clustering**
4. **Density based Clustering: DBSCAN**

5. **Model based clustering:**

   Gaussian Mixture Models (GMM)
   - See next lecture (21.5.2019)

What is clustering?

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- **Clustering = Finding groups in data**
- **Problem**: given n data points, separate them into K clusters

  n: number of data points

  K: number of clusters ($K \ll n$)

  Δ: a partition, $\Delta = \{C_1, C_2, \dots, C_K\}$

  $\mathcal{L}(\Delta)$: loss of Δ to be minmized

- **Hard clustering**: each data point is assigned a unique cluster: Δ
- **Soft clustering:** each data point i is assigned a probability that it is in cluster k: $\gamma = \{\gamma_{ki}\}_{k=1:K}$

  γ_{ki}: The degree of membership of data point i to cluster k with $\sum_k \gamma_{ki} = 1$ for all i

  Usually associated with a probabilistic model: cost $\mathcal{L}(\gamma) = -$likelihood

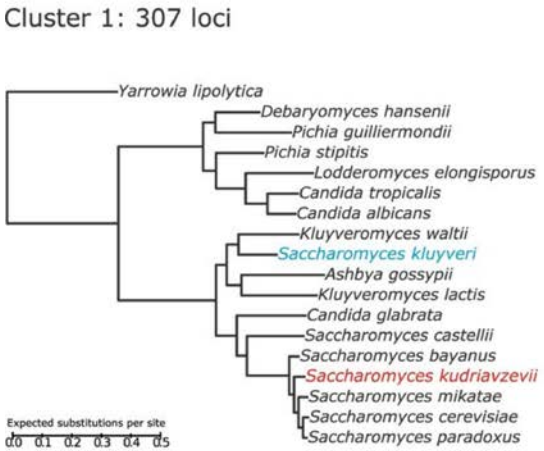
How would you cluster these animals?

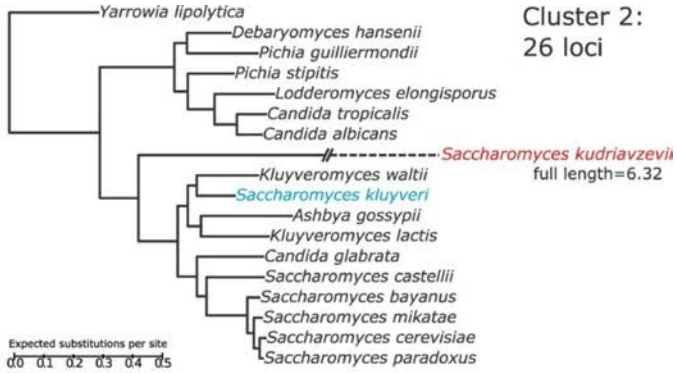
NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

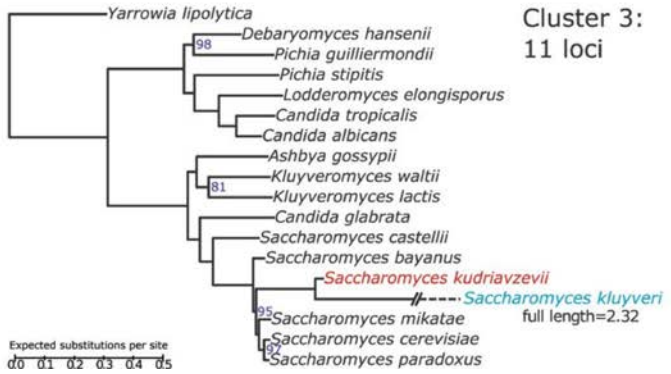
Example:
Clustering Genes of Common Evolutionary History

NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- Phylogenetic trees inferred from the three clusters found in the yeast analysis with `treeCl`







From: Clustering Genes of Common Evolutionary History
Mol Biol Evol. 2016;33(6):1590-1605. doi:10.1093/molbev/msw038
https://academic.oup.com/mbe/article/33/6/1590/2579727
https://github.com/kgori/treeCl
http://etetoolkit.org/

- Clustering is related to vector quantization
  - Dicitionary of vectors (the cluster centers)
  - Each original value represented using a dictionary index
  - Each center claims a nearby region (Voronoi region)
- Example: Image compression (color)
- Example: Text compression: Xerox

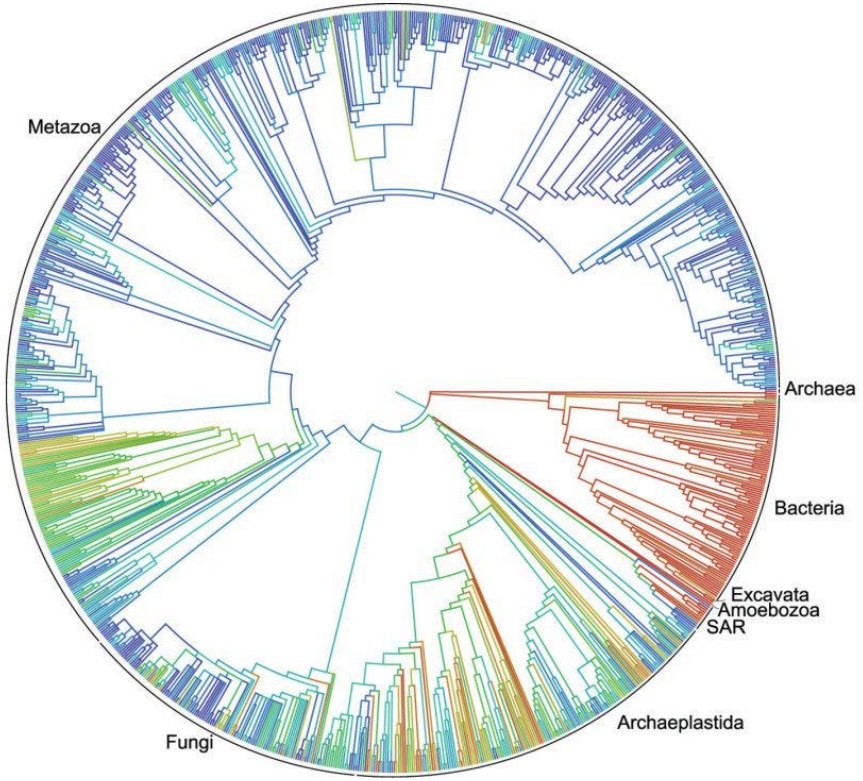
  David Kriesel: Traue keinem Scan, den du nicht selbst gefälscht hast

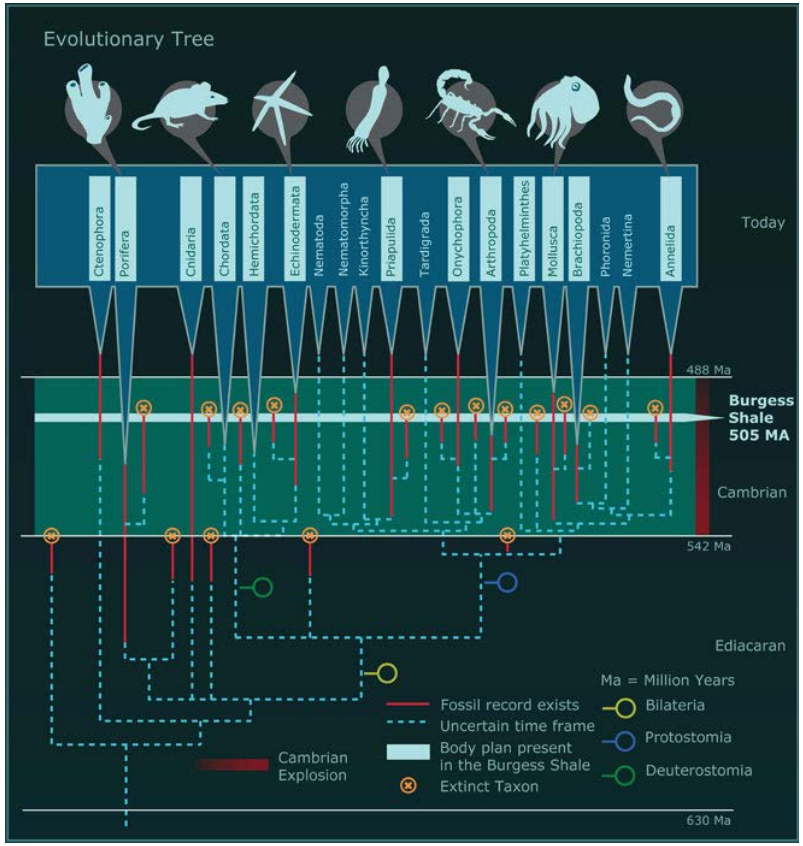
  https://www.youtube.com/watch?v=7FeqF1-Z1g0

Evolution:

NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- Hierarchical clustering





http://www.onezoom.org/life.html

Classification versus Clustering

NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

	Classification	Clustering
Cost $\mathcal{L}(\Delta)$	Expected error	Many! (probabilistic or not)
Type	Supervised	Unsupervised
Generalization	Performance on new data is what matters	Performance on current data is what matters
K	Known	Unknown
Goal	Prediction	Exploration
Stage of field	Mature	Young (growing)

Taxonomy of Clustering I

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

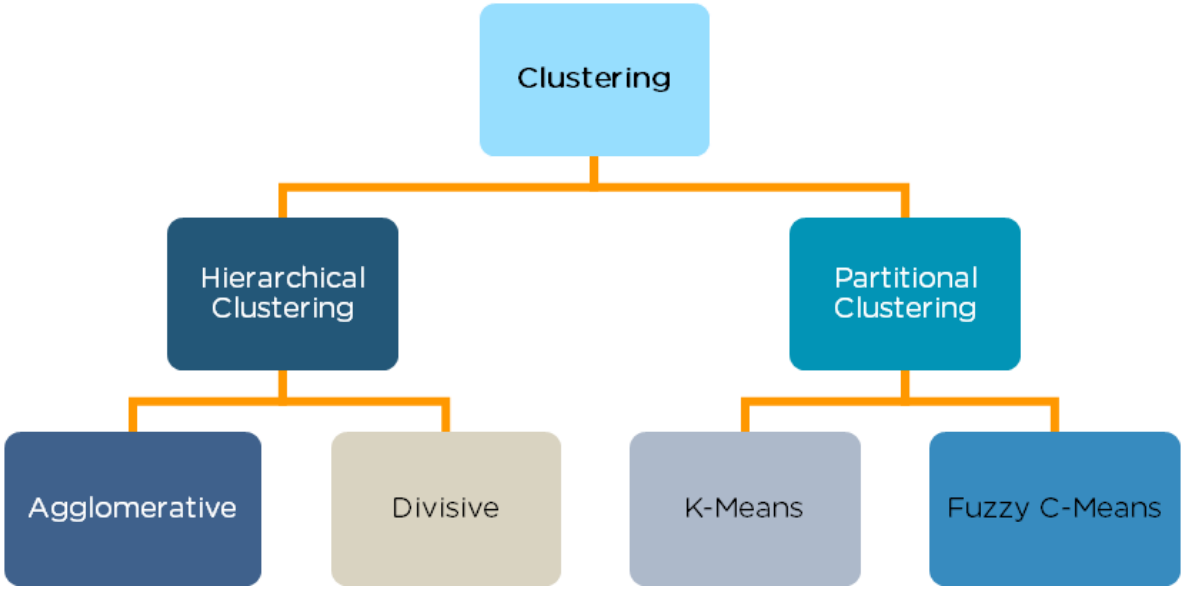
- **Parametric clustering:** K known
- **Non-parmetric:** K determined by algorithm (e.g Dirichlet process, information bottleneck)

Hierarchcal Clustering (HCA) seeks to build a hierarchy of clusters. Strategies for hierarchical clustering generally fall into two types:

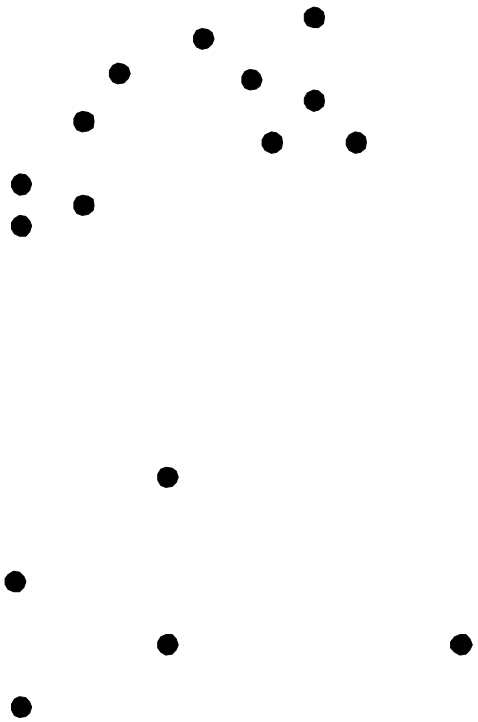
- **Agglomerative**: This is a **bottom-up** approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
- **Divisive**: This is a **top-down** approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

Taxonomy of Clustering II

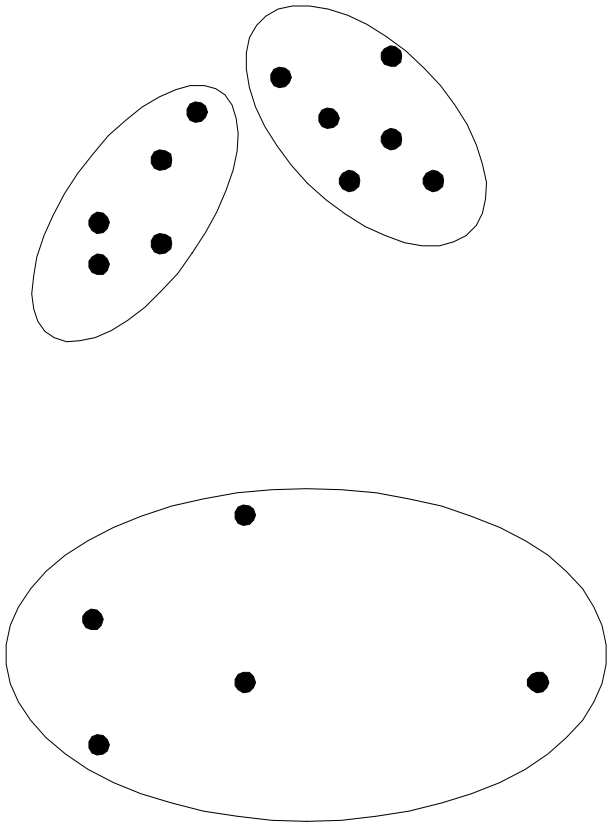
- In **Hierarchical clustering**, clusters have a tree like structure or a *parent child relationship*. Here, the two most similar clusters are combined together and continue to combine until all objects are in the same cluster.
- **K- means** is a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters. It is a division of objects into clusters such that each object is in exactly one cluster, not several.



Partitional Clustering

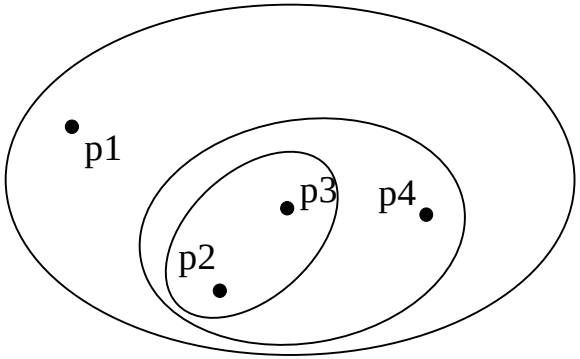


Original Points

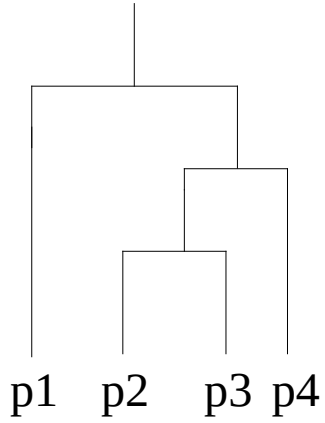


A Partitional Clustering

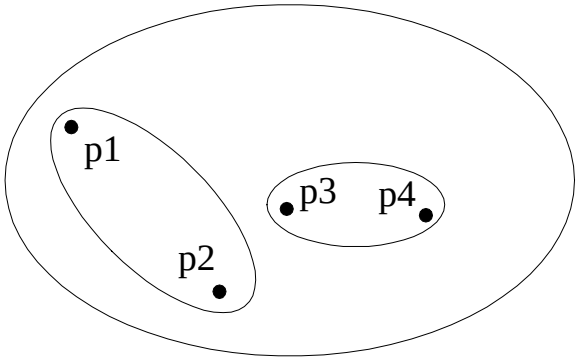
Hierarchical Clustering



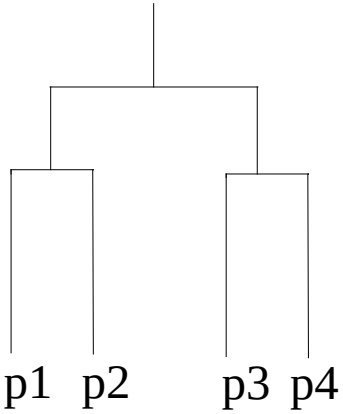
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

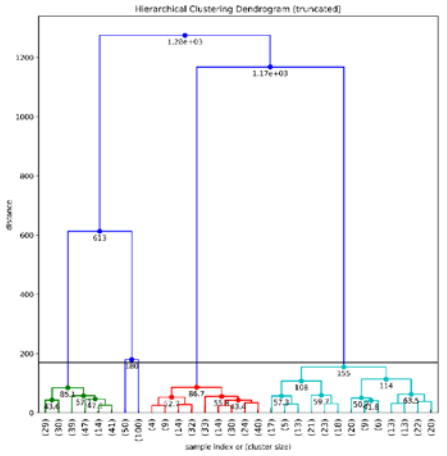
Other Distinctions Between Sets of Clusters

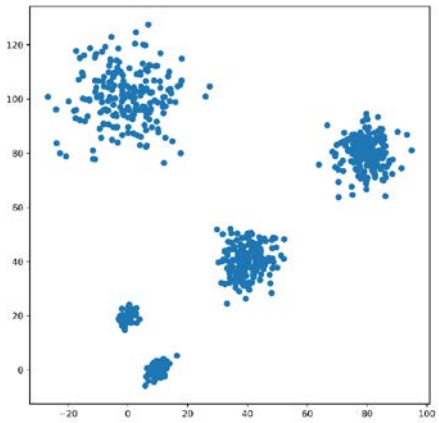
NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- **Exclusive versus non-exclusive**
  - In non-exclusive clusterings, points may belong to multiple clusters.
  - Can represent multiple classes or ‘border’ points
- **Fuzzy versus non-fuzzy**
  - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
  - Weights must sum to 1
  - Probabilistic clustering has similar characteristics
- **Partial versus complete**
  - In some cases, we only want to cluster some of the data
- **Heterogeneous versus homogeneous**
  - Clusters of widely different sizes, shapes, and densities

1. Hierarchical Clustering

- Similarity and metrices
- Linkage criteria
- Basic agglomerative linkage algorithm
- Example of a divisive linkage algorithm (single linkage MST)





1.1 Hierarchical Clustering

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- In order to decide which clusters should be combined (for **agglomerative**), or where a cluster should be split (for **divisive**), a measure of dissimilarity between sets of observations is required.
- In most methods of hierarchical clustering, this is achieved by use of an **appropriate metric** (a measure of distance between pairs of observations), and a **linkage criterion** which specifies the dissimilarity of sets as a function of the pairwise distances of observations in the sets.
- We start with N datapoints that initially form N clusters. **The two clusters with the smallest linkage are fused together** to form N-1 clusters. This is repeated until there is only one single cluster.

1.2 Similarity: defined by a metric

NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

A **metric** $d(x, y)$ is a generalized distance measure that follows the following axioms

1. Non-negativity: $d(x, y) \geq 0$
2. Coincidence: $d(x, y) = 0 \Leftrightarrow x = y$
3. Symmetry: $d(x, y) = d(y, x)$
4. Triangle inequality: $d(x, y) + d(y, z) \geq d(x, z)$

- L_p metric: $d_p(x, y) = \|x - y\|_p = \sqrt[p]{\sum_i |x_i - y_i|^p}$
- L_∞ metric: $\|x\|_\infty = \max_i \{|x_i|\}$
- L_1 metric: $d_1(x, y) = \|x - y\|_1 = \sum_i |x_i - y_i|$
  (taxicab distance, Manhattan distance)

1.4 Linkage Criteria = Fusion Criteria

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

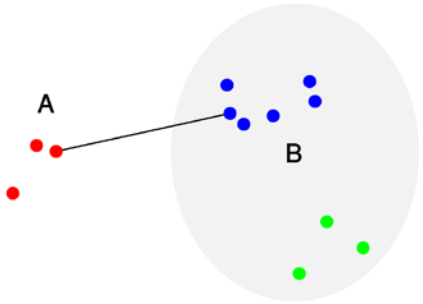
- The linkage criterion determines **together with a metric $d(x, y)$ when two clusters A and B should be merged together in hierarchical clustering (fusion criterium)**.

Names	Formula
Maximum or complete-linkage clustering	$\max\{\, d(a,b) : a \in A,\, b \in B \,\}.$
Minimum or single-linkage clustering	$\min\{\, d(a,b) : a \in A,\, b \in B \,\}.$
Mean or average linkage clustering, or UPGMA	$\frac{1}{\lvert A\rvert . \lvert B\rvert} \sum_{a \in A} \sum_{b \in B} d(a,b).$
Centroid linkage clustering, or UPGMC	$\lVert c_s - c_t \rVert$ where c_s and c_t are the centroids of clusters *s* and *t*, respectively.

1.4 Linkage Criteria

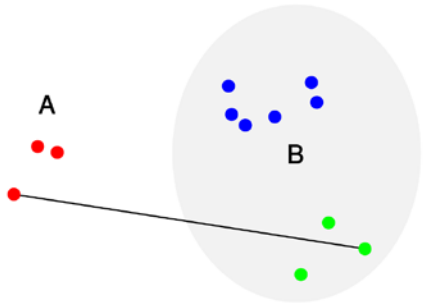
- **Single-Linkage**

$$D_{\mathrm{s}}(A, B) := \min_{a \in A, b \in B} \{d(a, b)\}$$



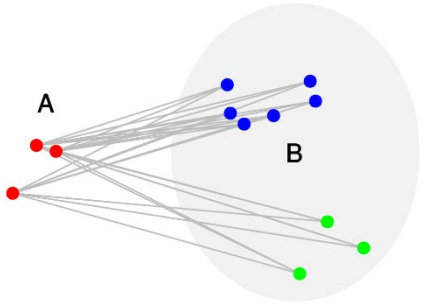
- **Complete Linkage**

$$D_{\mathrm{c}}(A, B) := \max_{a \in A, b \in B} \{d(a, b)\}$$



- **Average-Linkage**

$$D_{\mathrm{avg}}(A, B) := \frac{1}{|A||B|} \sum_{a \in A, b \in B} d(a, b)$$



1.4 Linkage Criteria

NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

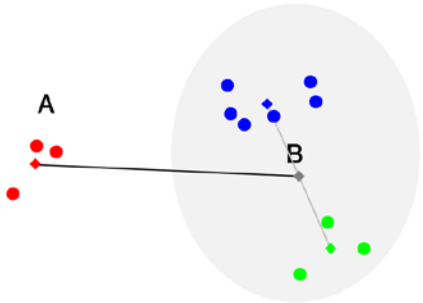
Ward Linkage

- increase of variance when fusing A and B

$$D_{\text{Ward}}(A, B) := \frac{d(\bar{a}, \bar{b})^2}{1/|A| + 1/|B|}$$

Centroid-Linkage

$$D_{\text{Centroid}}(A, B) := d(\bar{a}, \bar{b})$$



Example: Given a distance matrix d_{ij}

d_{ij}	c_1 o_1	c_2 o_2	c_3 o_3	c_4 o_4
o_1	0			
o_2	4	0		
o_3	7	5	0	
o_4	8	10	9	0

→

d_{ij}	c_1 o_1, o_2	c_2' o_3	c_3' o_4
o_{12}	0		
o_3	7 | 5	0	
o_4	8 | 10	9	0

Single-Linkage

d_{ij}	c_1	c_2	c_3
o_{12}	0		
o_3	5	0	
o_4	8	9	0

Complete-Linkage

d_{ij}	c_1	c_2	c_3
o_{12}	0		
o_3	7	0	
o_4	10	9	0

Average-Linkage

d_{ij}	c_1	c_2	c_3
o_{12}	0		
o_3	6	0	
o_4	9	9	0

Lance-Williams formula: allows to calculate fusioning based on distance matrix only.
https://arxiv.org/abs/cs/0608049v2

1.6 Basic form of an agglomerative linkage algorithm (bottom-up)

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

Input:

- Distance matrix D between data points (size $n \times n$)
- function `dist` to compute a distance between clusters (usually takes D as input)

Initialization: Clustering $\mathcal{C}^{(0)} = \{C_1^{(0)}, C_2^{(0)}, \ldots, C_n^{(0)}\} = \{i\}$

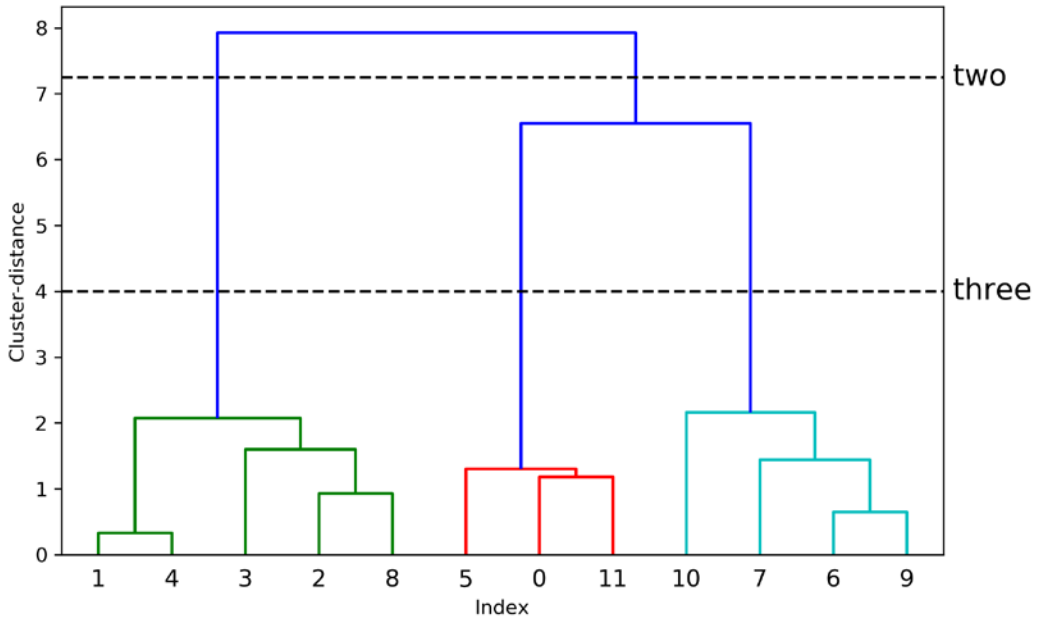
  - Each data point is its own cluster at the beginning.

- **While** the current number of clusters is > 1:
  - find the two clusters which have the **smallest distance (linkage)** to each other
  - **merge** them to one cluster
- **Output**: Resulting **dendrogram:** The dendrogram is a tree that represents the hierarchical division of the data set O into ever smaller subsets.

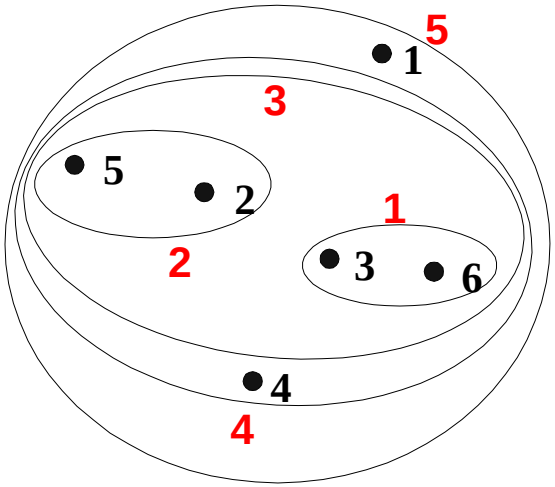
1.7 Hierarchy → Dendrogram

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

```
# Import dendrogram and ward clustering from SciPy
from scipy.cluster.hierarchy import dendrogram, ward
X, y = make_blobs(random_state=0, n_samples=12)
#Perform ward clustering on the data in array X. The function ward in SciPy
returns an array with the #distances bridged in agglomerative clustering
linkage_array = ward(X)
# draw a dendrogram
dendrogram(linkage_array)
```

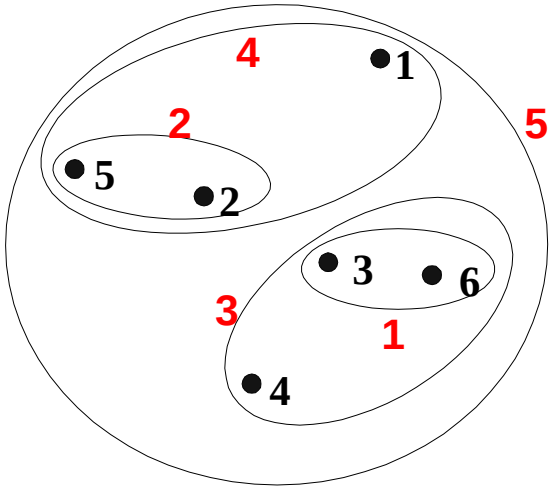


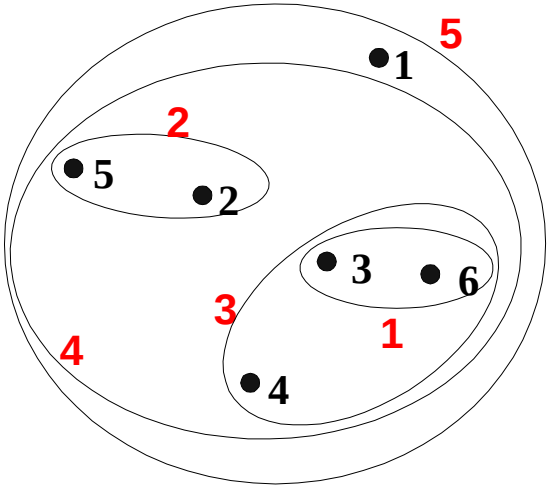
Hierarchical Clustering: Comparison



MIN

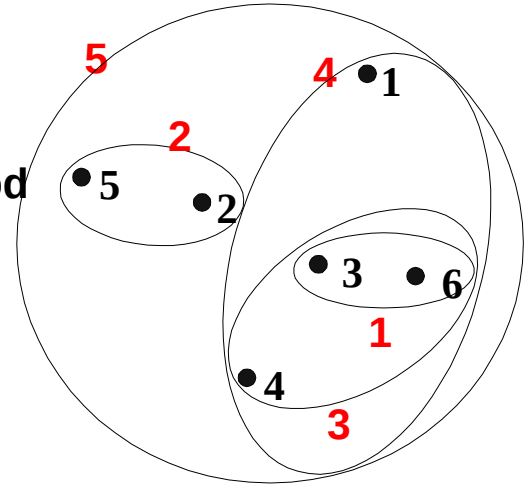
MAX





Group Average

Ward's Method



1.8 Comments

NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

`sklearn.cluster.AgglomerativeClustering`

- **Single linkage** tends to generate long "**chains**"
- **Complete linkage** tends to produce more "**compact**" clusters
- Linkage algorithms are very **vulnerable to outliers**
- one cannot "undo" a bad link
- Single linkage can also be described using the **minimal spanning tree** of data points (e.g., cutting the longest edge of an MST gives the first two single linkage clusters)
- Advantage of hierarchical clustering: do not need to decide on "the right" number of clusters
- There exist many more ways of generating different trees from a given distance matrix.

1.9 Divisive Clustering: Single Linkage Algorithm based on MST

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- **Input**: Data $\mathcal{D} = \{x_i\}_{i=1:N}$, number of clusters K

1. Construct the Minimum Spanning Tree (**MST**) of $\mathcal{D}$
2. Delete the largest $(K-1)$ edges

- **Cost**: $\mathcal{L}(\Delta) = \min_{k,k'} d(C_k, C_{k'})$

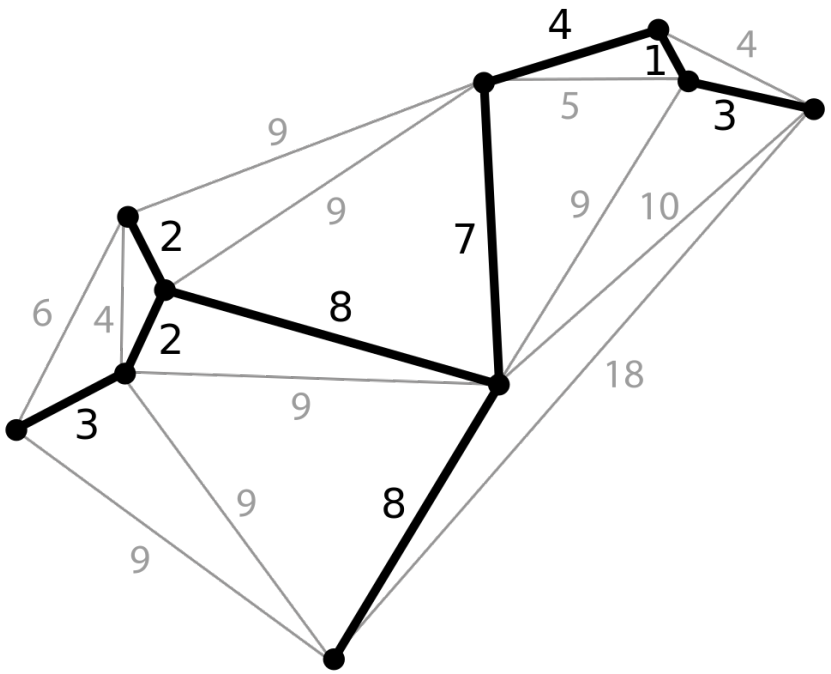
  $d(A,B) = \min_{x \in A, y \in B} \|x - y\|$

- Sensitive to outliers
- Cost can be evaluated in polynomial time $O(n^2)$

1.10 Minimum Spanning Tree (MST)

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- A minimum spanning tree (MST) is a subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together, without any cycles and **with the minimum possible total edge weight**. That is, it is a spanning tree whose sum of edge weights is as small as possible.



A planar graph and its minimum spanning tree. Each edge is labeled with its weight, which here is roughly proportional to its length.

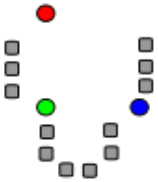
2. K-means

- 2.1 K-means clustering in a **nutshell**
- 2.2 K-means **algorithm** (peseudocode)
- 2.3 K-means **cost function**
- 2.4. K-Means Algorithm for a given K: **Details**
- 2.5: Picking the Initial Centers (**initialization**)
- 2.6 **Implementation** in `scikit-learn`
- 2.7 More **variants** of K-means
- 2.8 **K-medoid** | **K-maxoid** clustering
- 2.9 **Heuristics** for improving the result
- 2.10 How do we **choose K**?

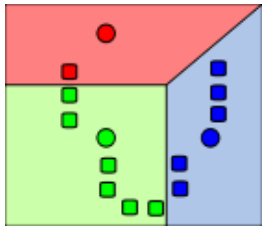
2.1. K-means clustering in a nutshell

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

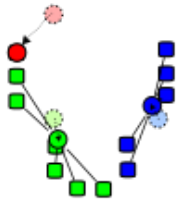
The standard algorithm: non-probabilistic EM



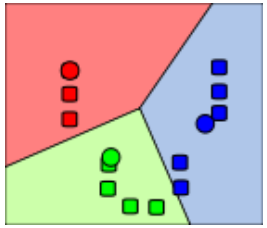
1. k initial "means" (in this case $k = 3$) are randomly generated within the data domain (shown in color).



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.

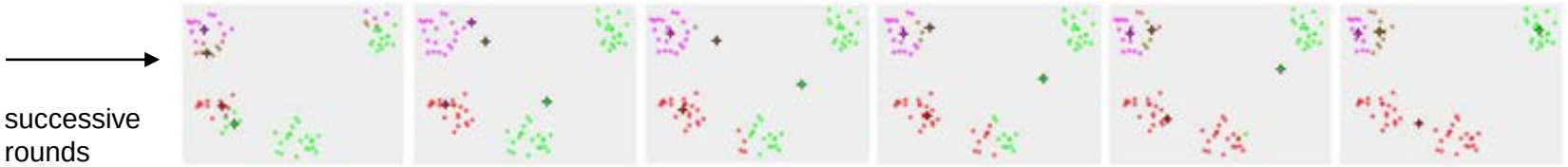


4. Steps 2 and 3 are repeated until convergence has been reached.

...

Properties

- Problems: Very sensitive to choice of k; even with correct k it may converge to wrong local minimum



- Variants: k-medoids (centroid to be member of data set), k-maxoids (for extremes rather than means)

Source: https://en.wikipedia.org/wiki/K-means_clustering

2.2. K-means algorithm (peseudocode)

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- **Input**: Data $\mathcal{D} = \{x_i\}_{i=1:N}$, number of clusters K
- **Initialize**: centers $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^d$ at random
- Iterate until convergence:

1. for $i = 1{:}n$

   $k(i) = \text{argmin}_k \|x_i - \mu_k\|$

   (assign points to cluster → new clustering)

2. for $k = 1{:}K$

   $\mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i$ (recalculate centers)

- **Convergence:** if Δ does not change after iteration m, it will never change after that.

2.3. K-means: cost function

- «least-squares» cost, also called **distortion (within cluster inertia W)**

$$\mathcal{L}(\Delta) = \sum_{i=1}^{n} \|x_i - \mu_{k(i)}\|^2$$
$$= \sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - \mu_{k(i)}\|^2$$

- The distortion can also be expressed as **sum of (squared) intracluster distances**

$$\mathcal{L}(\Delta) = \frac{1}{2} \sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - x_j\|^2 + \text{const}$$

2.4. K-Means Algorithm for a given K: Details

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

```
centers ← pick k initial Centers

while (centers are changing) {
   // Compute the assignments
   asg ← [(x, nearest(centers, x)) for x in data]
```

What do we mean by “nearest”?
A: Squared Euclidean distance

2.4. K-Means Algorithm: Details

NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

```
centers ← pick k initial Centers

while (centers are changing) {
   // Compute the assignments
   asg ← [(x, nearest(centers, x)) for x in data]

   // Compute the new centers
   for j in range(K):
      centers[j] =
         mean([x for (x, c) in asg if c == j])
}
```

2.4. K-Means Algorithm: Details

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

```
centers ← pick k initial Centers

while (centers are changing) {
   // Compute the assignments
   asg ← [(x, nearest(centers, x)) for x in data]

   // Compute the new centers
   for j in range(k):
      centers[j] =
         mean([x for (x, c) in asg if c == j])
}
```

Guaranteed to converge!

… to what?

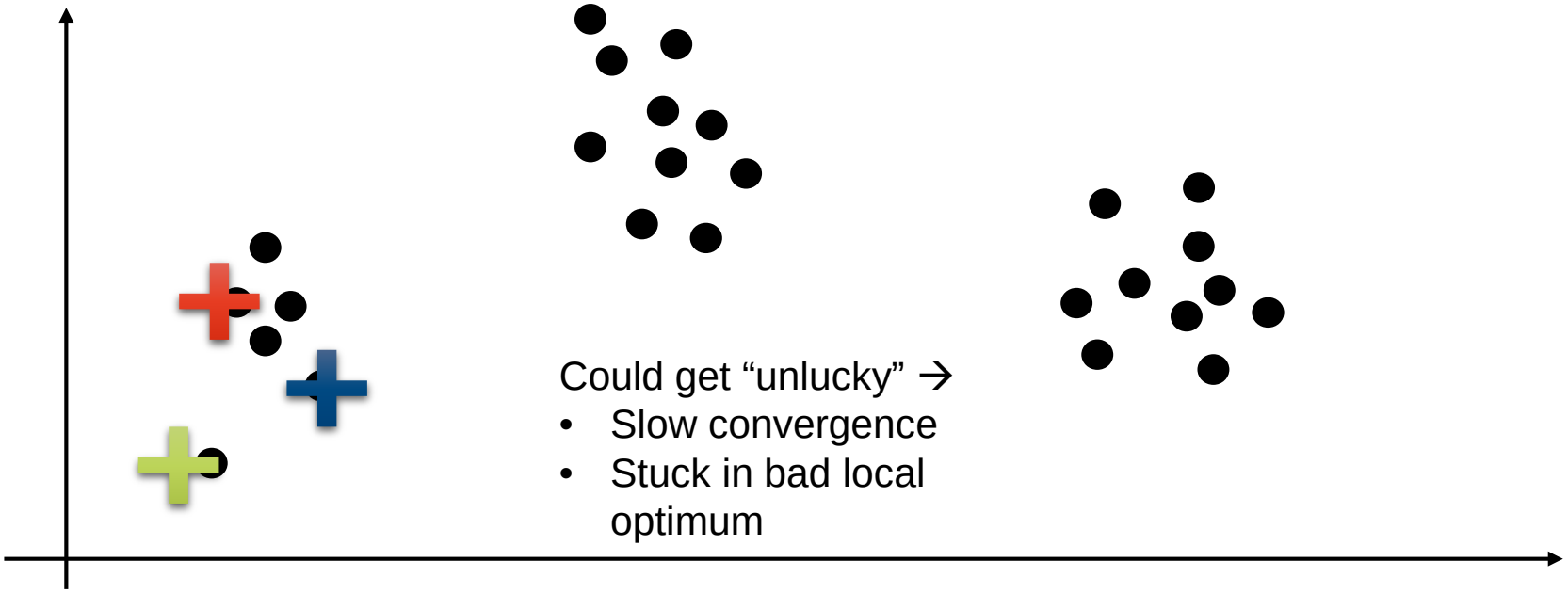
To a local optimum. ☹

Depends on Initial Centers

2.5: Picking the Initial Centers

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- **Simple Strategy:** select *k* data points at random
- **What could go wrong?**



Effect of initialization: https://www.youtube.com/watch?v=9nKfViAfajY

2.5 Initialization

NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

Random initialization:

- Most common: randomly choose some data points as starting centers.
- Draw starting points randomly from $\mathbb{R}^d$.
- Initialize the centers using the solution of an even simpler clustering algorithm.
- Ideally have prior knowledge, for example that certain points are in different clusters.

Common problem for all those methods: empty clusters (centers to which no data point is assigned). Then best solution: restart...

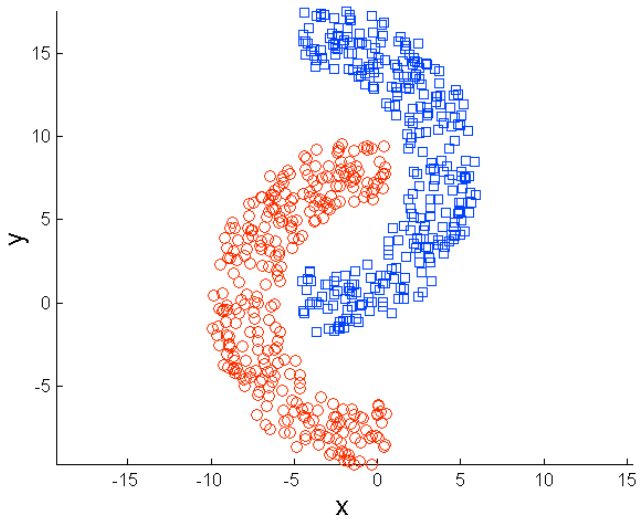
2.6 Implementation in scikit-learn

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

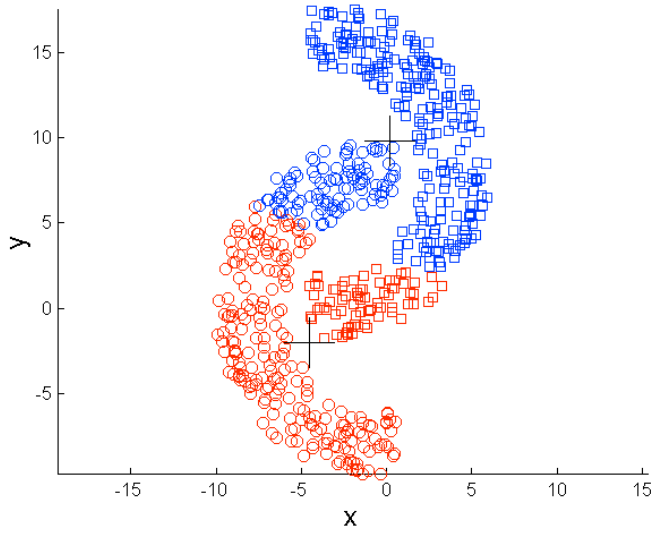
```
from sklearn.datasets import make_blobs
 # create blobs
data = make_blobs(n_samples=200, n_features=2, centers=4,
cluster_std=1.6, random_state=50)
# create np array for data points
points = data[0]

from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=4)
# fit kmeans object to data
kmeans.fit(points)
# print location of clusters learned by kmeans object
print(kmeans.cluster_centers_)
# save new clusters for chart
y_km = kmeans.fit_predict(points)
```

Limitations of K-means: Non-globular Shapes



Original Points



K-means (2 Clusters)

a. **Inertia W makes the assumption that clusters are convex and isotropic**, which is not always the case. It responds poorly to elongated clusters, or manifolds with irregular shapes.
b. **Inertia W is not a normalized metric:** we just know that lower values are better and zero is optimal. But in very high-dimensional spaces, Euclidean distances tend to become inflated (this is an instance of the so-called “curse of dimensionality”). Running a dimensionality reduction algorithm such as PCA prior to k-means clustering can alleviate this problem and speed up the computations.

2.7 More variants of K-means

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- **K-median**: here the centers are always data points. Can be used if we only have distances, but no coordinates of data points.
- **weighted K-means**: introduce weights for the individual data points
- **kernel-K-means**: the kernelized version of K-means (note that all boundaries between clusters are linear)
- **soft K-means**: no hard assignments, but “soft” assignments (often interpreted as “probability” of belonging to a certain cluster)

K-means is a simplified version of an EM-algorithm fitting a Gaussian mixture model.

2.8 K-medoid | K-maxoid clustering

- The **medoid (maxoid)** m of $\mathcal{X}$ coincides with the data point $x_j \in \mathcal{X}$ that is closest (farthest) to the mean μ. The point $x_j \in \mathcal{X}$ with the smallest (largest) average distance to all other points in X is closest to the sample mean μ.

Medoid:
$$\mathbf{m} = \mathbf{x}_j = \underset{\mathbf{x}_l}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{x}_l - \mathbf{x}_i\|^2$$

Maxoid:
$$\mathbf{m} = \mathbf{x}_j = \underset{\mathbf{x}_l}{\operatorname{argmax}} \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{x}_l - \mathbf{x}_i\|^2$$

- contrary to the μ_k in K-means, the m_k in K-medoids (maxoids) are guaranteed to coincide with data points so that **K-medoids (maxoids) clustering exclusively relies on distances between data points.**
- all distances evaluated during K-medoids (maxoids) clustering can therefore be precomputed and stored in a distance matrix $\boldsymbol{D}$ where

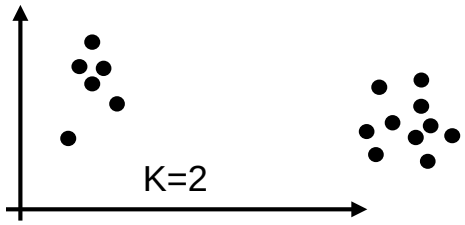
$$\mathbf{D}_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2$$

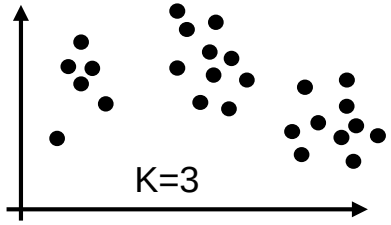
2.9 Heuristics for improving the result

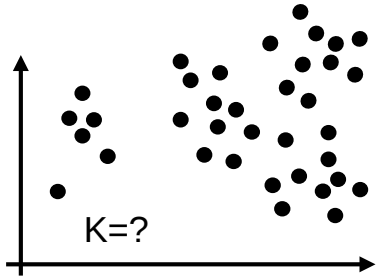
NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- **Restart** many times with different initializations.
- **Swap** individual points between clusters.
- **Remove a cluster center**, and introduce a completely new center instead.
- **Merge clusters**, and additionally introduce a completely new cluster center.
- **Split a cluster in two pieces** (preferably, one which has a very bad objective function). Then reduce the number of clusters again, for example by randomly removing one.

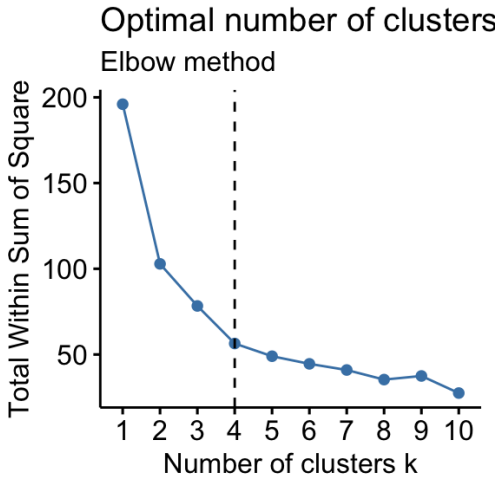
2.10 How do we choose K?







- Basic **Elbow Method**
- Try range of K values and plot average distance to centers
- Silhouette (graphical method, popular in stats)
- Cross-Validation (better)
  - **Repeatedly split the data into training and validation datasets**
  - **Cluster the training dataset**
  - **Measure avg. dist. to centers on validation data**



S. Still and W. Bialek. How many clusters? An information-theoretic perspective. Neural Comput., 16(12):2483 - 2506, 2004.

2.10 Silhouette (Peter J. Rousseeuw, 1986):

graphic method for K selection

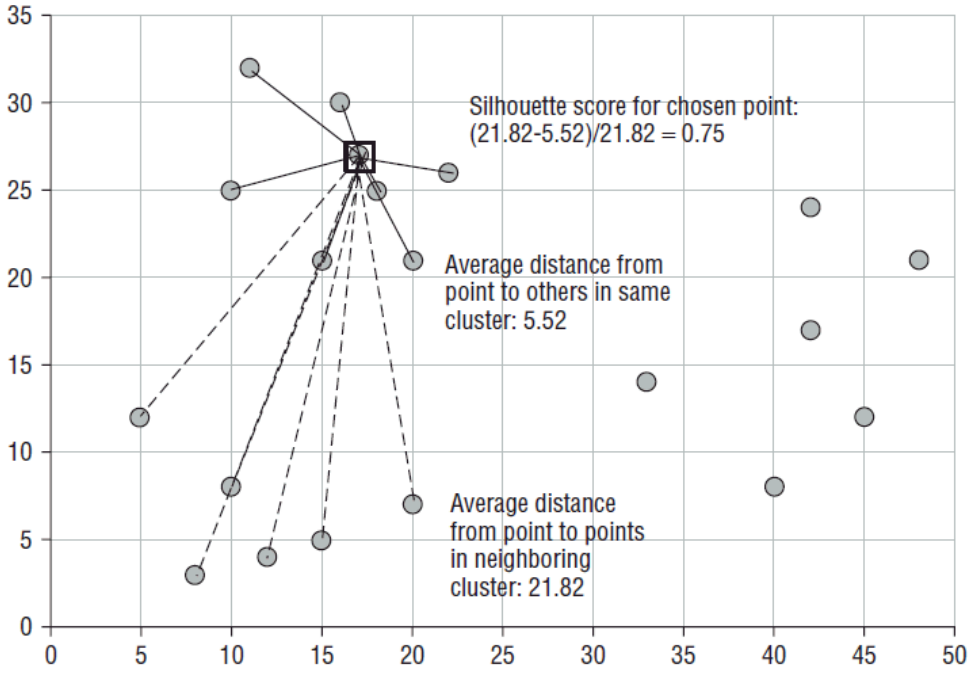
NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- Given K and K clusters, given any data point i, let a_i be the average distance or dissimilarity of i with all other points in the same cluster.
- For Euclidean k-means, use Euclidean distance for dissimilarity. a_i measures how well i fits into its cluster. b_i is the smallest average distance of i to other clusters.
- Define: **Silhouette score** $s_i \in [-1,1]$ $\quad s_i = \frac{b_i - a_i}{max(b_i, a_i)}$

- s_i is close to 1 if point i is in a tight cluster and far away from other clusters; close to -1, if it is in a loose cluster and close to other clusters.
- Maximize $\frac{1}{n}\sum_{i=1}^{n} s_i$ over k.

2.10 An example: consider the i^{th} point in the box

- Silhouette analysis can be used to study the separation distance between the resulting clusters.



a_i = 5.52

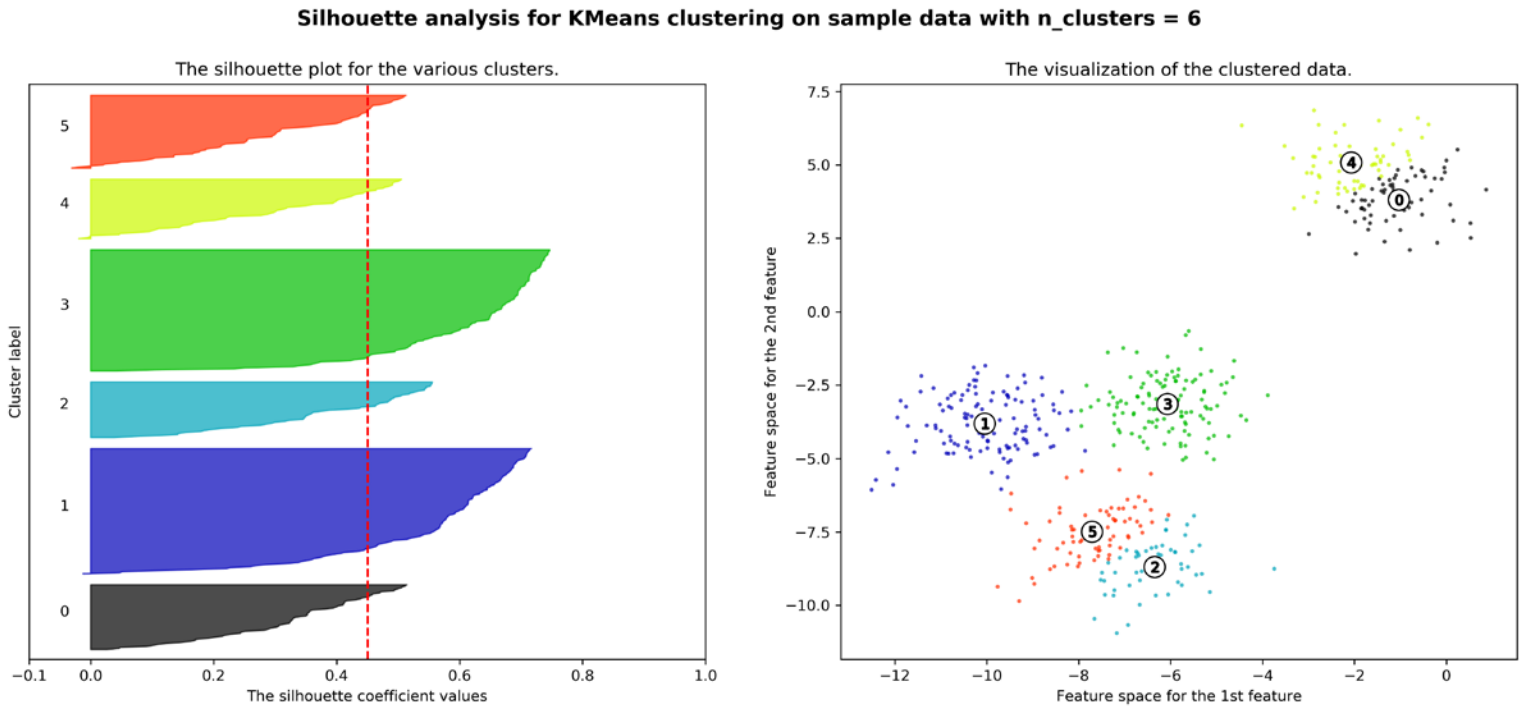
b_i = 21.82 (because the other cluster is further away by visal inspection)

s_i = 0.75 is the Silhouette score

So the ith point is in a pretty tight cluster

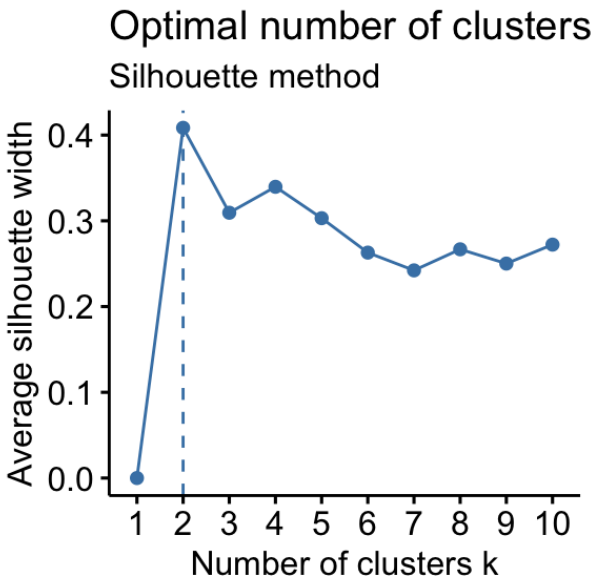
2.10 Silhouette Plot

- The **silhouette plot** displays a measure of how close each point in one cluster is to points in the neighboring clusters and thus provides a way to assess parameters like number of clusters visually.



2.11 Average Silhouette method

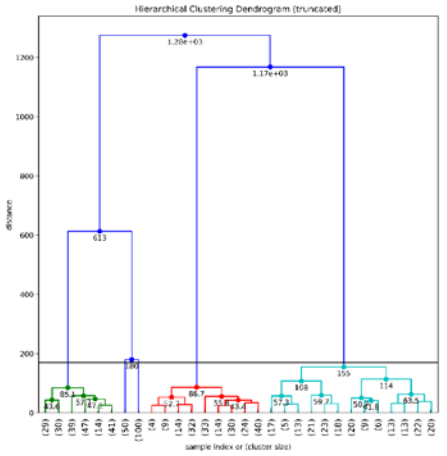
- Compute clustering algorithm (e.g., k-means clustering) for different values of k. For instance, by varying k from 1 to 10 clusters.
- For each k, calculate the **average silhouette of observations.**
- Plot the curve of average silhouette as function of to the number of clusters k.
- The location of the maximum is considered as the appropriate number of clusters.

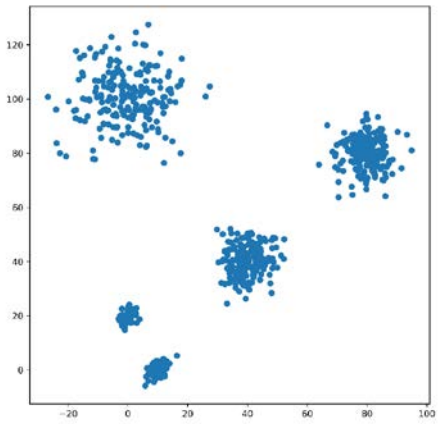


Alternative: **Gap Statistic**: Tibshirani R, Walther G, Hastie T. Estimating the number of clusters in a dataset via the gap statistic. Journal of the Royal Statistics Society 2001. (https://statweb.stanford.edu/~gwalther/gap)

3. Cluster Metrics (scores)

- **Inertia**
- **ARI**: Adjusted Rand Index
- **NMI**: Normalized Mutual Information
- **BIC**: Bayesian Information Criterium





3.1 Inertia W (`KMeans.inertia_`)

- The **within-cluster inertia W** of the partition C_K is the sum of the *inertia* of the clusters and measures then the **heterogeneity** within the clusters.

$$W = \sum_{k=1}^{K} I(C_k) \qquad I(C_k) = \sum_{i \in C_k} \|x_i - \mu_k\|^2$$

- The **between-cluster inertia B** of the partition C_K is the inertia of the gravity centers of the clusters weighted by μ_k and measures then the *separation between the clusters*. A good partition has a **large between-cluster inertia** and a **small within-cluster inertia**.

```
Sum_of_squared_distances = []
K = range(1,15) for k in K:
    km = KMeans(n_clusters=k)
    km = km.fit(data_transformed)
    Sum_of_squared_distances.append(km.inertia_)

plt.plot(K, Sum_of_squared_distances, 'bx-')
```

3.2 ARI (Rand index adjusted for chance)

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

```
sklearn.metrics.adjusted_rand_score(labels_true, labels_pred)
```

- The **Rand Index** (RI) computes a similarity measure between two clusterings by *considering all pairs of samples and counting pairs that are assigned in the same or different clusters* in the predicted and true clusterings.
- The raw RI score is then “adjusted for chance” into the ARI score using the following scheme:

$$\mathrm{ARI} = \frac{\mathrm{RI} - \mathrm{ExpectedRI}}{\max(\mathrm{RI}) - \mathrm{ExpectedRI}}$$

- The adjusted Rand index is thus ensured to have a value close to 0.0 for random labeling independently of the number of clusters and samples and exactly 1.0 when the clusterings are identical (up to a permutation).
- ARI is a symmetric measure.

[1] L. Hubert and P. Arabie, Comparing Partitions, Journal of Classification 1985
http://link.springer.com/article/10.1007%2FBF01908075
[2] https://en.wikipedia.org/wiki/Rand_index#Adjusted_Rand_index

3.3 NMI: Normalized Mutual Information

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

```
sklearn.metrics.normalized_mutual_info_score(labels_true,
 labels_pred, average_method='warn')
```

- **NMI** is a good measure for determining the quality of clustering.
- It is an external measure because we need the class labels of the instances to determine the NMI.
- Since it's normalized we can measure and compare the NMI between different clusterings having different number of clusters.

$$\mathrm{NMI} = \frac{2 \cdot I(Y;\mathrm{C})}{H[Y] + H[C]}$$

$$I(Y;C) = H[Y] - \sum_k H[Y|C_k]$$

$I[Y;C]$: mutual information between Y and C
$H[Y]$: Entropy of class labels
$H[C]$: Entropy of cluster labels

3.4 BIC: Bayesian Information Criterium

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

$$\mathrm{BIC} = \ln(\mathrm{n}) \cdot k - 2\ln(\hat{L})$$

$\hat{L}$: the maximized value of the likelihood function of the model M
$\hat{L} = p(x|\hat{\theta}, M)$ where $\hat{\theta}$ are the ML estimates of the parameters
x: observed data
n: number of data points
k: number of parameters estimated by the model M

- It can measure the efficiency of the parameterized model in terms of predicting the data.
- It penalizes the complexity of the model where complexity refers to the number of parameters in the model.
- It can be used to choose the number of clusters according to the intrinsic complexity present in a particular dataset.
- It is independent of the prior.

3.4 Metrics to evaluate cluster algorithms: BIC

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- Bayesian Information Criterium (BIC)

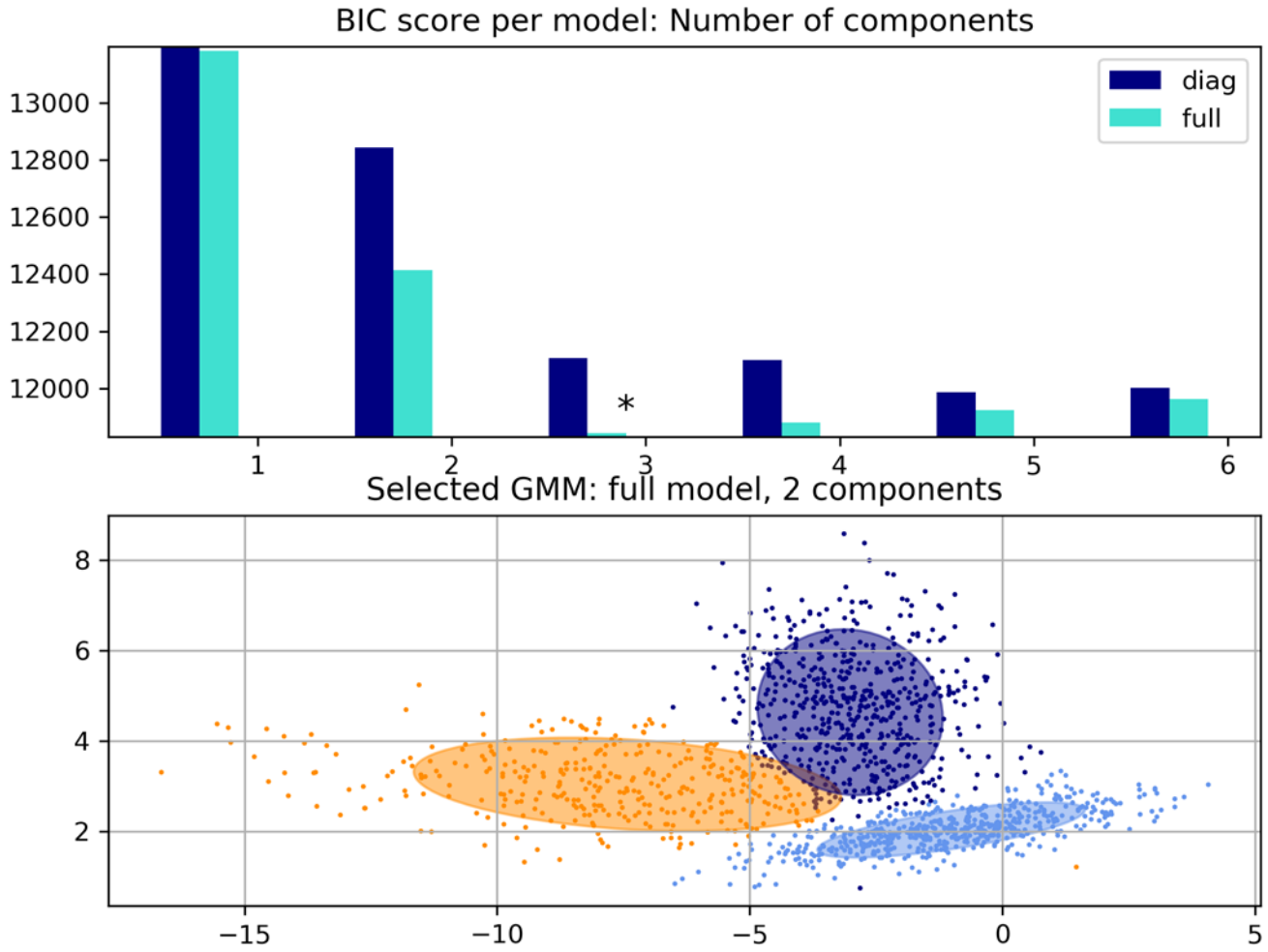
```
from sklearn import mixture
lowest_bic = np.infty; bic = []
n_components_range = range(1, 7)
for n_components in n_components_range:
     # Fit a Gaussian mixture with EM
    gmm =mixture.GaussianMixture(n_components=n_components,
        covariance_type='full')
    gmm.fit(X)
    bic.append(gmm.bic(X))
    if bic[-1] < lowest_bic:
        lowest_bic = bic[-1]
        best_gmm = gmm

bic = np.array(bic)
```

3.4 BIC = Bayesian Information Criterium

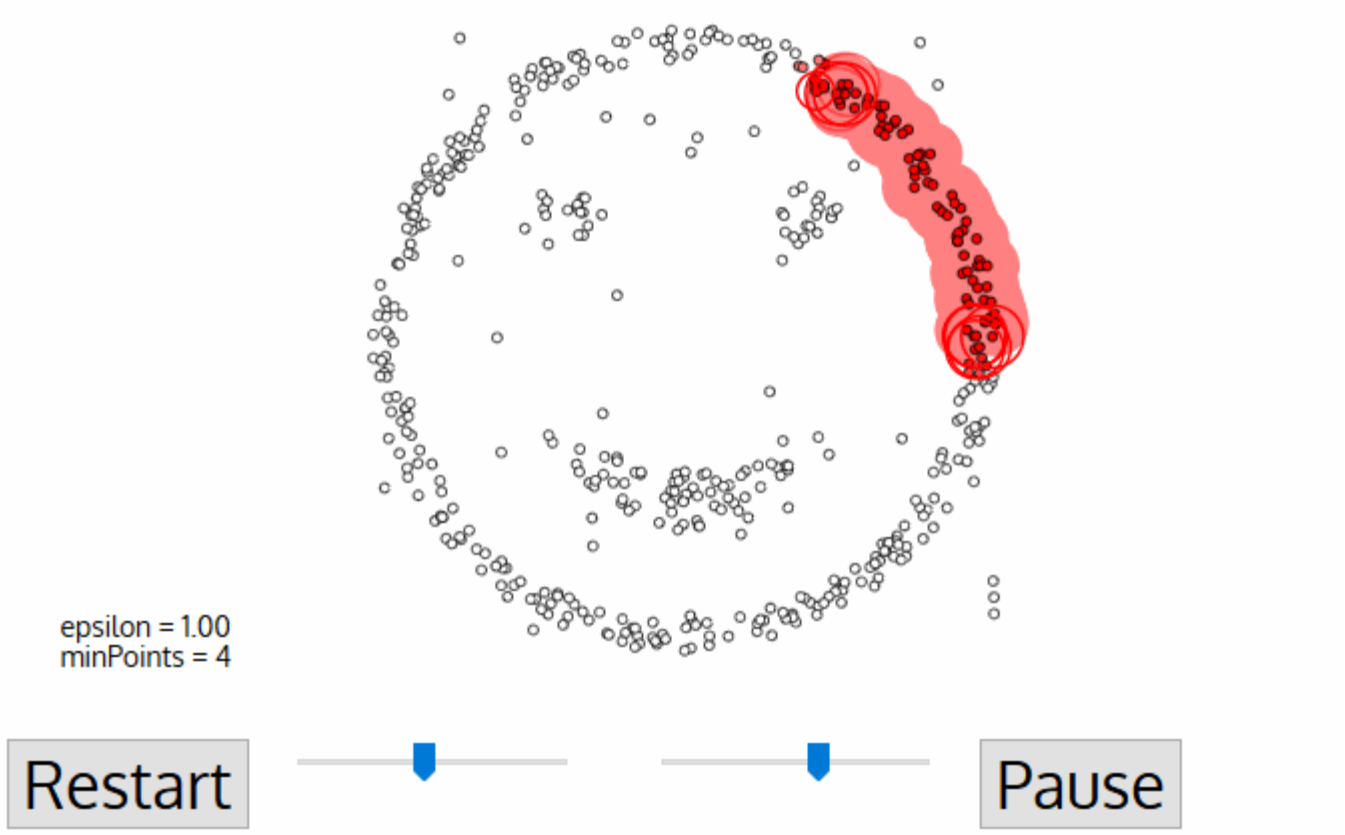
NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- Alternative to elbow-curve (plotting of the inertia)



4. Density basesd Clustering

- DBSCAN



4. DBSCAN=density-based spatial clustering of applications with noise

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- The basic idea of DBSCAN is that **clusters form dense regions** in the data and are separated by relatively empty areas. Points within a dense region are called core points. DBSCAN identifies points in "densely populated" regions of the feature space in which many data points lie close together.
- Advantages of DBSCAN:
  - the user **can not set the number of clusters** a priori
  - DBSCAN is able to capture clusters with **complex shapes**,
  - it identifies points that do not belong to any of the clusters.
- The DBSCAN procedure is slower than the agglomerative clustering and k-Means, but scales relatively well for large data sets.

4.1 How DBSCAN works

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- In DBSCAN there are two parameters: **`min_samples`** and **`eps`**.
- If at least `min_samples` data points are within the distance `eps` to a given point, this data point is classified as a **core object**. Core objects that are closer than `eps` to each other are assigned to the same cluster.
- At the beginning, the algorithm selects any starting point. Then it finds all points at distance `eps` or closer to this point. If less than `min_samples` points are found within the distance `eps` to the starting point, this point will be classified as **noise**. (It does not belong to any cluster).
- If there is more as `min_samples` points at a distance of `eps`, the point is used as core object and receives a new cluster designation.

4.3 DBSCAN Algorithm

Eliminate noise points

Perform clustering on the remaining points

$current\_cluster\_label \leftarrow 1$
for all core points **do**
  if the core point has no cluster label **then**
    $current\_cluster\_label \leftarrow current\_cluster\_label + 1$
    Label the current core point with cluster label $current\_cluster\_label$
  end if
  for all points in the Eps-neighborhood, except i^{th} the point itself **do**
    if the point does not have a cluster label **then**
      Label the point with cluster label $current\_cluster\_label$
    end if
  end for
end for

Summary

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- Clustering is an old activity and is used for information organization
- **Agglomerative clustering**: Linkage and distance metrics
- **K-means algorithm**:
  - initial values, choice of K
  - *Euclidean distance* in K-means corresponds to taking means – sensitive to outliers because of the squared Euclidean distance;
  - using *median* corresponds to absolute loss function, robust.
- Advantages of DBSCAN:
  - the user **can not set the number of clusters** a priori
  - DBSCAN is able to capture clusters with **complex shapes**,
  - it identifies points that do not belong to any of the clusters.
- We do not always have labels to compare – other investigation is needed to back up why the clustering results are meaningful in context

Appendix

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large n_samples, medium n_clusterswith MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry	Distances between points
Spectral clustering	number of clusters	Medium n_samples, small n_clusters	Few clusters, even cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters	Large n_samples and n_clusters	Many clusters, possibly connectivity constraints	Distances between points
Agglomerative clustering	number of clusters, linkage type, distance	Large n_samples and n_clusters	Many clusters, possibly connectivity constraints, non Euclidean distances	Any pairwise distance
DBSCAN	neighborhood size	Very large n_samples, medium n_clusters	Non-flat geometry, uneven cluster sizes	Distances between nearest points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation	Mahalanobis distances to centers
Birch	branching factor, threshold, optional global clusterer.	Large n_clustersand n_samples	Large dataset, outlier removal, data reduction.	Euclidean distance between points

https://scikit-learn.org/stable/modules/clustering.html

Characteristics of Data, Clusters, and Clustering Algorithms

NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- A cluster analysis is affected by characteristics of
- **Data**
- **Clusters**
- **Clustering algorithms**

- Looking at these characteristics gives us a number of dimensions that you can use to describe clustering algorithms and the results that they produce

Comparison of DBSCAN and K-means

NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- Both are partitional.
- K-means is complete; DBSCAN is not.
- K-means has a prototype-based notion of a cluster; DB uses a density-based notion.
- K-means can find clusters that are not well-separated. DBSCAN will merge clusters that touch.
- DBSCAN handles clusters of different shapes and sizes; K-means prefers globular clusters.

Comparison of DBSCAN and K-means

NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- DBSCAN can handle noise and outliers; K-means performs poorly in the presence of outliers
- K-means can only be applied to data for which a centroid is meaningful; DBSCAN requires a meaningful definition of density
- DBSCAN works poorly on high-dimensional data; K-means works well for some types of high-dimensional data
- Both techniques were designed for Euclidean data, but extended to other types of data
- DBSCAN makes no distribution assumptions; K-means is really assuming spherical Gaussian distributions

Comparison of DBSCAN and K-means

NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- K-means has an O(n) time complexity; DBSCAN is O(n^2)
- Because of random initialization, the clusters found by K-means can vary from one run to another; DBSCAN always produces the same clusters
- DBSCAN automatically determines the number of clusters; K-means does not
- K-means has only one parameter, DBSCAN has two.
- K-means clustering can be viewed as an optimization problem and as a special case of EM clustering; DBSCAN is not based on a formal model.

Literature

1) Tibshirani, G. Walther, and T. Hastie. *Estimating the number of clusters in a dataset via the gap statistic*. J. Royal. Statist. Soc. B, 63(2):411-423, 2001.

2) S. Still and W. Bialek. *How many clusters? An information-theoretic perspective*. Neural Comput., 16(12):2483 - 2506, 2004.

3) C. Fraley and A. E. Raftery. *Model-based clustering, discriminant analysis, and density estimation*. JASA, 97:611 -631, 2002.

4) S. Zhong and J. Ghosh. *A Unified framework for model-based clustering*. JMLR, 4:1001 - 1037, 2003

5) S. C. Johnson. *Hierarchical clustering schemes*. Psychometrika, 2:241 - 254, 1967.