Data clustering

Thomas Bonald

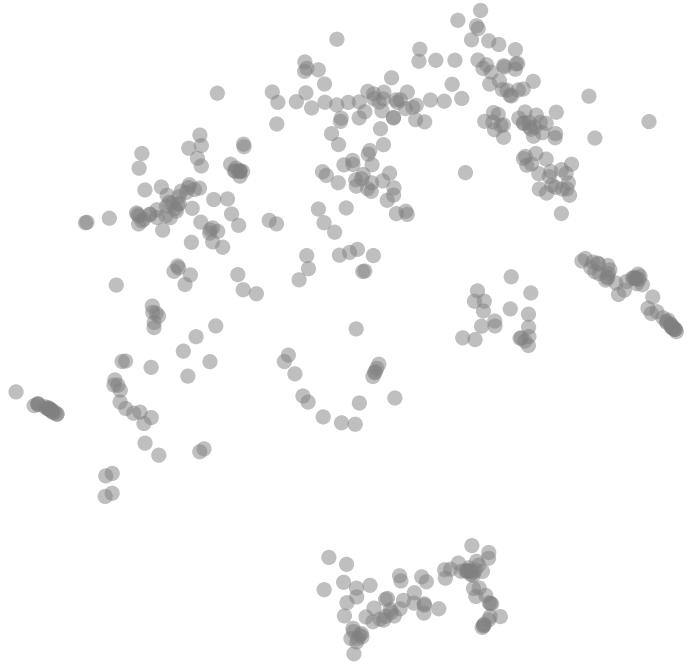
2019 – 2020

Data clustering

Let $x_1, \ldots, x_n$ be n data samples of $\mathbb{R}^d$

Problem

How to group these n samples into clusters, so that **close** samples tend to be in the same cluster?



Motivation

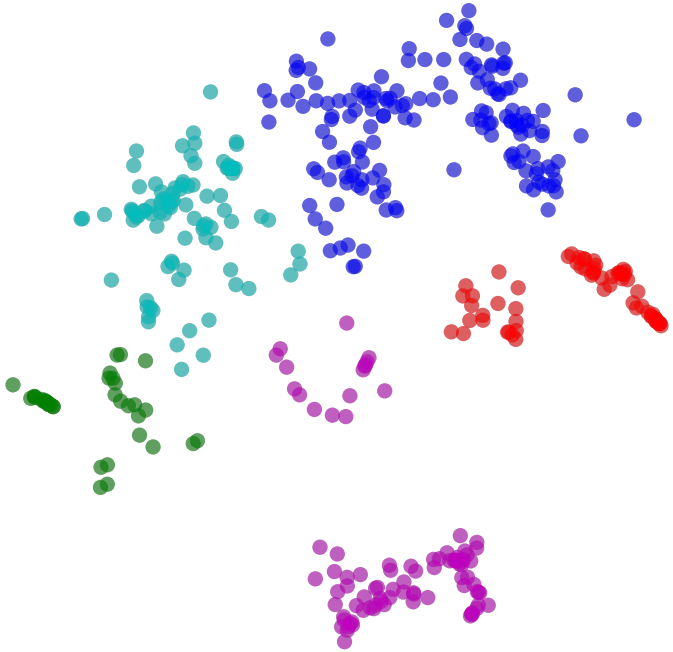
A fundamental problem in **unsupervised learning**

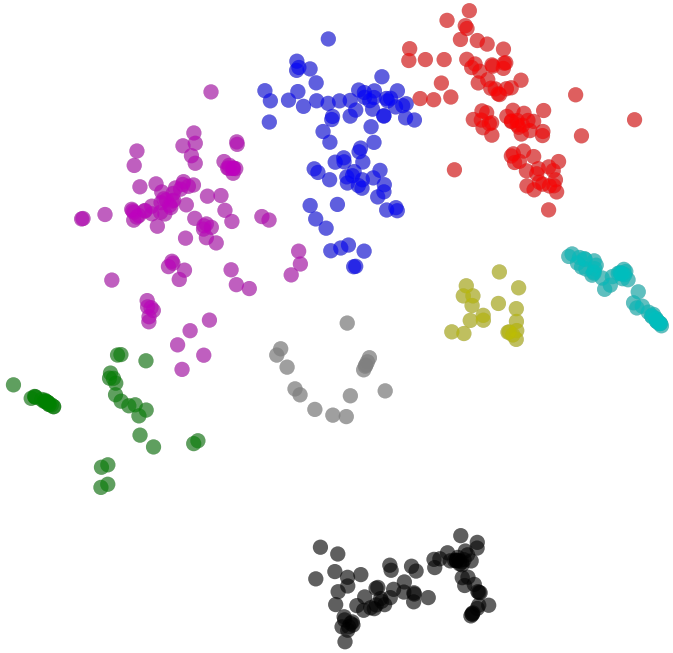
Many applications:

- Recommender systems
- Anomaly detection
- Visualization
- Community detection
- Search engines
- Image segmentation
- NLP

An ill-posed problem

- What is a **good** clustering?
- **How many** clusters?





Kleinberg's impossibility theorem

Viewing clustering as a function $f : x \mapsto C$ with $x = (x_1, \ldots, x_n)$

Axioms

1. **Scale-invariance:** $\forall \alpha > 0, f(\alpha x) = f(x)$
2. **Richness:** f surjective
3. **Consistency:** $\forall y \succ x$ [1], $f(y) = f(x)$

There is **no** clustering function f satisfying these 3 axioms!
Kleinberg 2002

[1] Points closer in y than in x if and only if in the same cluster.

Main clustering algorithms

Algorithm	Param.	Soft	Scalable	Online
k-means	k	(✓)	+	
Mini-batch k-means	k	(✓)	++	(✓)
Gaussian Mixture	k	✓		(✓)
DBSCAN[2]	Distance		++	
Mean Shift	Distance		−	
Affinity Propagation	Damping	(✓)	−−	
Ward			+	
BIRCH[3]	Tree		+	(✓)

See `https://scikit-learn.org/`

[2] Density-Based Spatial Clustering of Applications with Noise

[3] Balanced Iterative Reducing and Clustering using Hierarchies

Outline of the course

Algorithms:

1. k-means
2. Gaussian mixture
3. Hierarchical clustering

Performance metrics

The k-means algorithm

Thomas Bonald

2019 – 2020

Introduction

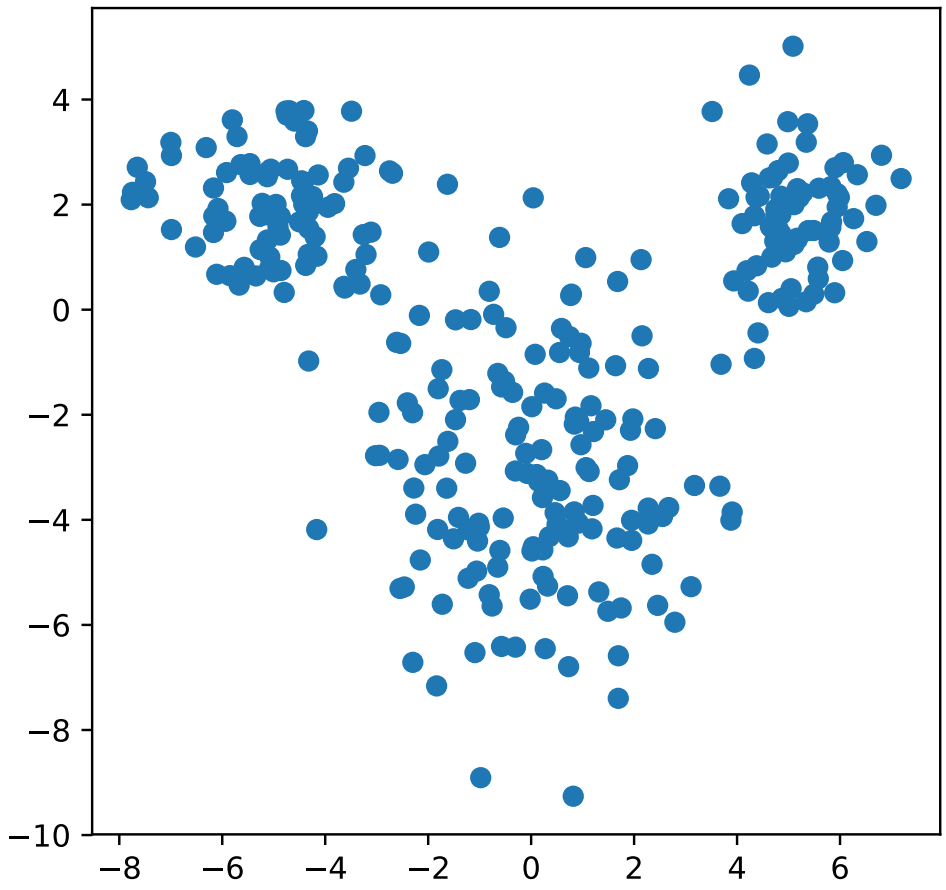
Let $x_1, \ldots, x_n$ be n data samples of $\mathbb{R}^d$

Problem

How to group these n samples into k clusters, so that **close** samples (for the Euclidean distance) tend to be in the same cluster?

Note: The number of clusters k is an **input** parameter.

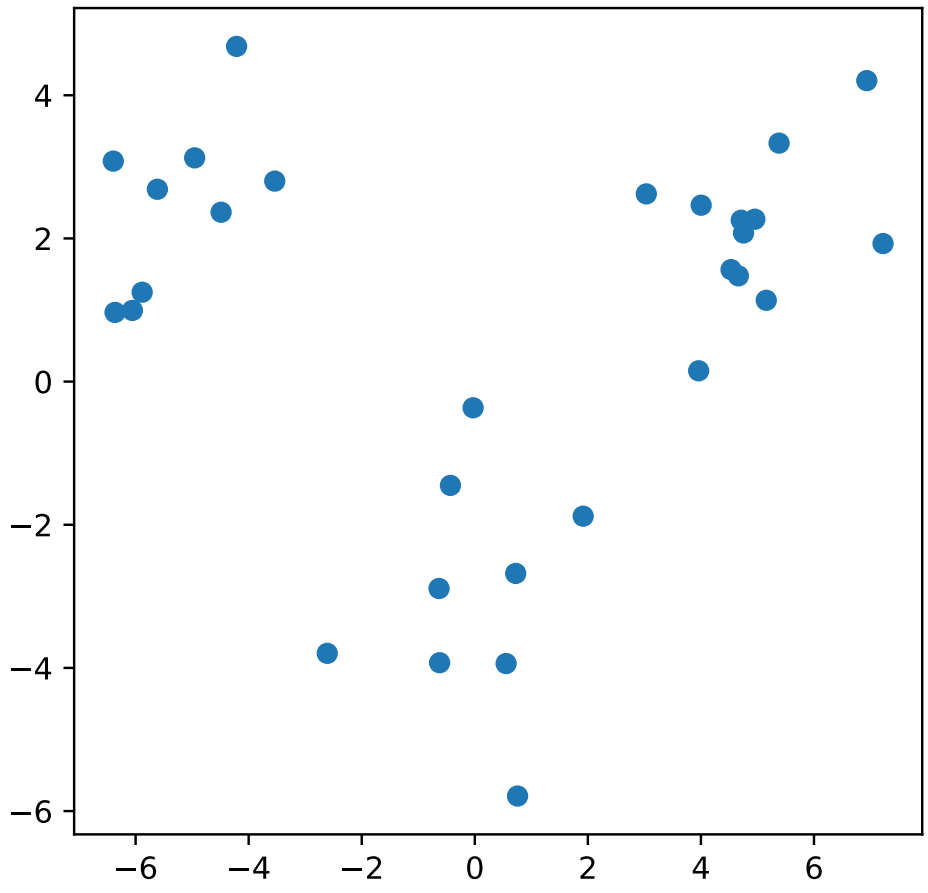
Example in $\mathbb{R}^2$



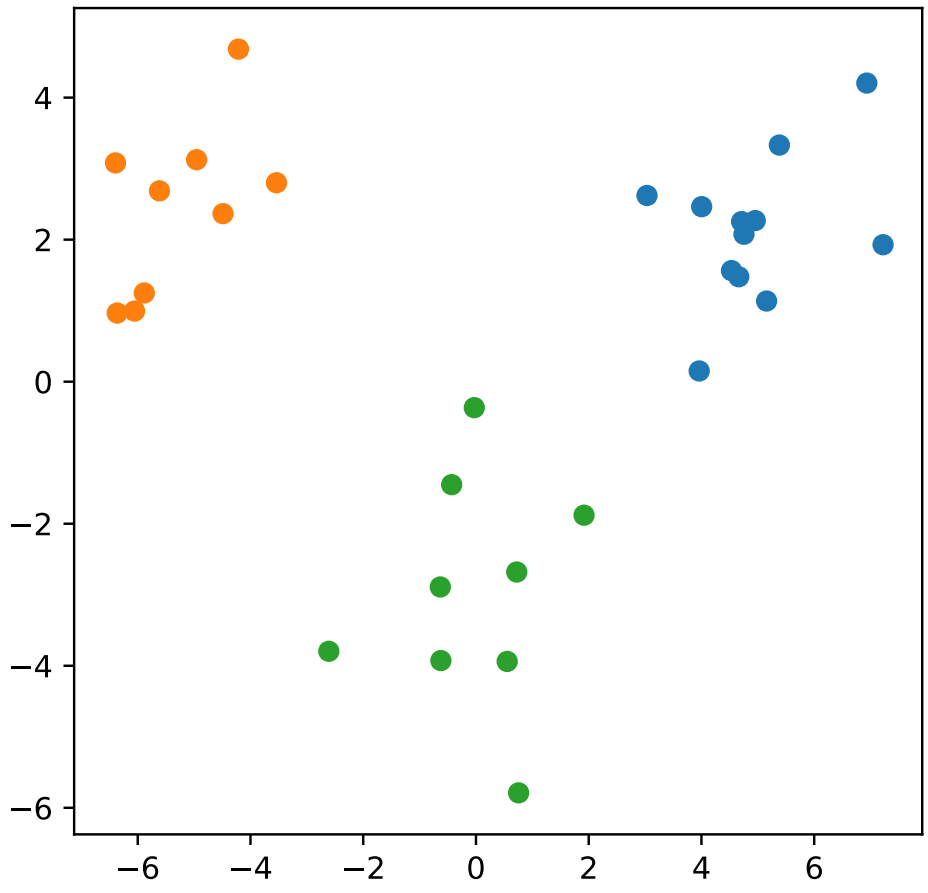
Outline

1. Cost function
2. The k-means algorithm
3. Variants
4. Applications

Cost function



Cost function



Cost function

Let $C_1, \ldots, C_k$ be a **partition** of the n samples

Cost function

Sum of the **square distances** to the cluster center:

$$\sum_{j=1}^{k} \sum_{i \in C_j} ||x_i - \mu_j||^2$$

where μ_j is the center of cluster j:

$$\mu_j = \frac{1}{|C_j|} \sum_{i \in C_j} x_i$$

Note: Can be viewed as the **moment of inertia** of the system

Optimization problem

Consider the problem:

$$\min_{C_1,\ldots,C_k} \sum_{j=1}^{k} \sum_{i \in C_j} ||x_i - \mu_j||^2$$

with

$$\mu_j = \frac{1}{|C_j|} \sum_{i \in C_j} x_i$$

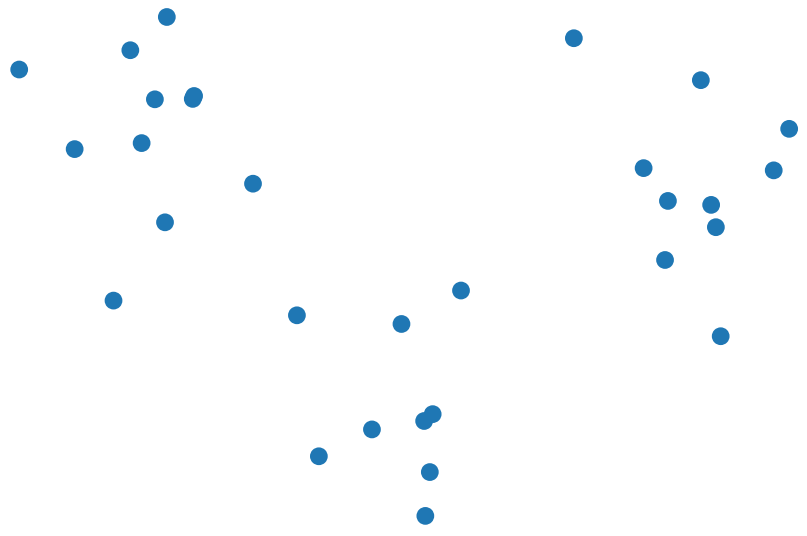
- This problem is **NP-hard**
- The k-means algorithm provides an **approximate solution**

The k-means algorithm (Lloyd 1957)

Algorithm

Starting from some arbitrary location of the cluster centers:

- Find the partition induced by the **nearest cluster center**
- Update the cluster centers
- Iterate until convergence



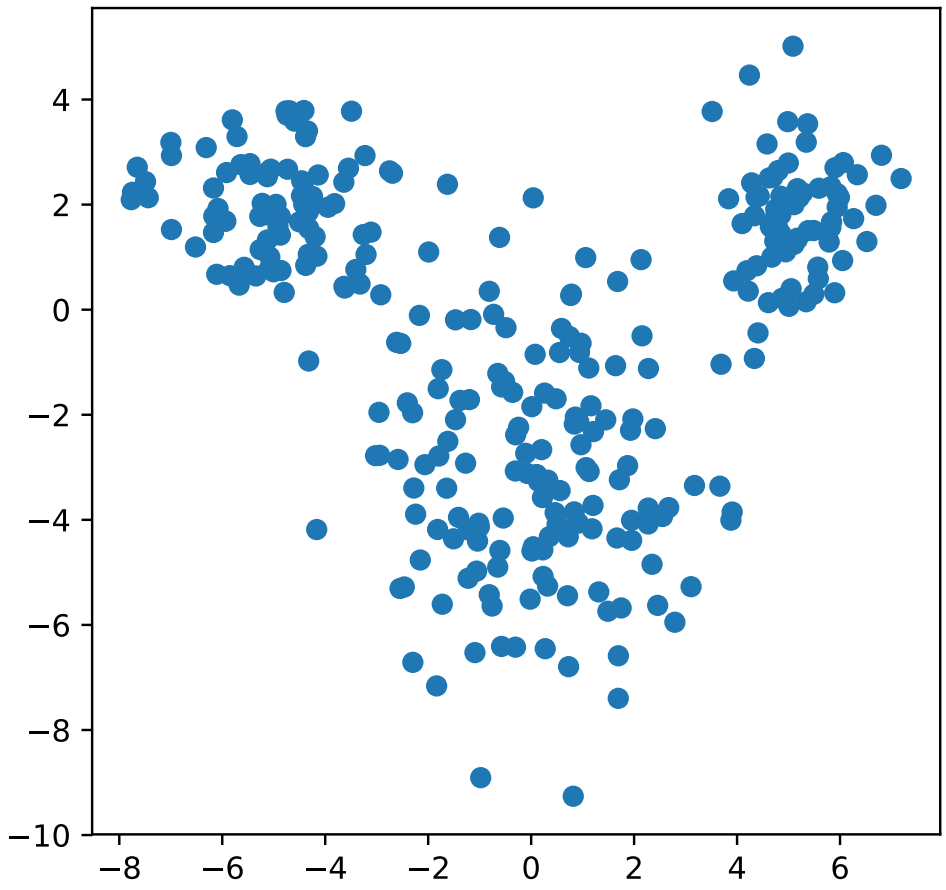
Convergence

- Writing the optimization problem as:

$$\min_{C_1,\ldots,C_k;\mu_1,\ldots,\mu_k} \sum_{j=1}^{k} \sum_{i \in C_j} ||x_i - \mu_j||^2$$

  we see that k-means performs **alternating minimization**:
  1. in the clusters $C_1, \ldots, C_k$, for fixed $\mu_1, \ldots, \mu_k$,
  2. in the cluster centers $\mu_1, \ldots, \mu_k$, for fixed $C_1, \ldots, C_k$.
- In particular, the cost **decreases** at each update
- Since there is a finite number of configurations, the algorithm **converges**

Initialization

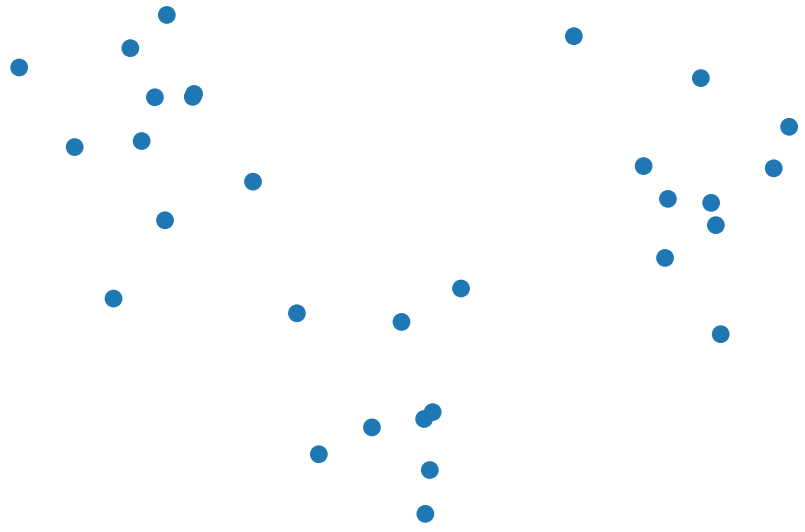


k-means++

Idea: Select initial cluster centers far from one another

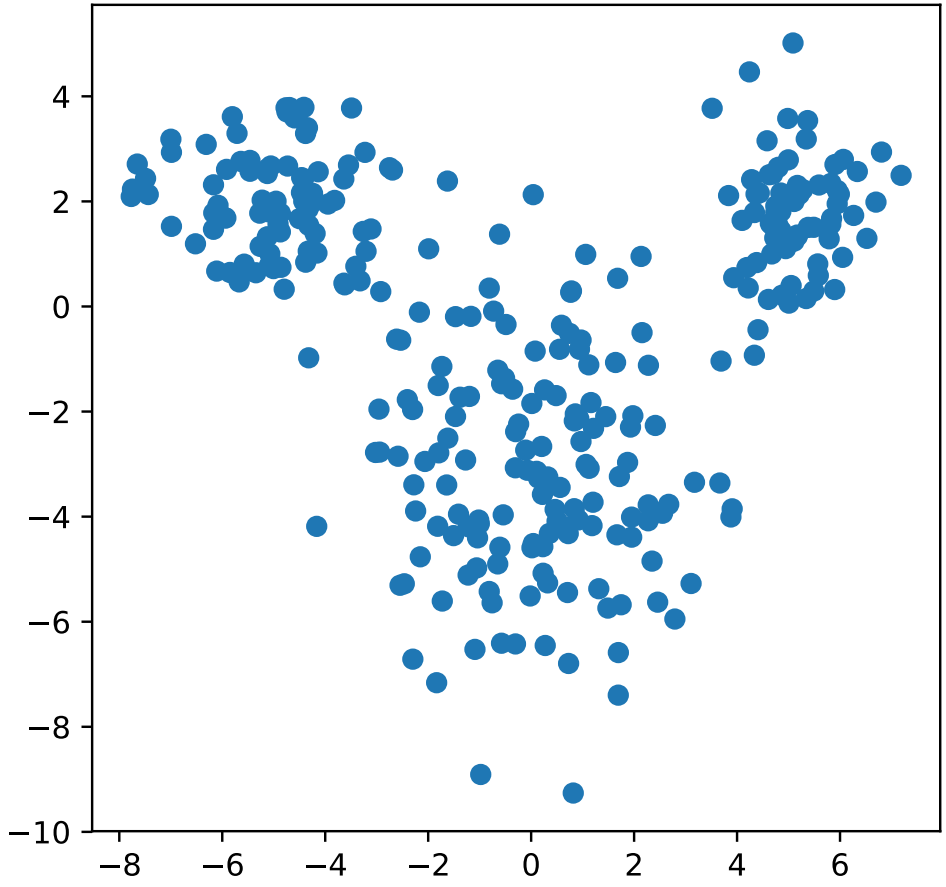
Algorithm

- Select the first cluster center uniformly at random
- Select the following cluster centers at random among the data samples with a probability proportional to the **square distance** to the closest current cluster center



Complexity of k-means

- $O(nk)$ operations per iteration
- Not feasible for large n and k



Mini-batch k-means

Idea: Local update based on a mini-batch of b samples

Algorithm

Starting from some arbitrary location of the cluster centers:

- Choose at random a **mini-batch** $y_1, \ldots, y_b$ of b data samples
- Find the partition of these samples, say $B_1, \ldots, B_k$
- Update the cluster centers using for each cluster j:

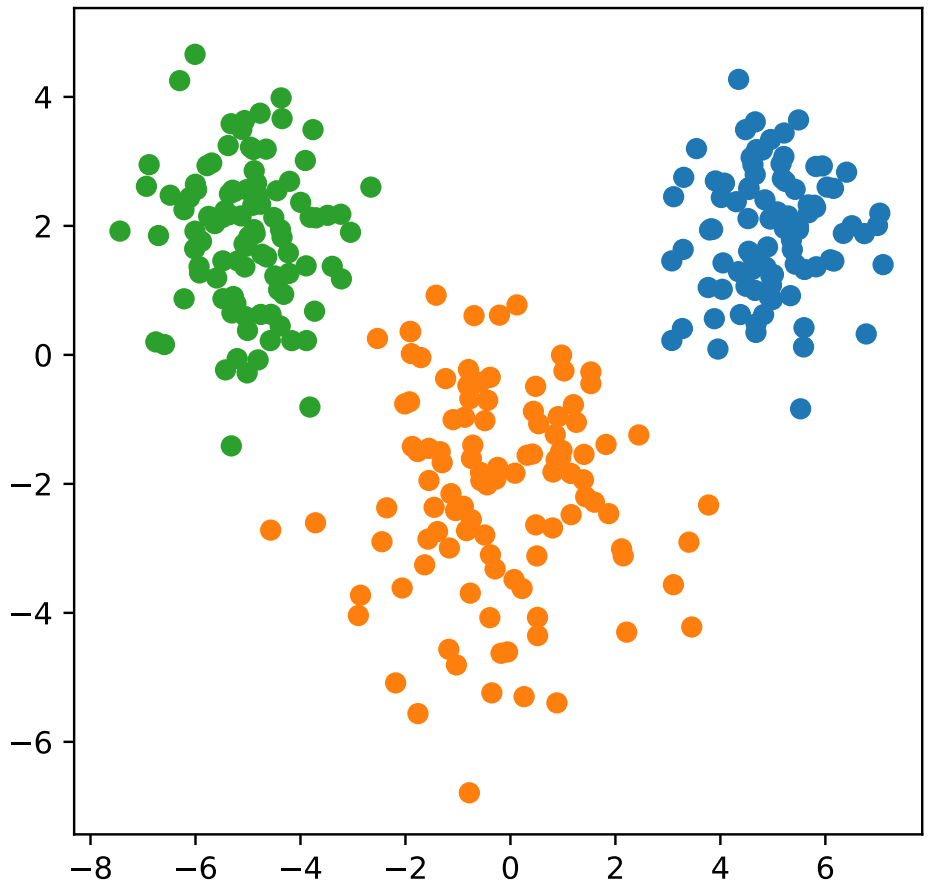
$$\mu_j \leftarrow \frac{s_j \mu_j + \sum_{i \in B_j} y_i}{s_j + |B_j|}$$

$$s_j \leftarrow s_j + |B_j|$$

- Iterate

Complexity in $O(bk)$ per iteration instead of $O(nk)$

Soft clustering



Cost function

Let p_{ij} be the **probability** that sample i belongs to cluster j

(Expected) cost function

$$\min_{p_1,\ldots,p_n;\mu_1,\ldots,\mu_k} \sum_{j=1}^{k} \sum_{i=1}^{n} p_{ij} ||x_i - \mu_j||^2$$

This is a **relaxation** of the original problem
We get the cluster centers:

$$\mu_j = \frac{\sum_{i=1}^{n} p_{ij} x_i}{\sum_{i=1}^{n} p_{ij}}$$

Note: The alternating minimization leads to k-means...

Soft k-means

Algorithm

Starting from some arbitrary location of the cluster centers:

- Compute the **probability** that sample i lies in cluster j,

$$p_{ij} \propto e^{-\frac{||x_i - \mu_j||^2}{2\sigma^2}}$$

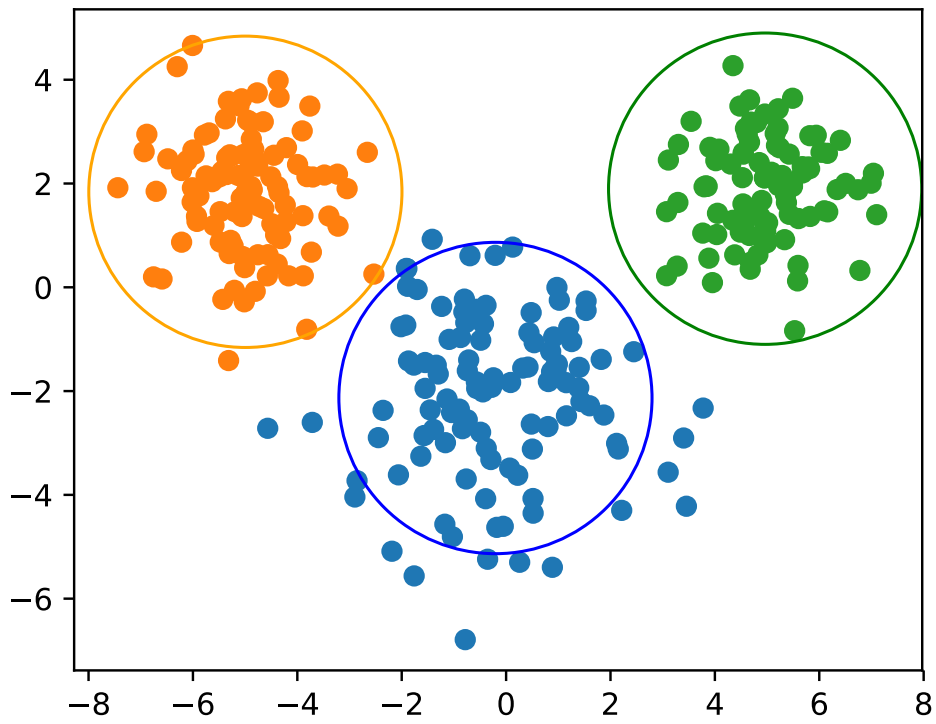
- Update the cluster centers,

$$\mu_j = \frac{\sum_{i=1}^{n} p_{ij} x_i}{\sum_{i=1}^{n} p_{ij}}$$

- Iterate until convergence

The parameter σ controls the **spread** of the clusters

Impact of σ



Fuzzy k-means (Dunn 1973)

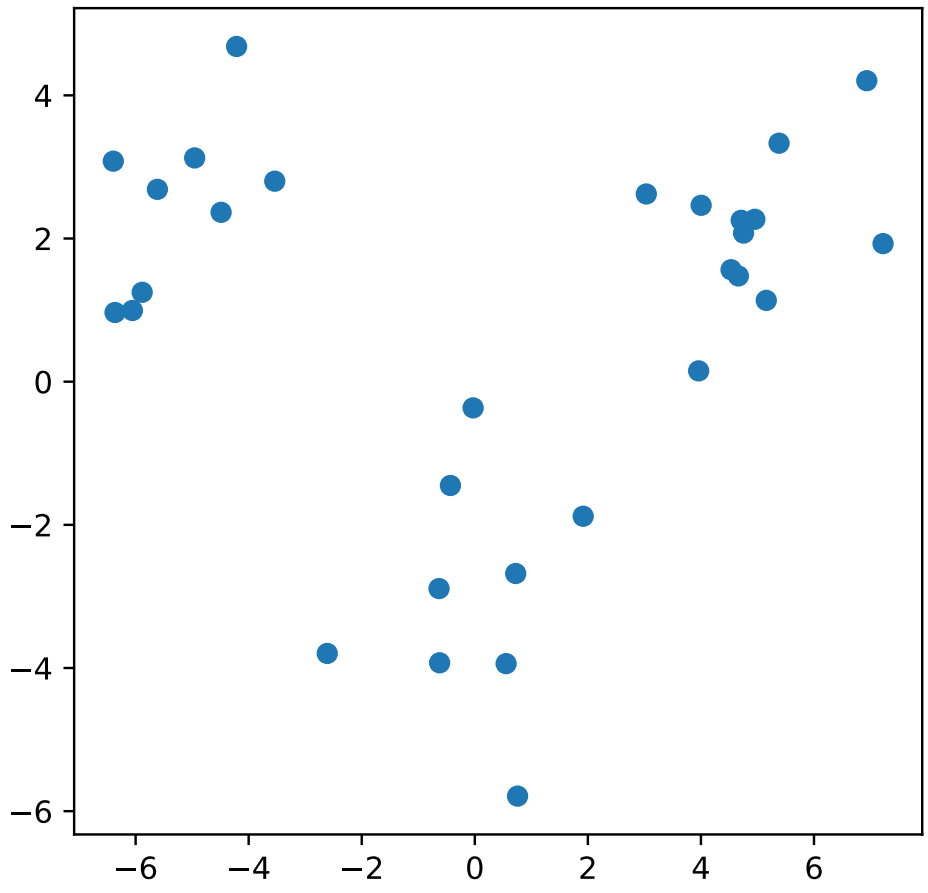
- The **fuzzy** k-**means** algorithm uses a Pareto distribution:

$$p_{ij} \propto \frac{1}{||x_i - \mu_j||^{\alpha}}$$

  for some parameter $\alpha > 0$
- For some reason, this algorithm is known as **fuzzy** c-**means** instead of fuzzy k-means...

Sample weights

- Assume each sample is assigned some positive **weight**
- The weight captures the relative **importance** of the sample, e.g., in terms of value, reliability, multiplicity



Cost function

Let $w_1, \ldots, w_n$ be the weights of the samples

Cost function

Weighted sum of **square distances** to the cluster centers:

$$\sum_{j=1}^{k} \sum_{i \in C_j} w_i ||x_i - \mu_j||^2$$

where μ_j is the centroid of cluster j:

$$\mu_j = \frac{\sum_{i \in C_j} w_i x_i}{\sum_{i \in C_j} w_i}$$

Note: Can be viewed as the **moment of inertia** of the system, with $w_1, \ldots, w_n$ the **masses** of the corresponding point particles

Weighted k-means

Algorithm

Starting from some arbitrary location of the cluster centers:

- Find the partition induced by the **nearest cluster center**
- Update the cluster centers:

$$\mu_j = \frac{\sum_{i \in C_j} w_i x_i}{\sum_{i \in C_j} w_i}$$

- Iterate until convergence

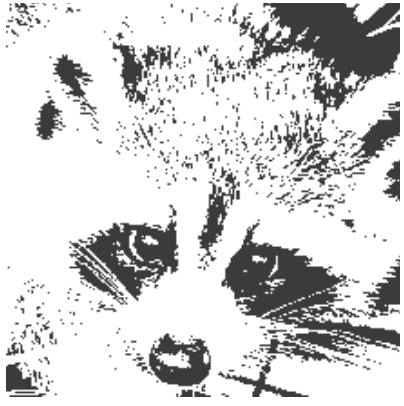
Note: The initial cluster centers must be chosen at random, in proportion to the weights

Application: Image segmentation by k-means

Data

k-means ($k = 4$)

Cluster 1



Cluster 2

Application: Clustering Wikipedia articles

k-means clustering (with $k = 20$) of the spectral embedding in dimension 100 of the graph of Wikipedia for schools[1]

#	Main articles
1113	Australia, Canada, North America, 20th century
326	UK, England, London, Scotland, Ireland
250	US, New York City, BBC, 21st century, Los Angeles
227	India, Japan, China,United Nations
218	Earth, Sun, Physics, Hydrogen, Moon, Astronomy
200	Mammal, Fish, Horse, Cattle, Extinction
200	France, Italy, Spain, Latin, Netherlands
198	Water, Agriculture, Coal, River, Antarctica
197	Germany, World War II, Russia, World War I
187	Mexico, Brazil, Atlantic Ocean, Argentina
185	Human, Philosophy, Slavery, Religion, Democracy
184	Plant, Rice, Fruit, Sugar, Wine, Maize, Cotton
177	Gold, Iron, Oxygen, Copper, Electron, Color
170	Egypt, Turkey, Israel, Islam, Iran, Middle East
159	English, 19th century, William Shakespeare, Novel
158	Africa, South Africa, Time zone, Portugal
157	Europe, Scientific classification, Animal, Asia
141	Washington, D.C., President of the United States
72	Dinosaur, Fossil, Reptile, Cretaceous, Jurassic
70	Paris, Art, Architecture, Painting, Hist. of painting

[1] Graph of 4,589 nodes and 106,644 edges

The Gaussian Mixture Model

Thomas Bonald

2019 – 2020

Introduction

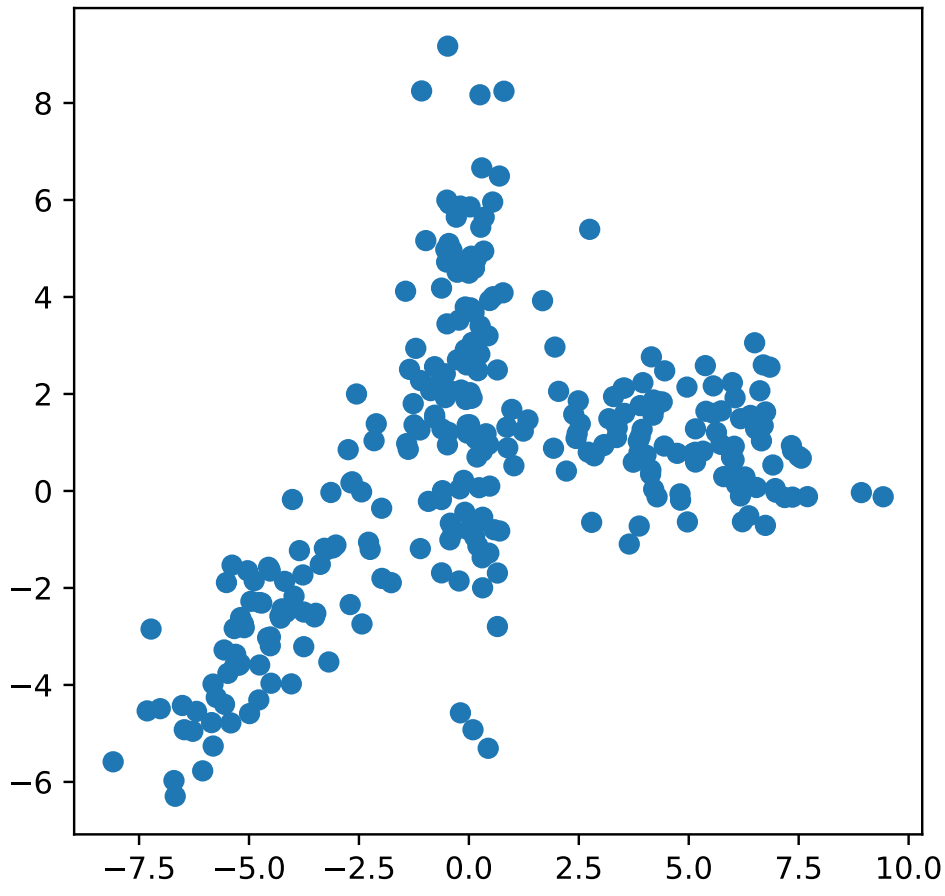
Let $x_1, \ldots, x_n$ be n data samples of $\mathbb{R}^d$

Problem

How to group these n samples into k clusters, so that **close** samples (for the Euclidean distance) tend to be in the same cluster?

Note: The number of clusters k is an **input** parameter

Example in $\mathbb{R}^2$



Outline

1. The Gaussian mixture model
2. Expectation-Maximization
3. Comparison with k-means
4. Examples

The Gaussian model

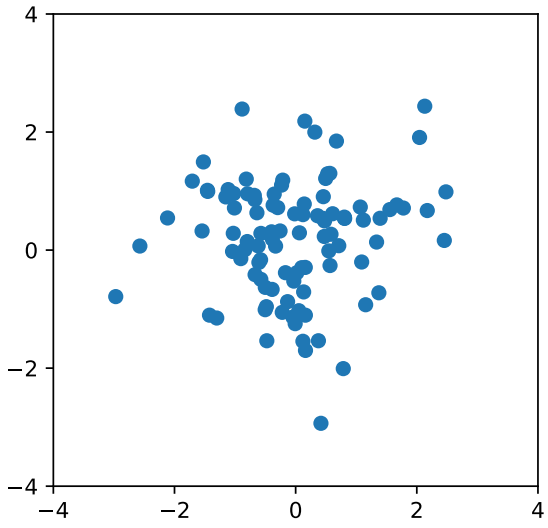
- Gaussian vector:

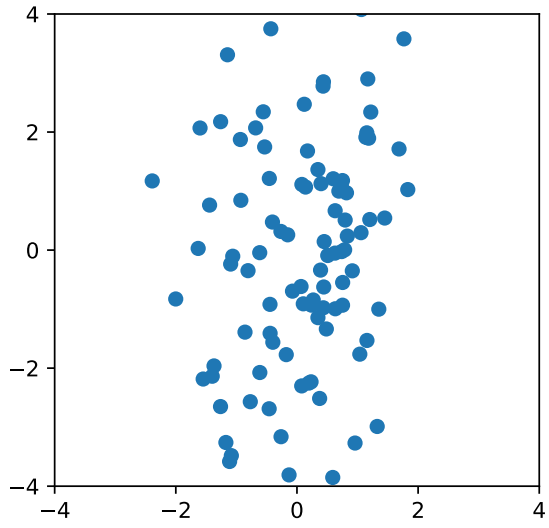
$$X \sim \mathcal{N}(\mu, \Sigma)$$

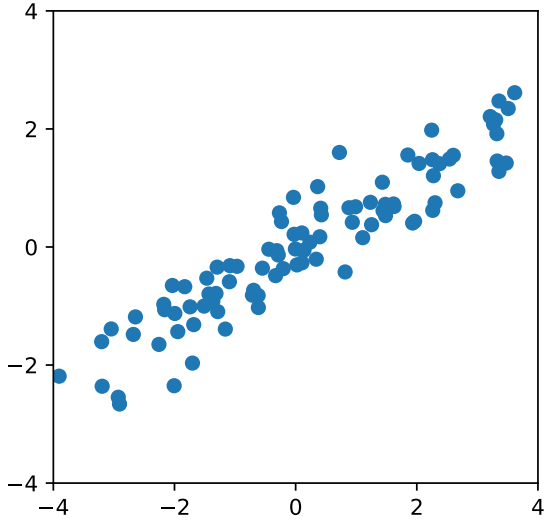
  where μ is the **mean** and Σ the **covariance matrix**
- Density if and only if the covariance matrix is **invertible**:

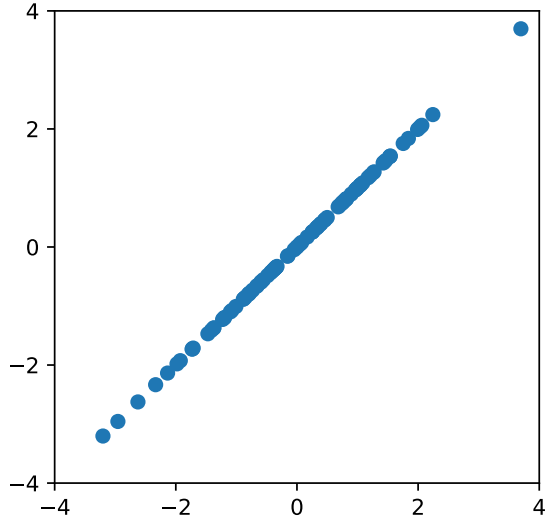
$$f(x) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

Examples









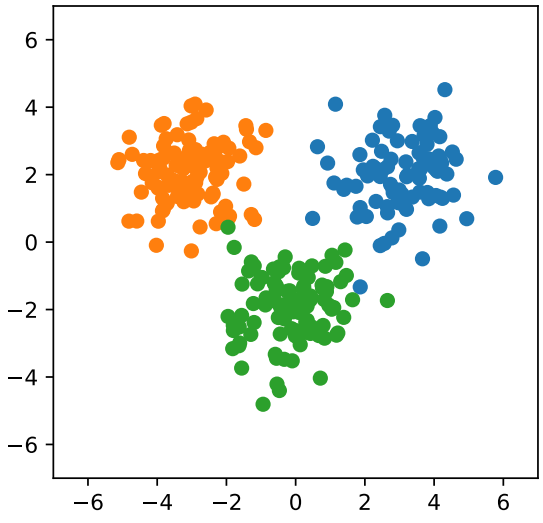
The Gaussian mixture model

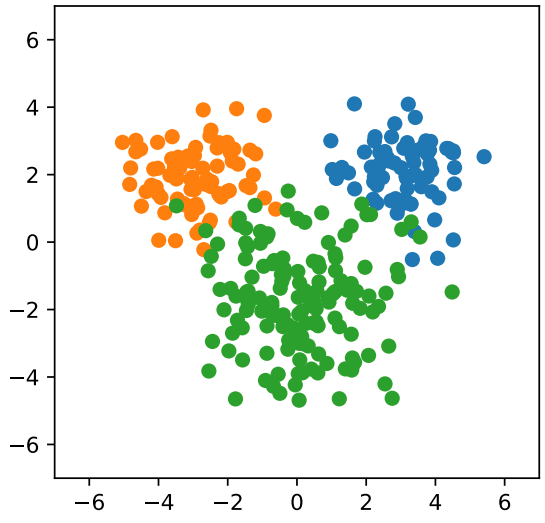
- Random choice between k Gaussian distributions
- Parameter $\theta = (\pi, \mu, \Sigma)$ where:
  $\pi = (\pi_1, \ldots, \pi_k)$ is the **mixing distribution**
  $\mu = (\mu_1, \ldots, \mu_k)$ is the set of **means**
  $\Sigma = (\Sigma_1, \ldots, \Sigma_k)$ is the set of **covariance matrices**
- Density:

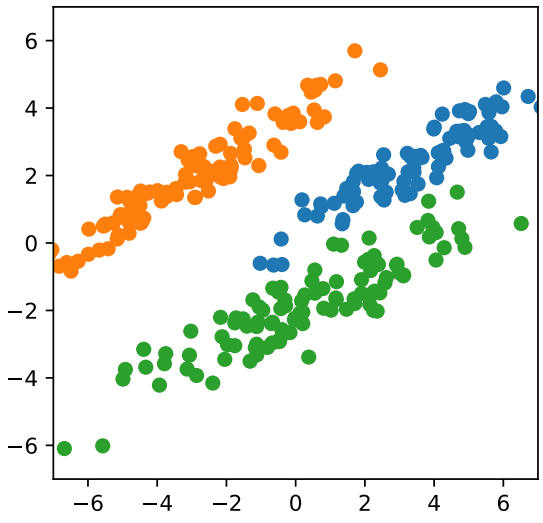
$$p_\theta(x) = \sum_{j=1}^{k} \pi_j f_j(x)$$

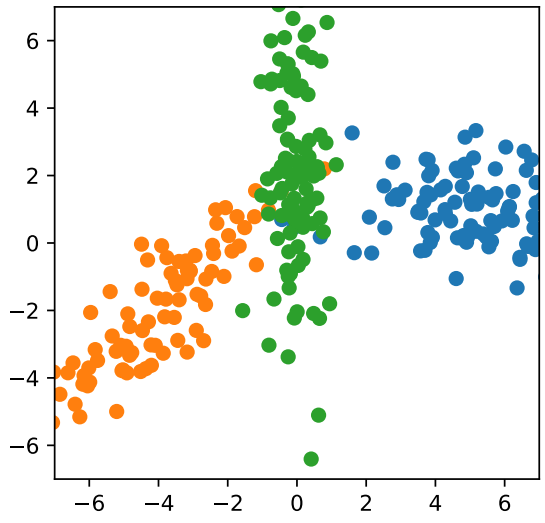
  where $f_1, \ldots, f_k$ are the densities of the k Gaussian distributions (assuming these exist)

Examples


6
4
2
0
−2
−4
−6
−6 −4 −2 0 2 4 6


6
4
2
0
−2
−4
−6
−6 −4 −2 0 2 4 6


6
4
2
0
−2
−4
−6
−6 −4 −2 0 2 4 6


6
4
2
0
−2
−4
−6
−6 −4 −2 0 2 4 6

Maximum likelihood

Consider n i.i.d. samples $x_1, \ldots, x_n$

- Likelihood:

$$p_\theta(x) = \prod_{i=1}^{n} p_\theta(x_i)$$

- Log-likelihood:

$$\ell(\theta) = \log p_\theta(x) = \sum_{i=1}^{n} \log p_\theta(x_i) = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{k} \pi_j f_j(x_i) \right)$$

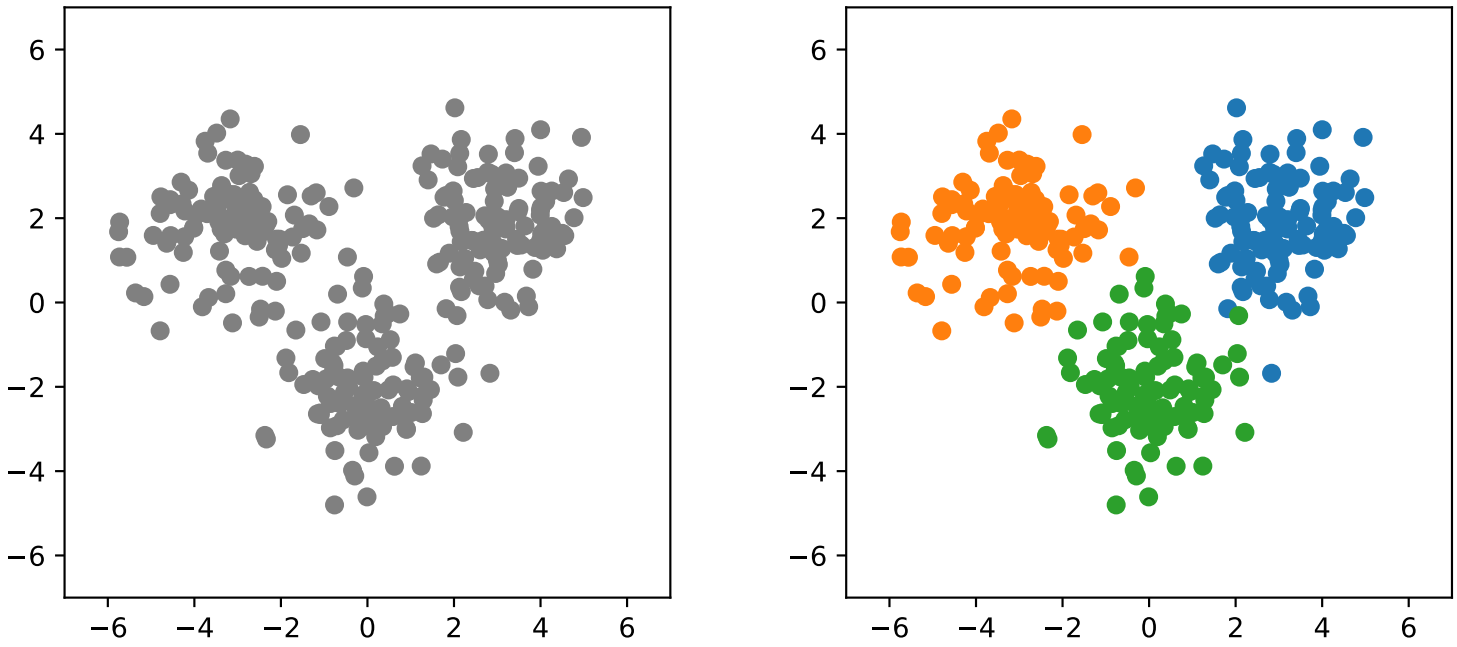
- Maximum likelihood:

$$\theta^\star = \arg\max_\theta \ell(\theta)$$

Hard to solve in practice, because the function $\theta \mapsto -\ell(\theta)$ is **not convex**

Latent variables

The color of each sample!



$$X \sim \mathcal{N}(\mu_Z, \Sigma_Z) \quad \text{with} \quad Z \sim \pi$$

Known latent variables

Data samples $x_1, \ldots, x_n$, with latent variables $z_1, \ldots, z_n$

- Joint distribution:

$$p_\theta(x, z) = p_\theta(z) p_\theta(x|z)$$

with

$$p_\theta(z) = \prod_{i=1}^{n} \pi_{z_i} \quad \text{and} \quad p_\theta(x|z) = \prod_{i=1}^{n} f_{z_i}(x_i).$$

- Log-likelihood:

$$\ell(\theta;\ z) = \log p_\theta(x, z) = \sum_{i=1}^{n} \log \pi_{z_i} + \sum_{i=1}^{n} \log f_{z_i}(x_i)$$

Maximum likelihood

$$\hat{\theta} = \arg\max_{\theta} \ell(\theta;\ z)$$

with

$$\ell(\theta;\ z) = \sum_{i=1}^{n} \log \pi_{z_i} + \sum_{i=1}^{n} \log f_{z_i}(x_i)$$

Solution

Let $n_j = \sum_{i=1}^{n} 1_{\{z_i=j\}}$ be the number of samples in cluster j:

$$\hat{\pi}_j = \frac{n_j}{n}$$

$$\hat{\mu}_j = \frac{1}{n_j} \sum_{i=1}^{n} 1_{\{z_i=j\}} x_i$$

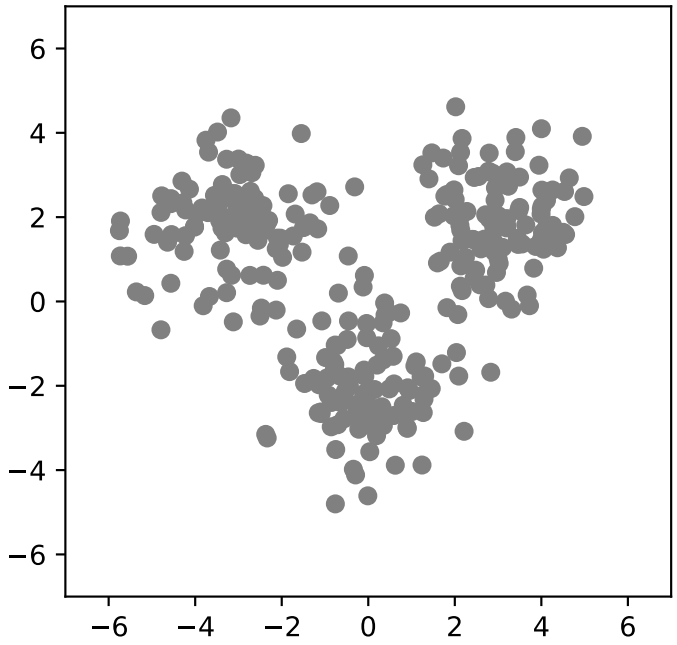
$$\hat{\Sigma}_j = \frac{1}{n_j} \sum_{i=1}^{n} 1_{\{z_i=j\}} (x_i - \hat{\mu}_j)(x_i - \hat{\mu}_j)^T$$

Estimation of the latent variables

$$p_\theta(z|x) = \frac{p_\theta(x,z)}{p_\theta(x)} \propto p_\theta(x,z) = \prod_{i=1}^{n} \pi_{z_i} f_{z_i}(x_i)$$

- Given some parameter θ, the **probability** that sample i comes from Gaussian distribution j is

$$p_{ij} \propto \pi_j f_j(x_i)$$



Expectation-Maximization

Algorithm

Starting from any parameter θ:

- **Expectation:** The **expected** likelihood is:

$$\sum_{j=1}^{k}\sum_{i=1}^{n} p_{ij}\left(\log \pi_j + \log f_j(x_i)\right) \quad \text{with} \quad p_{ij} \propto \pi_j f_j(x_i)$$

- **Maximization:** This is **maximum** for $\theta = \hat{\theta}$ with:

$$\hat{\pi}_j = \frac{n_j}{n} \quad n_j = \sum_{i=1}^{n} p_{ij}$$

$$\hat{\mu}_j = \frac{1}{n_j}\sum_{i=1}^{n} p_{ij} x_i \quad \hat{\Sigma}_j = \frac{1}{n_j}\sum_{i=1}^{n} p_{ij}(x_i - \hat{\mu}_j)(x_i - \hat{\mu}_j)^T$$

- Iterate

The symmetric Gaussian mixture model

Simplified model with:

- **Uniform** random choice between k Gaussian distributions
- Covariance matrices $\Sigma = (\sigma^2 I, \ldots, \sigma^2 I)$ for some **fixed** σ

The only parameter is $\theta = \mu = (\mu_1, \ldots, \mu_k)$, the set of **means**

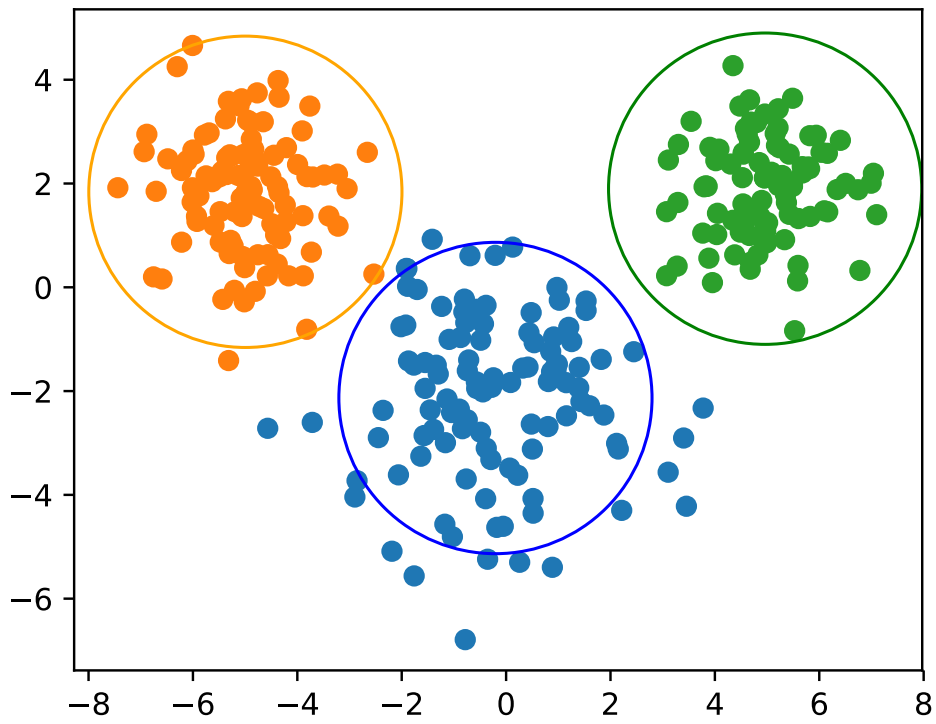
Density:

$$p_\theta(x) = \frac{1}{K} \sum_{j=1}^{k} f_j(x)$$

with

$$f_j(x) = \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} e^{-\frac{||x-\mu_j||^2}{2\sigma^2}}$$

Example in $\mathbb{R}^2$



Expectation-Maximization

Algorithm

Starting from any parameter μ:

- **Expectation:** The **expected** likelihood is:

$$\sum_{j=1}^{k}\sum_{i=1}^{n} p_{ij} \log f_j(x_i) \quad \text{with} \quad p_{ij} \propto f_j(x_i)$$

- **Maximization:** This is **maximum** for $\mu = \hat{\mu}$ with:

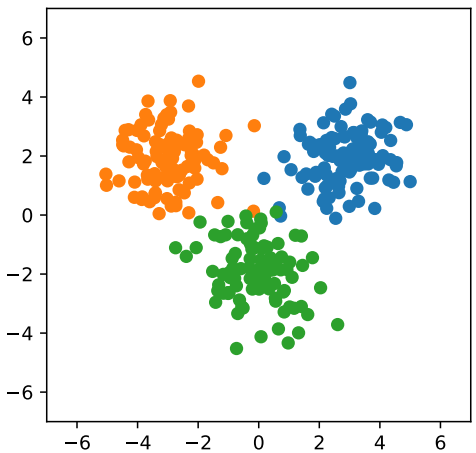
$$\hat{\mu}_j = \frac{1}{n_j}\sum_{i=1}^{n} p_{ij} x_i \quad n_j = \sum_{i=1}^{n} p_{ij}$$

- Iterate

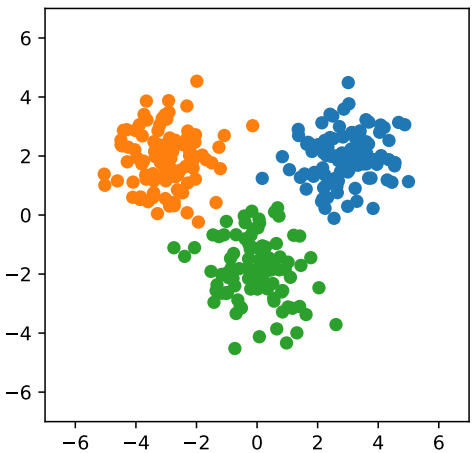
This is **soft** k-**means**! When $\sigma \to 0$, this is k-means

Example 1

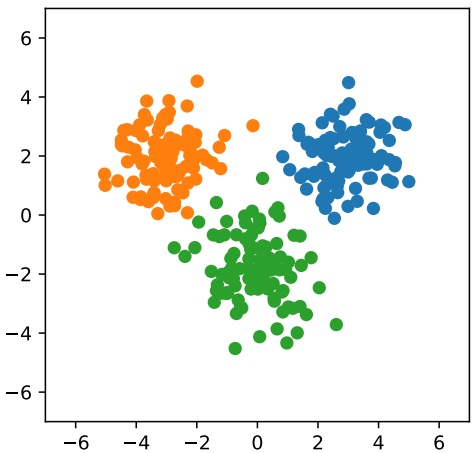
Data



k-means

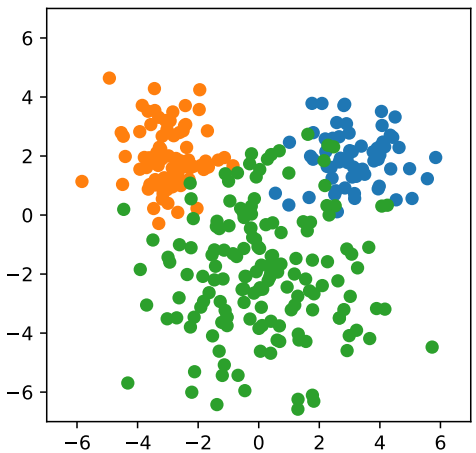


Gaussian mixture

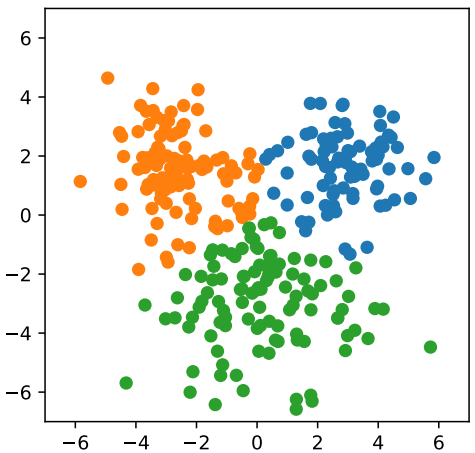


Example 2

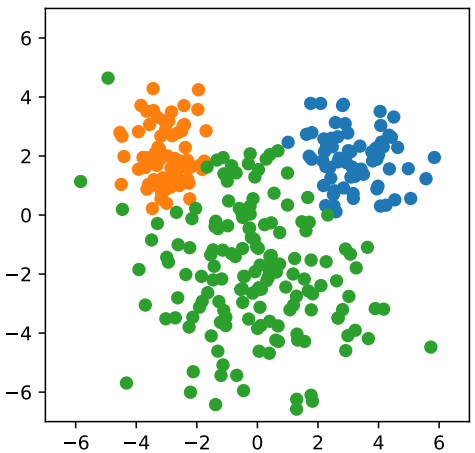
Data



k-means

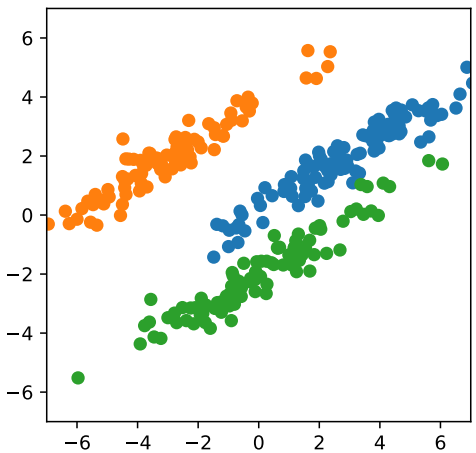


Gaussian mixture

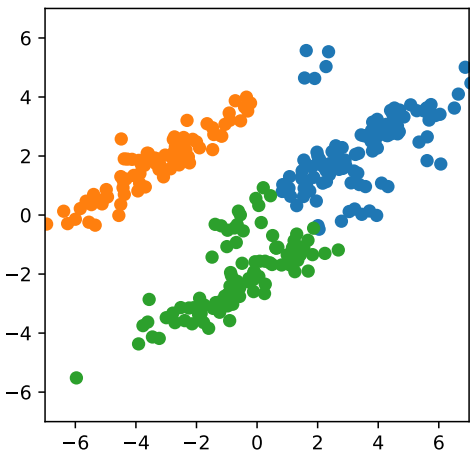


Example 3

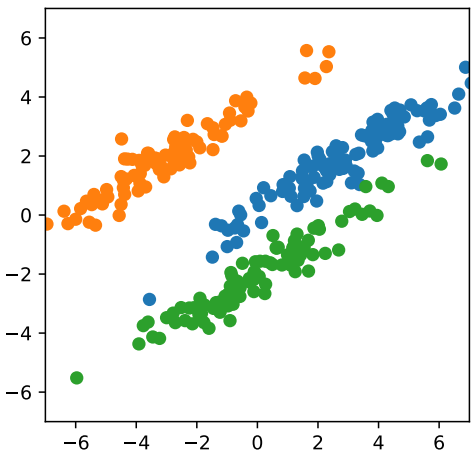
Data



k-means

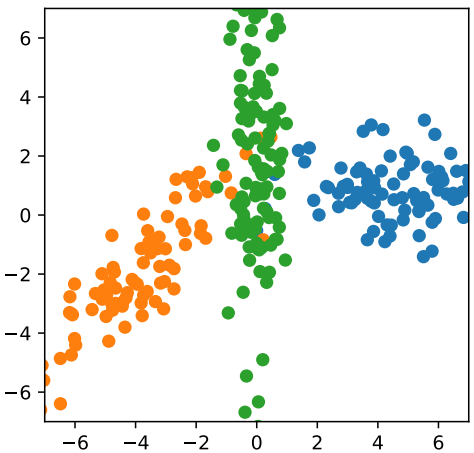


Gaussian mixture

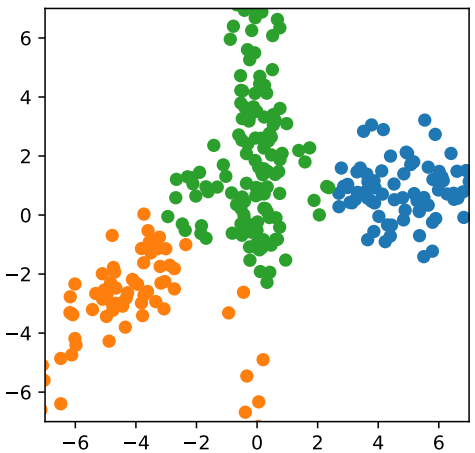


Example 4

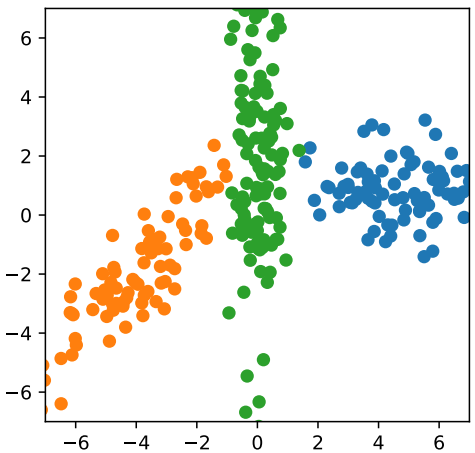
Data



k-means



Gaussian mixture



Hierarchical Clustering

Thomas Bonald

2019 – 2020

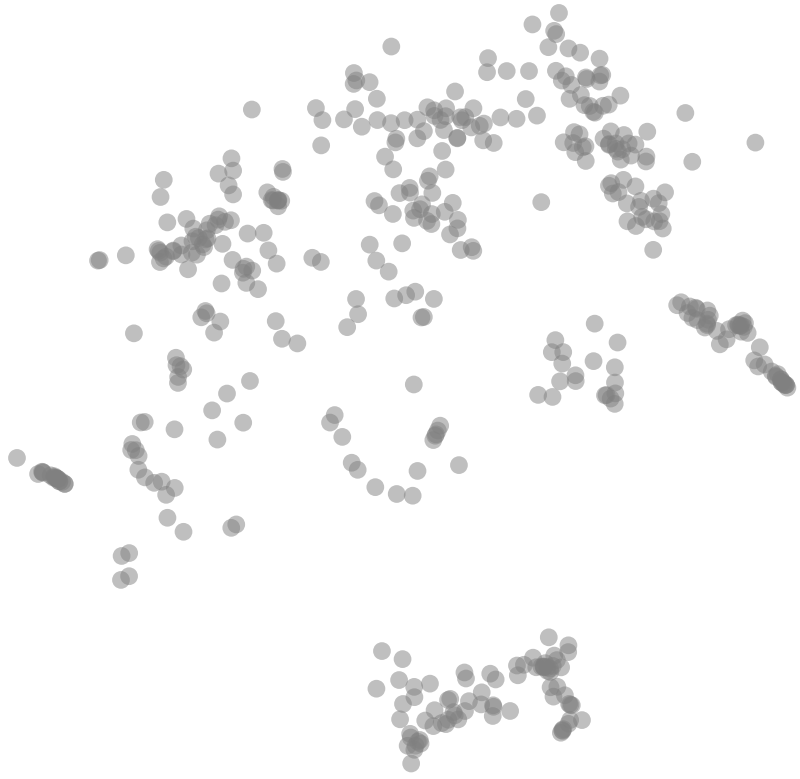
Introduction

Let $x_1, \ldots, x_n$ be n data samples of $\mathbb{R}^d$

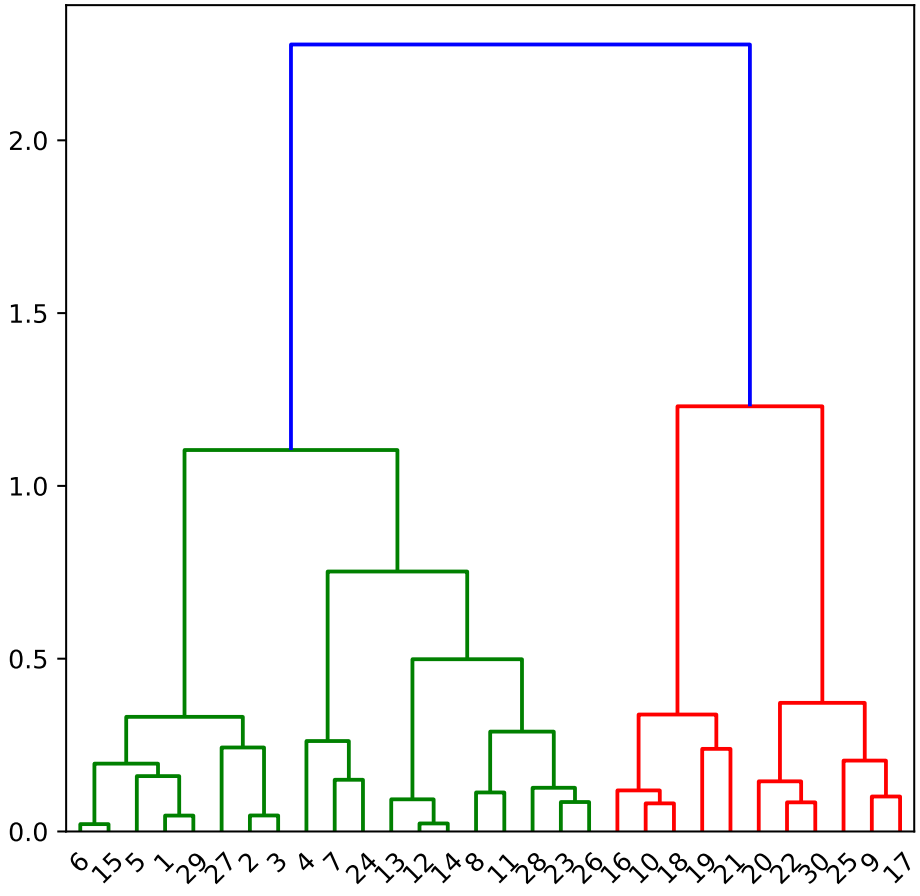
Problem

1. How to **cluster** the n data samples so that close samples (for the Euclidean distance) are in the same cluster?
2. How to reflect the **complex**, **multi-scale** nature of data?

Example in $\mathbb{R}^2$



Hierarchical clustering



A dendrogram ($n = 30$)

Outline

1. Divisive approach
2. Agglomerative approach
3. Distance between clusters
4. Update formulas
5. Nearest neighbor chain

Divisive approach (top-down)

- Start from a single cluster
- Split this cluster recursively (e.g., best cut)

Algorithm

$\mathcal{C} \leftarrow \{\{1, \ldots, n\}\}$
While $\mathcal{C}$ contains a cluster C with at least two nodes:

- $A, B \leftarrow$ best cut of C
- $\mathcal{C} \leftarrow \mathcal{C} \setminus \{C\}$
- $\mathcal{C} \leftarrow \mathcal{C} \cup \{A, B\}$
- Output $A, B, d(A, B)$

Agglomerative approach (bottom-up)

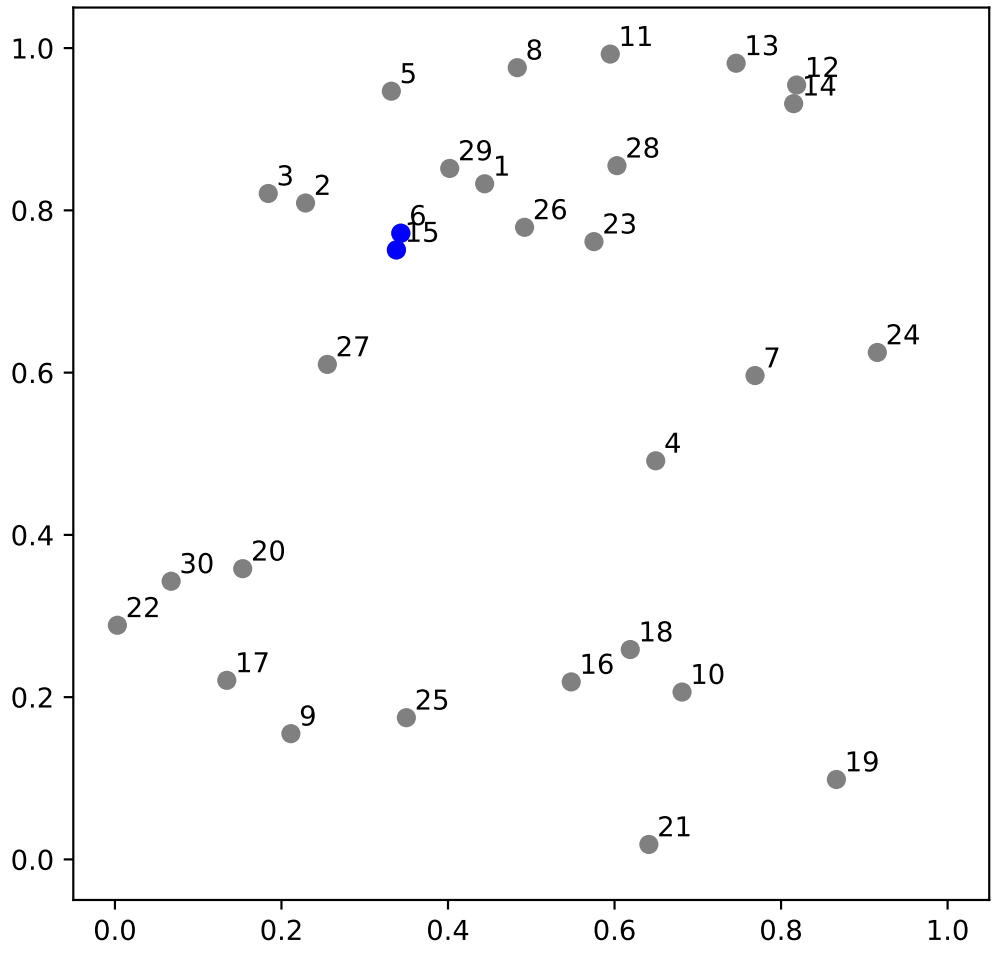
- Start from individual clusters
- Merge iteratively the **two closest** clusters

Algorithm

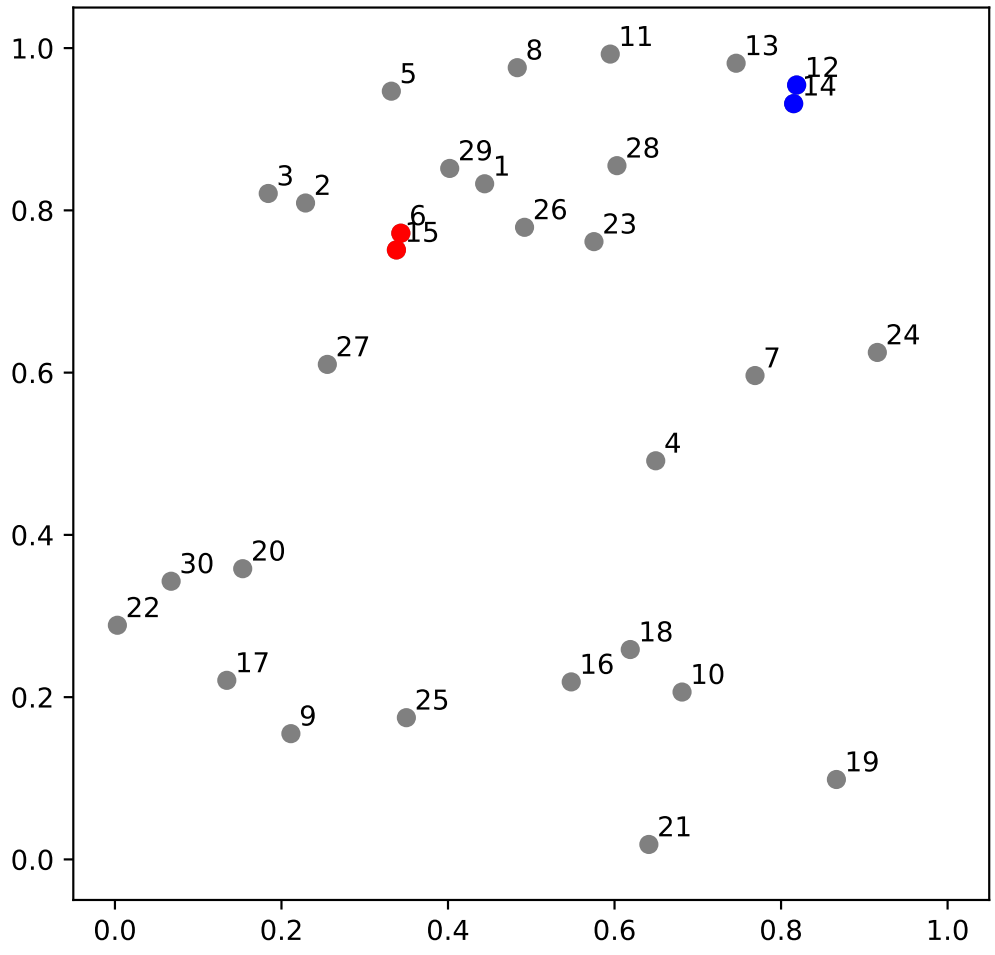
$\mathcal{C} \leftarrow \{\{1\}, \ldots, \{n\}\}$
For $t = 1, \ldots, n-1$:

- $A, B = \arg\min_{a,b \in \mathcal{C}, a \neq b} d(a, b)$
- $C \leftarrow A \cup B$
- $\mathcal{C} \leftarrow \mathcal{C} \setminus \{A, B\}$
- $\mathcal{C} \leftarrow \mathcal{C} \cup \{C\}$
- Output $A, B, d(A, B)$

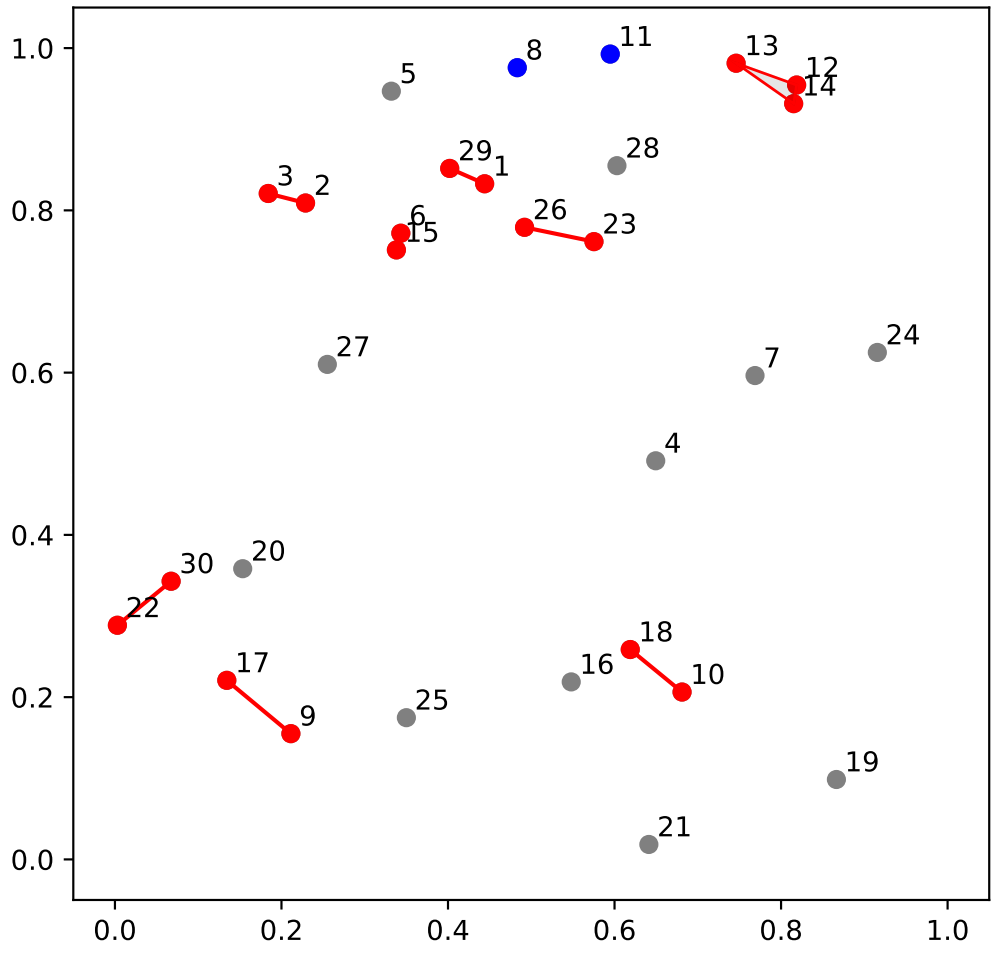
Example ($t = 1$)


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
11
8
13
5
12
14
29
1
28
3
2
6
26
15
23
27
24
7
4
30
20
22
18
17
16
10
25
9
19
21

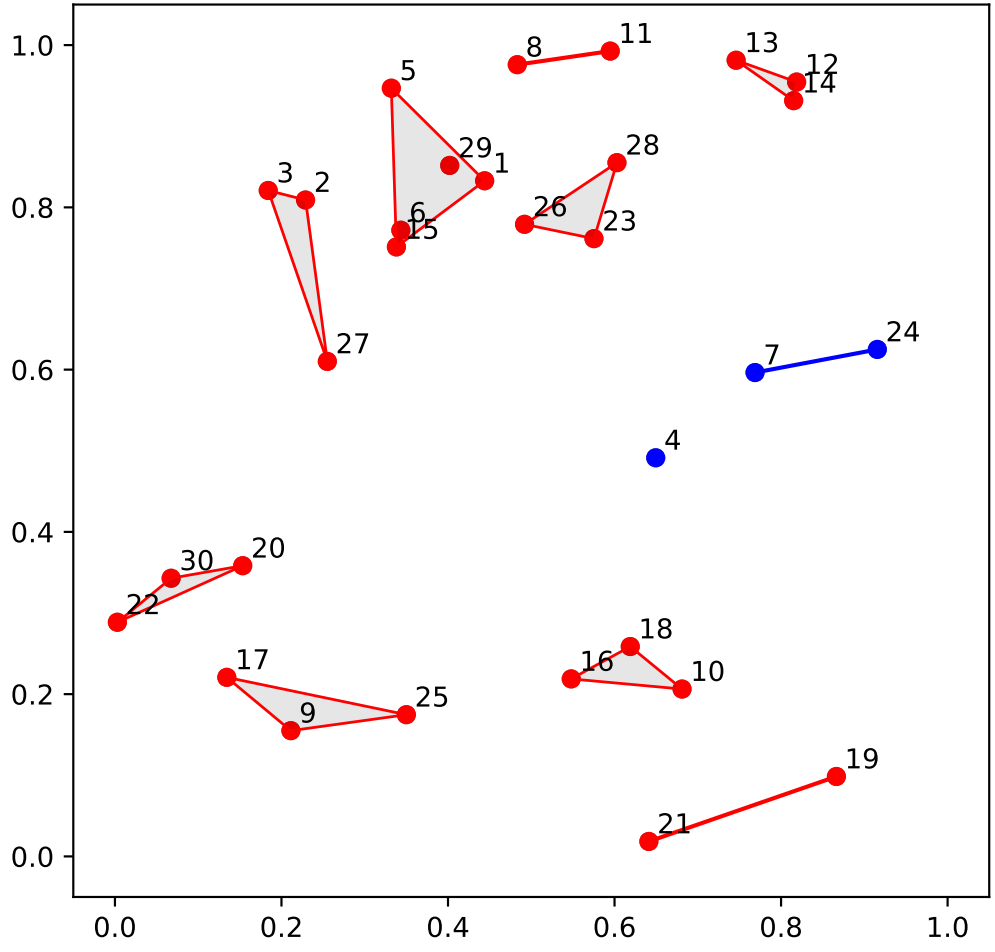
Example ($t = 2$)


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
11
8
13
5
12
14
29
1
28
3
2
6
26
15
23
27
24
7
4
30
20
22
18
17
16
10
25
9
19
21

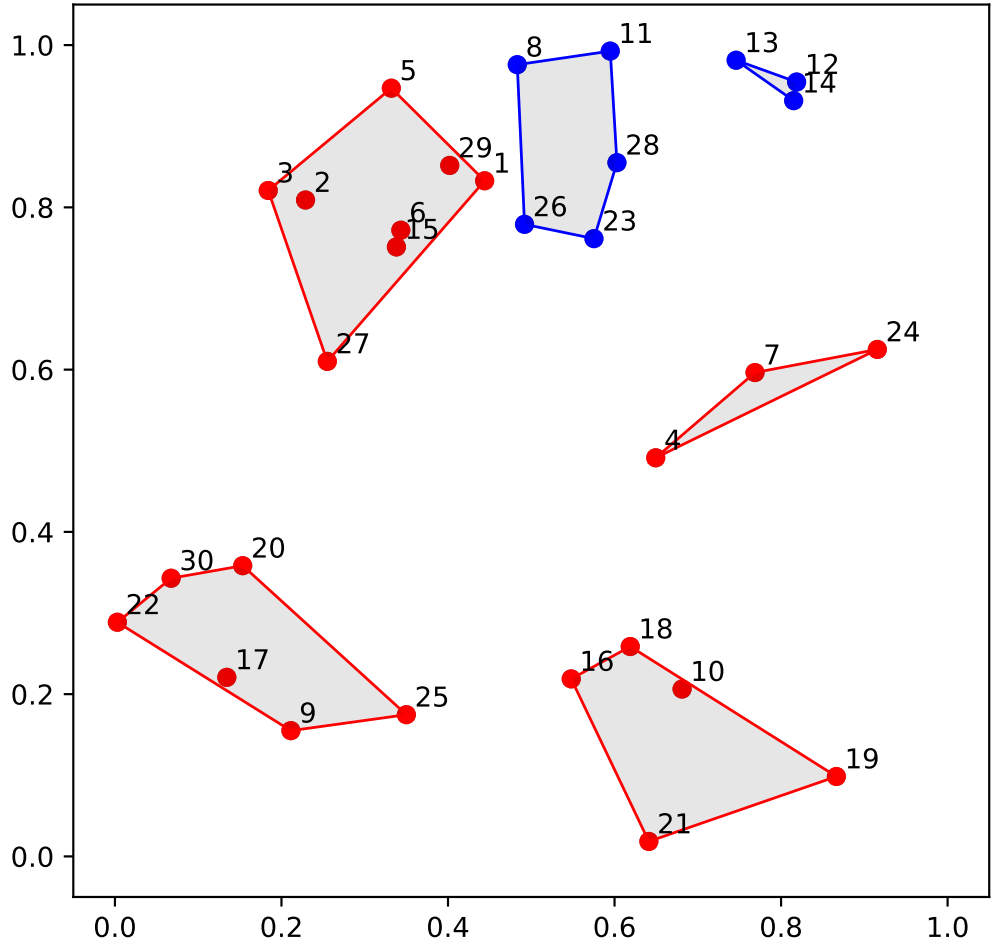
Example ($t = 10$)


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
8
11
13
12
14
5
29
1
28
3
2
6
15
26
23
27
7
24
4
30
20
22
18
17
16
10
9
25
19
21

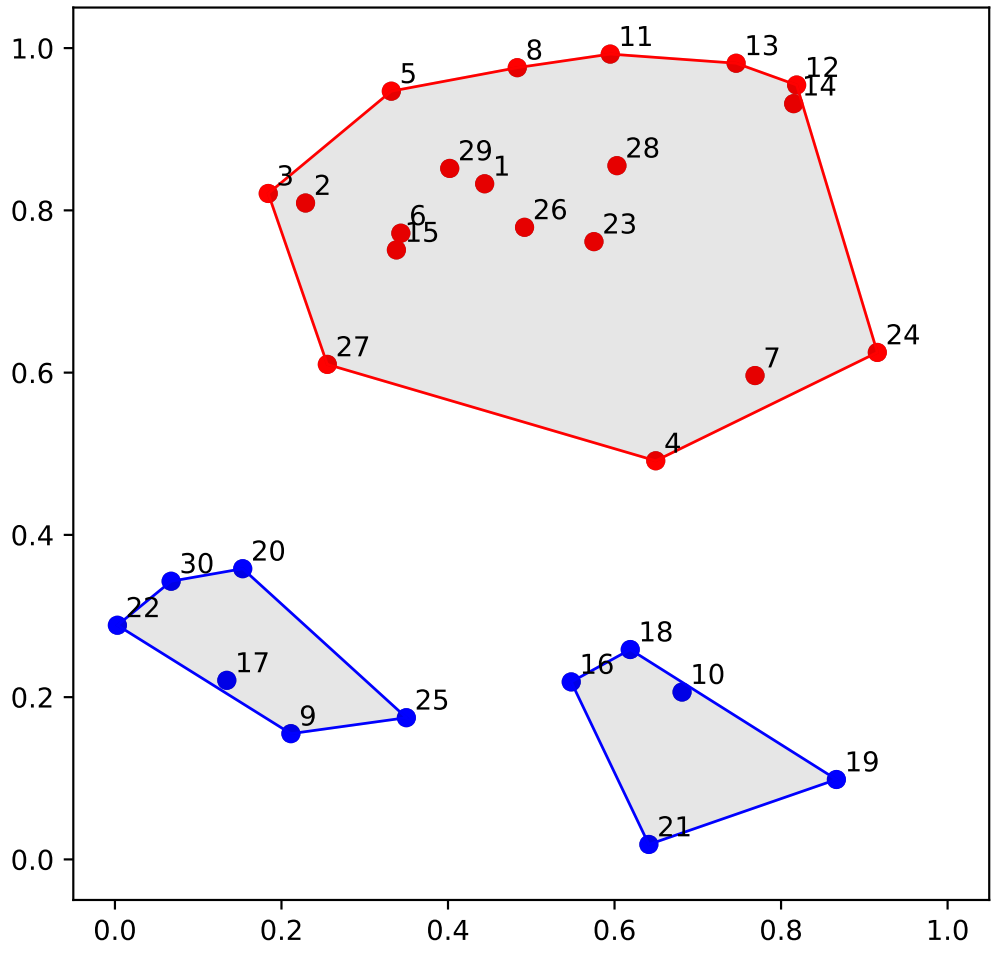
Example ($t = 20$)


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

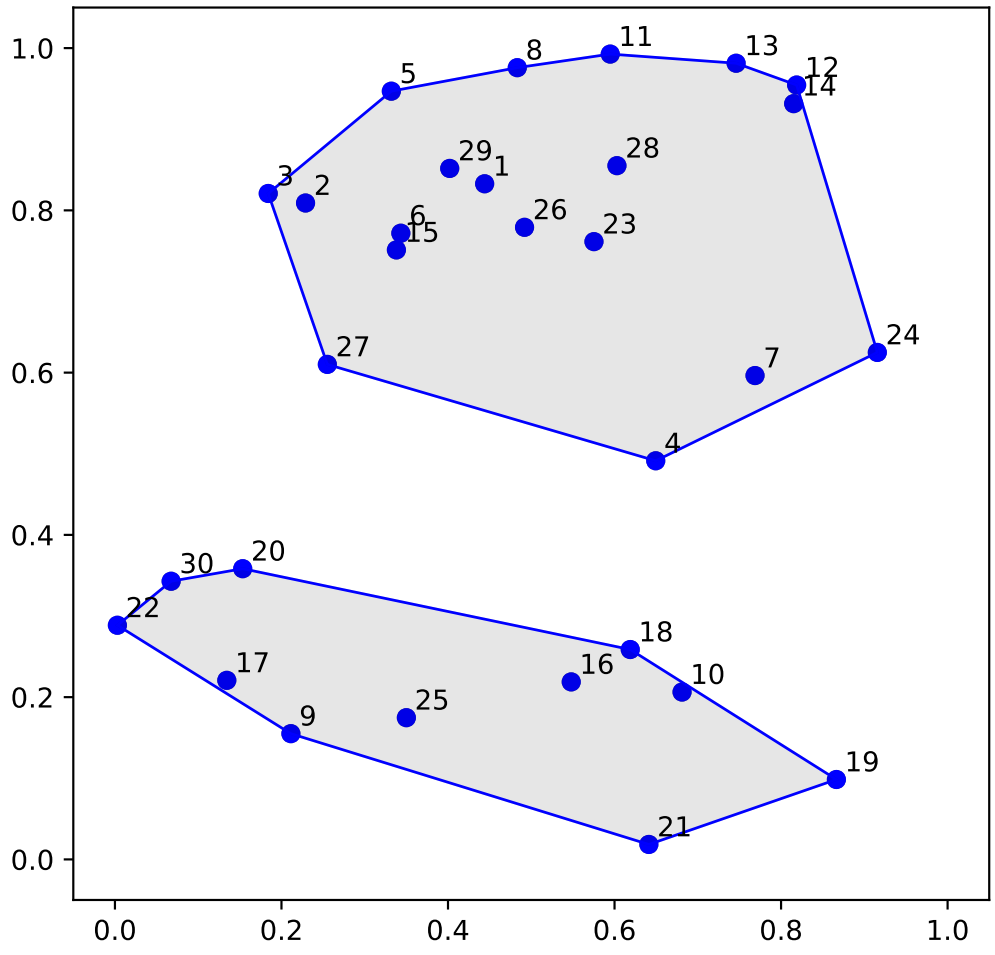
Example ($t = 25$)


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

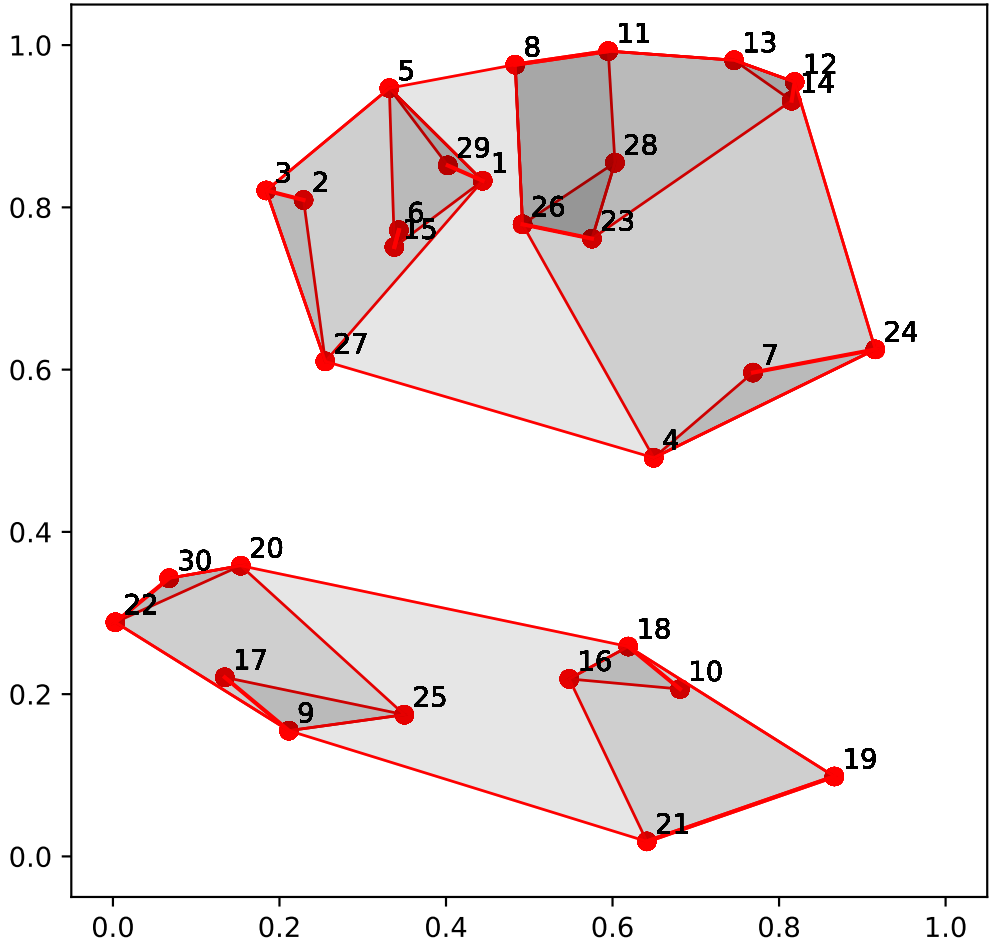
Example ($t = 28$)


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
11
8
13
5
12
14
29
1
28
3
2
6
26
23
15
27
24
7
4
30
20
22
18
17
16
10
25
9
19
21

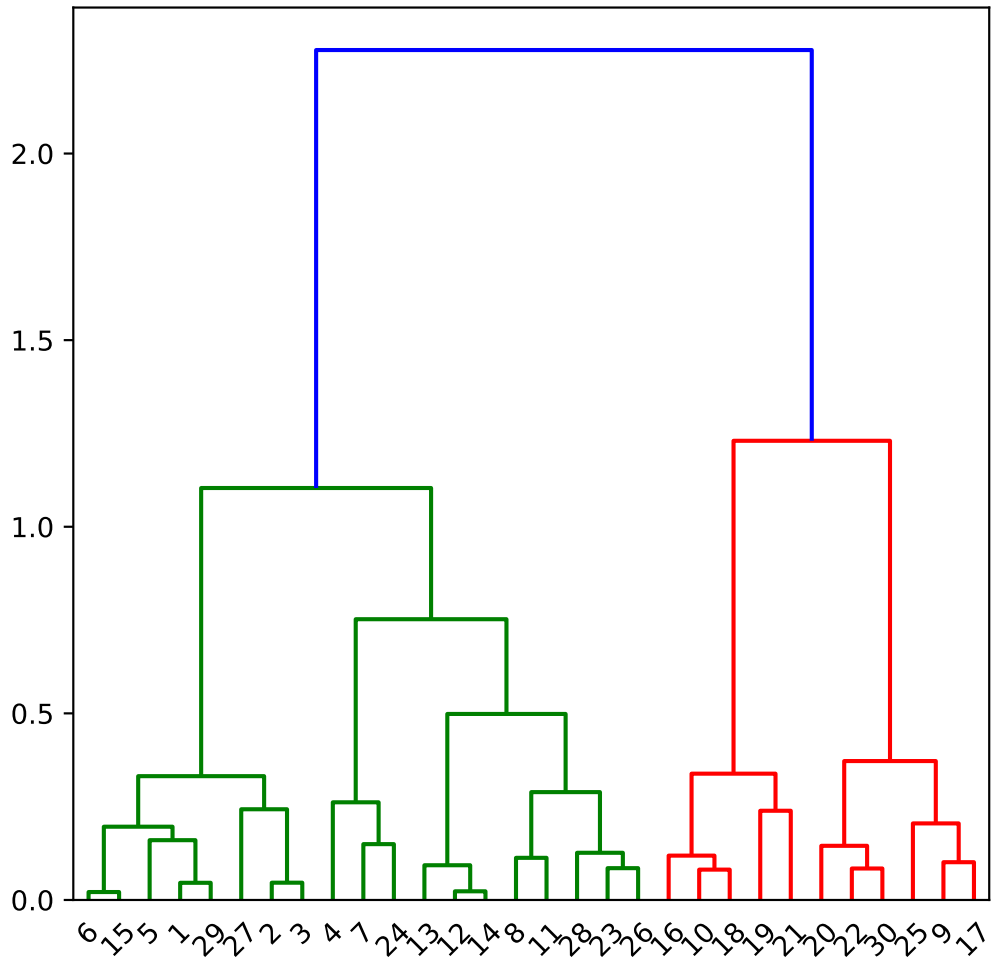
Example ($t = 29$)


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
8
11
13
12
14
5
29
1
28
3
2
6
15
26
23
27
7
24
4
30
20
22
18
17
16
10
25
9
19
21

Final clustering



Representation as a dendrogram



Distance between clusters

Minimum (single linkage)

$$d(a,b) = \min_{i \in a, j \in b} \|x_i - x_j\|$$

Maximum (complete linkage)

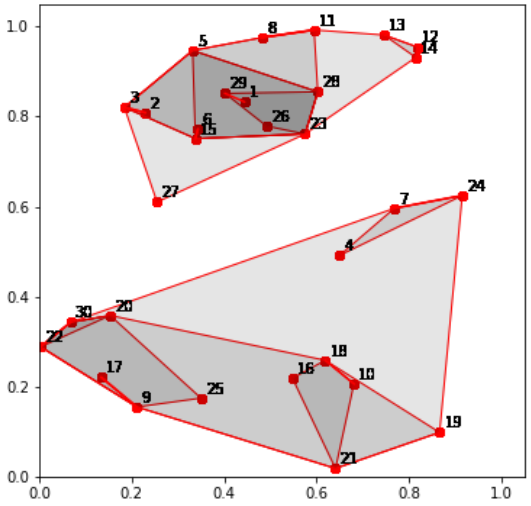
$$d(a,b) = \max_{i \in a, j \in b} \|x_i - x_j\|$$

Average (average linkage)

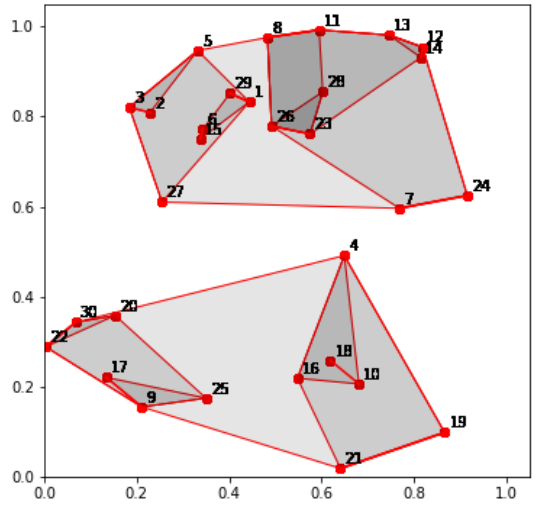
$$d(a,b) = \frac{1}{|a||b|} \sum_{i \in a, j \in b} \|x_i - x_j\|$$

Example

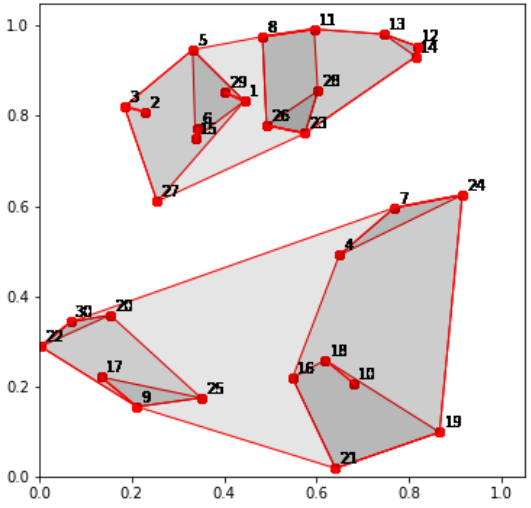
Single



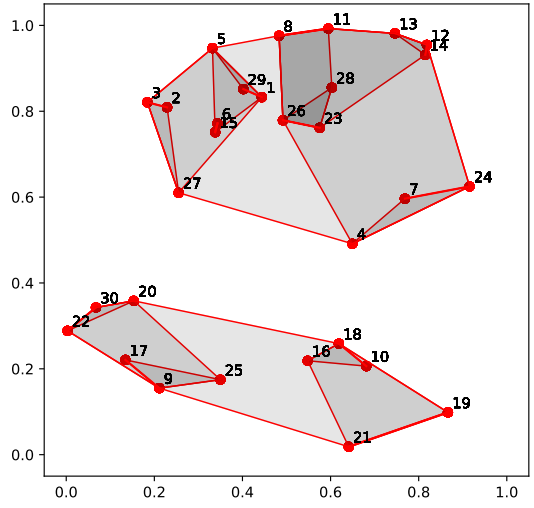
Complete



Average



Ward



Reducible distances

Proposition

The **min, max, average** distances are reducible,

$$d(a \cup b, c) \geq \min(d(a, c), d(b, c))$$

Consequence

The sequence of distances between merged clusters is **non-decreasing**.

If a, b are merged, then for any other cluster c,

$$d(a \cup b, c) \geq \min(d(a, c), d(b, c)) \geq d(a, b).$$

Ward's method

- Minimizes the sum of **square errors**,

$$S = \sum_{c \in \mathcal{C}} \underbrace{\sum_{i \in c} \|x_i - g(c)\|^2}_{S(c)}$$

where $g(c)$ is the **centroid** of cluster c,

$$g(c) = \frac{1}{|c|} \sum_{i \in c} x_i$$

and $\|\cdot\|$ is the **Euclidan norm** (cf. k-**means**)

- The distance between a and b is the **cost** of merging a and b,

$$d(a, b) = S(a \cup b) - S(a) - S(b) = \frac{|a|\ |b|}{|a| + |b|} \|g(a) - g(b)\|^2$$

Update formulas

Min/Max

$$d(a \cup b, c) = \min / \max(d(a, c), d(b, c))$$

Average

$$d(a \cup b, c) = \frac{|a|}{|a| + |b|} d(a, c) + \frac{|b|}{|a| + |b|} d(b, c)$$

Ward

$$d(a \cup b, c) = \frac{|a| + |c|}{|a| + |b| + |c|} d(a, c) + \frac{|b| + |c|}{|a| + |b| + |c|} d(b, c) - \frac{|c|}{|a| + |b| + |c|} d(a, b)$$

Nearest-neighbor chain algorithm

Hierarchical clustering in time $O(n^2)$ and space $O(n)$

Algorithm

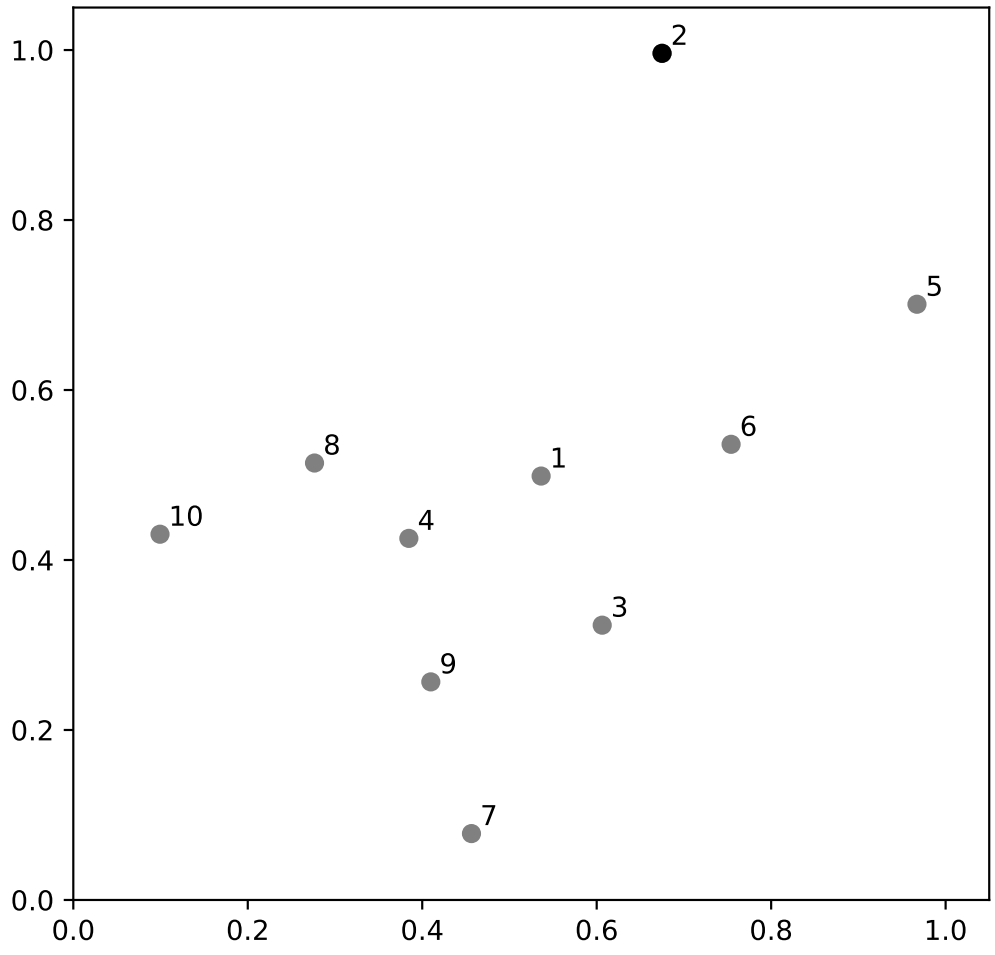
1. Start from any cluster
2. Build the (directed) **chain of nearest neighbors** until two clusters are jointly nearest neighbors
3. **Merge** these two clusters and proceed with the rest of the chain until the chain is empty
4. Go to step 1 if there are at least two clusters left

Remark

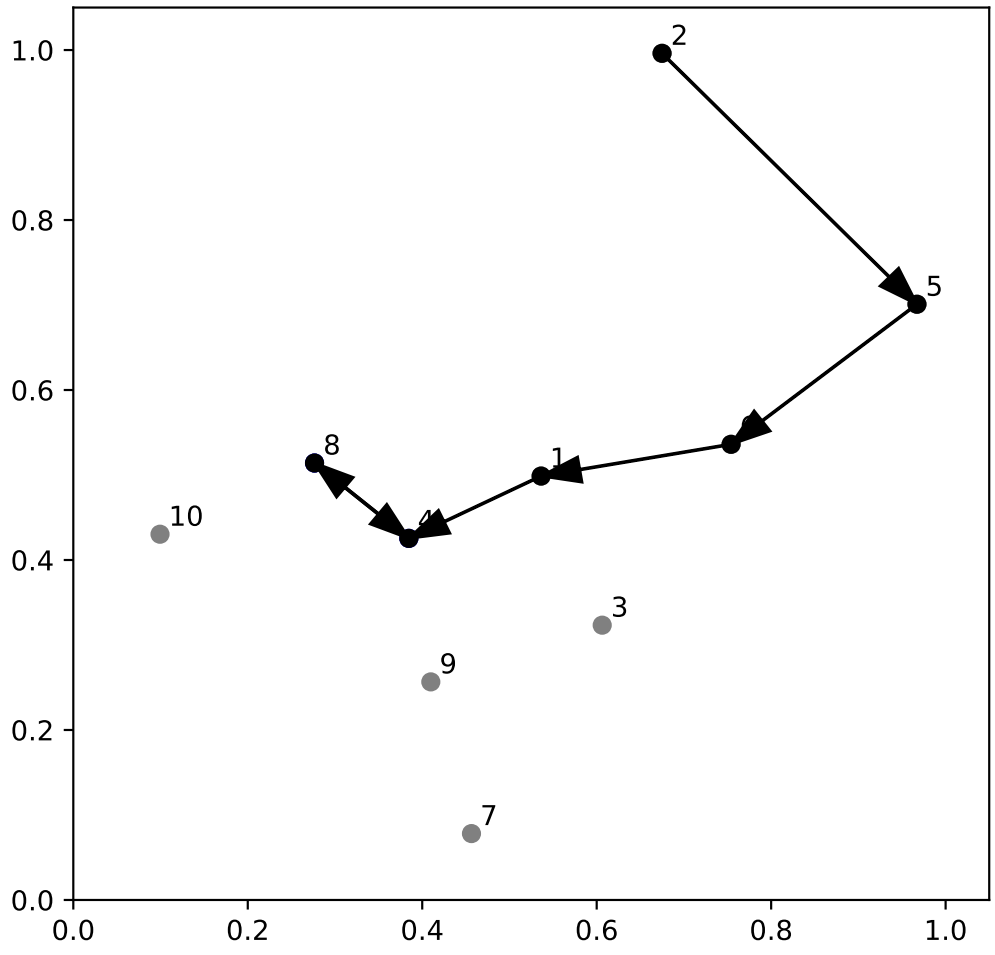
Works for a **reducible** distance,

$$d(a \cup b, c) \geq \min(d(a, c), d(b, c))$$

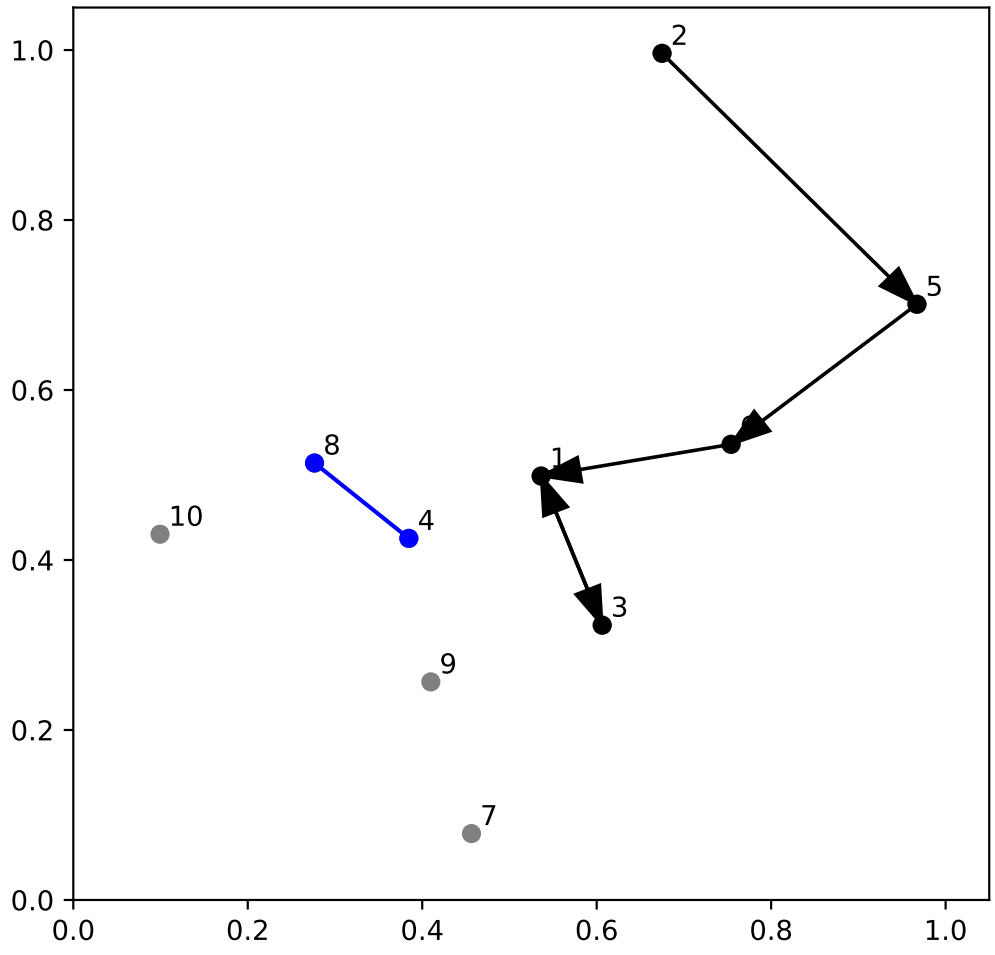
Example (complete linkage)



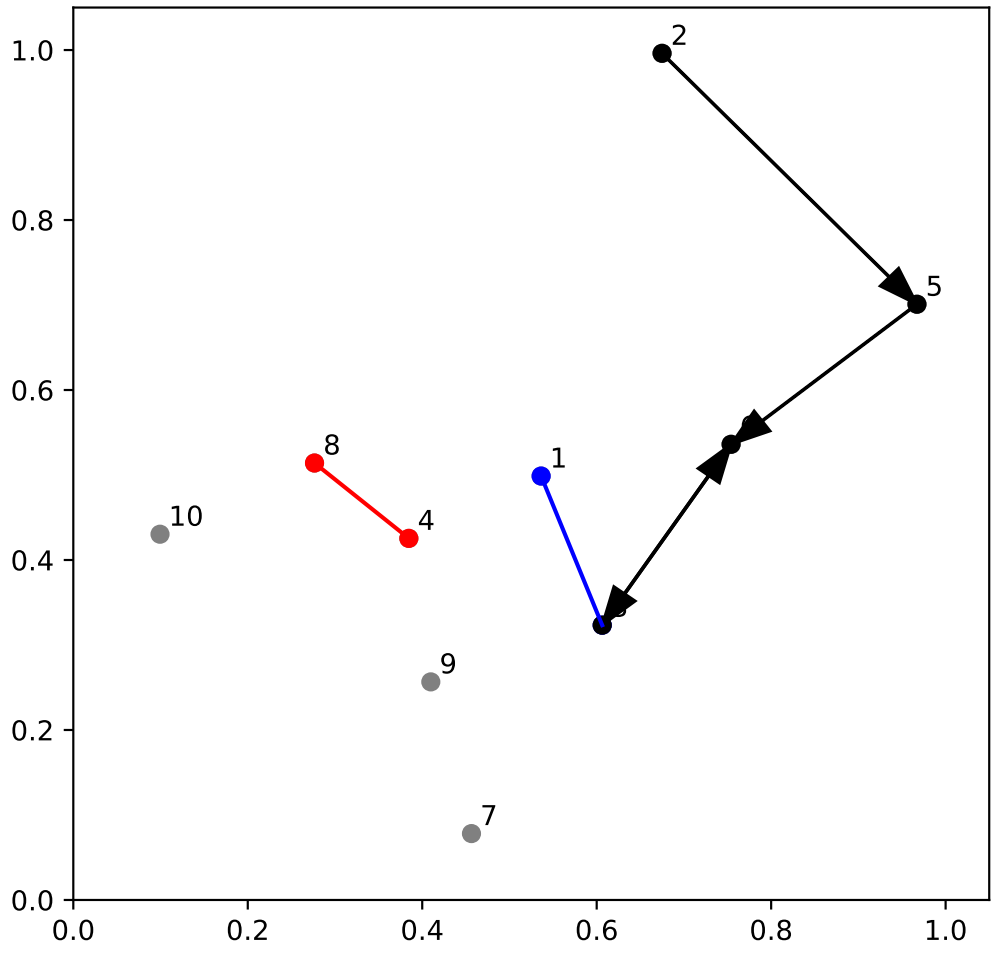
Example (complete linkage)



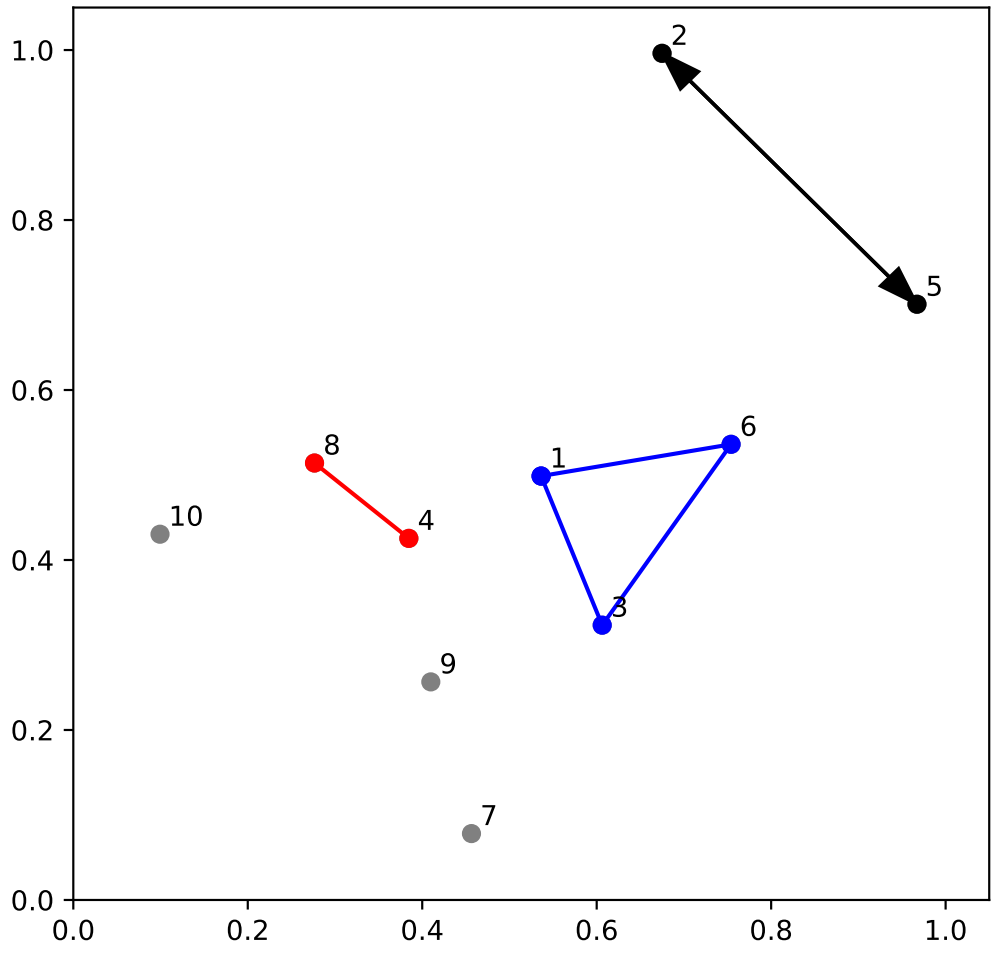
Example (complete linkage)



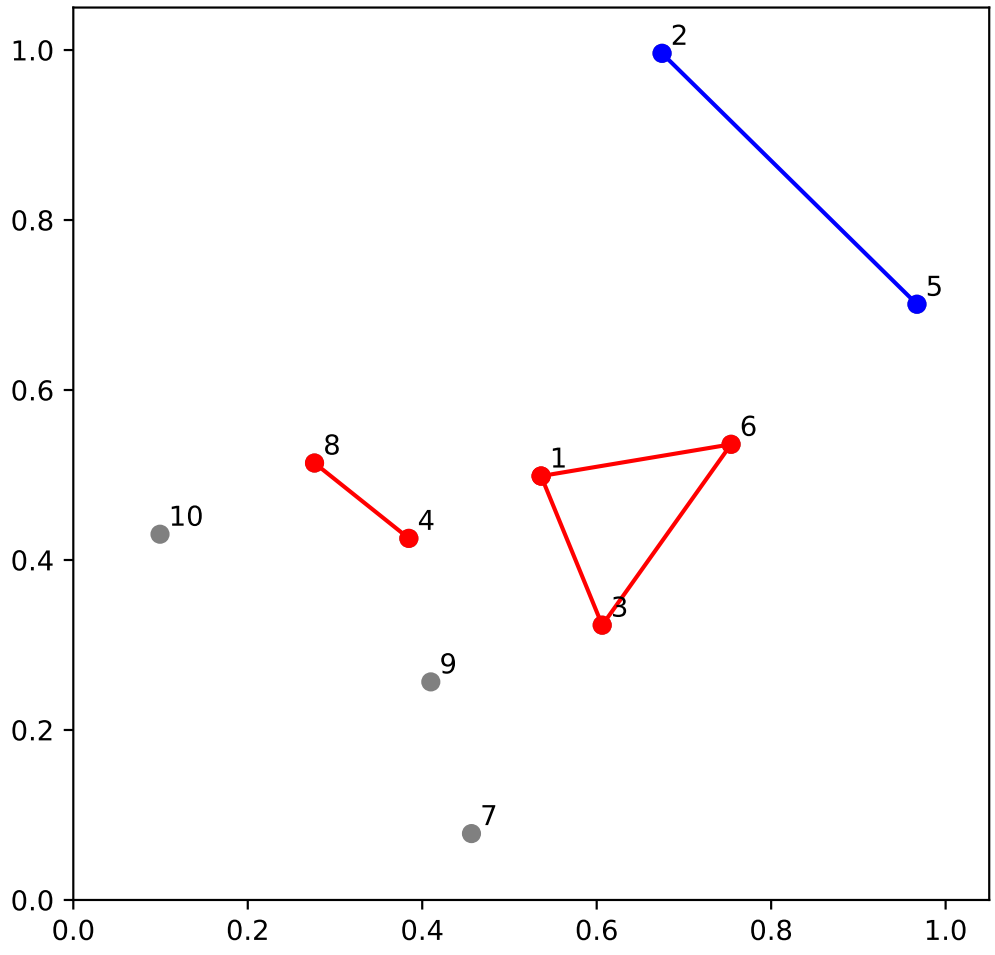
Example (complete linkage)



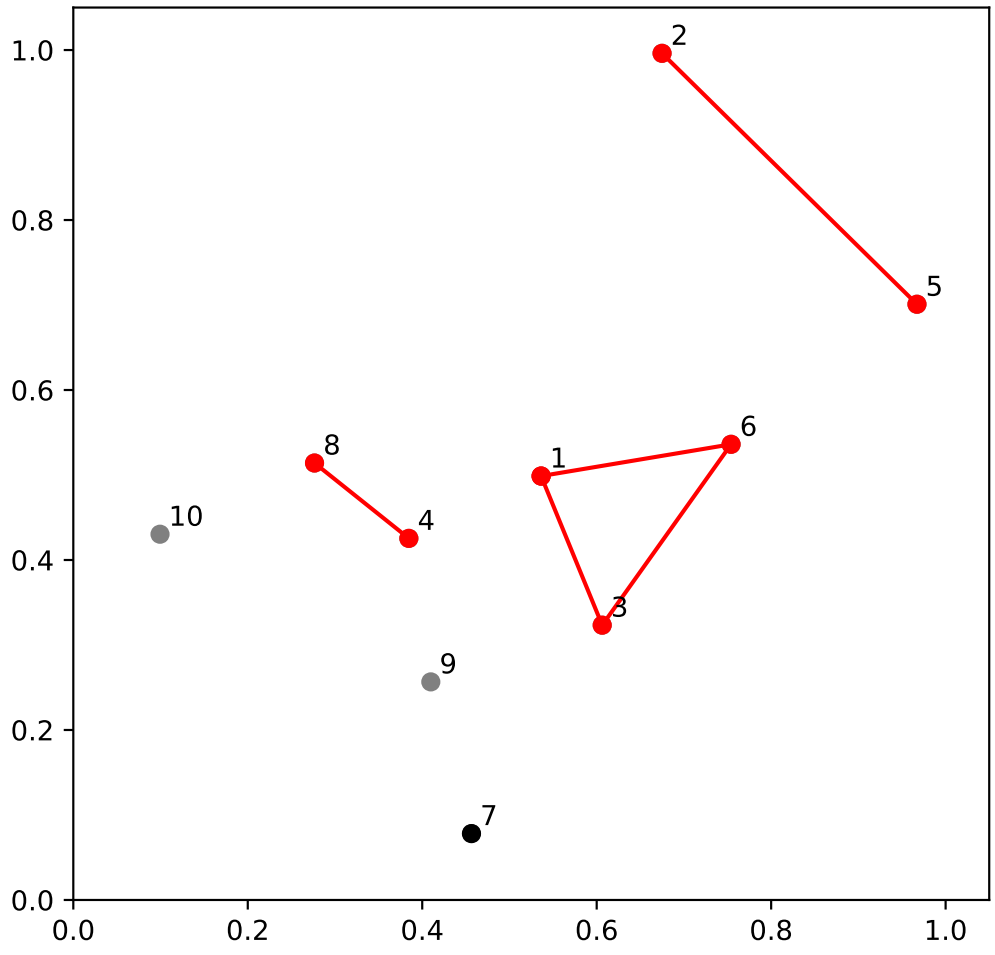
Example (complete linkage)



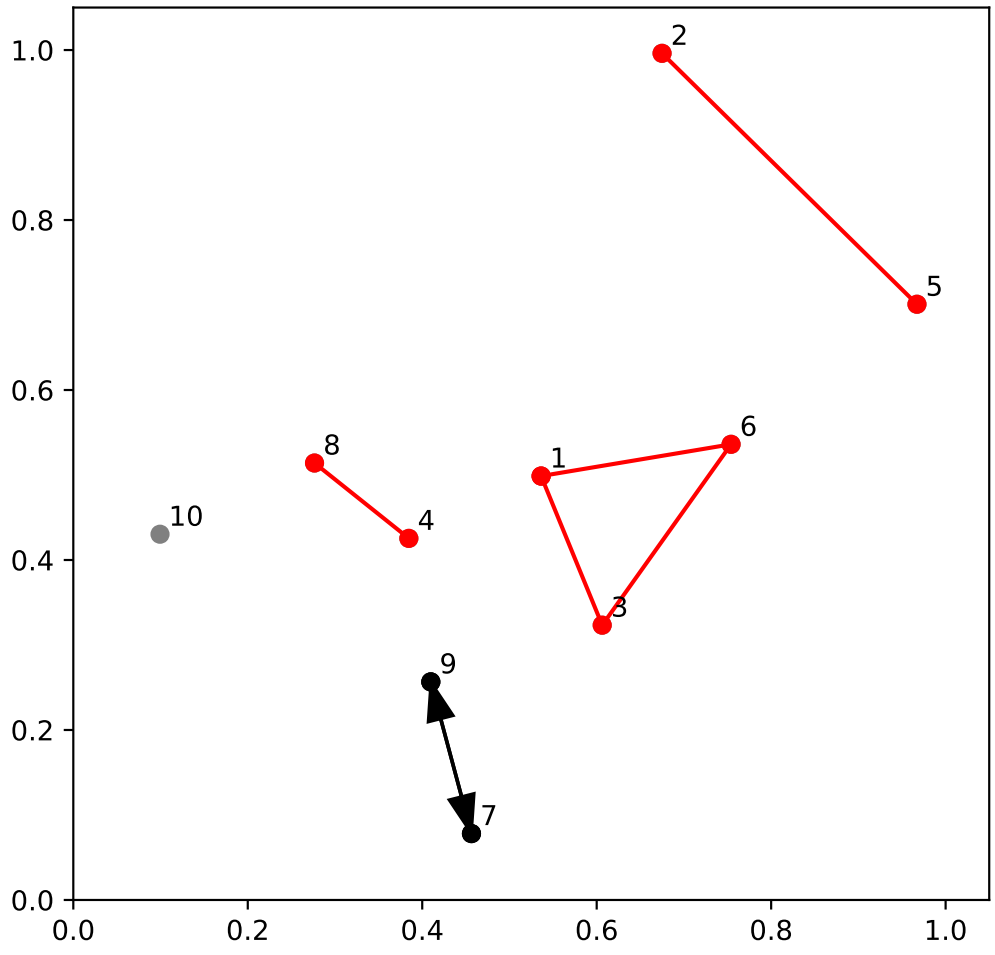
Example (complete linkage)



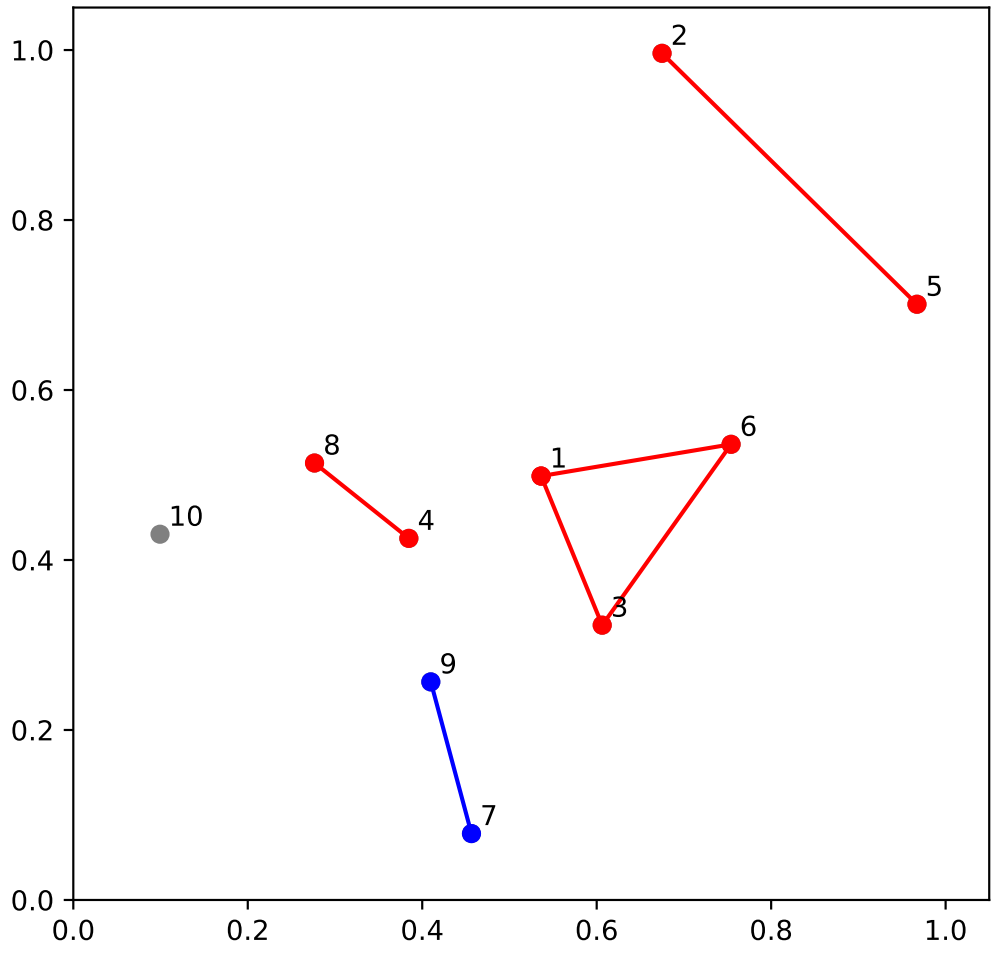
Example (complete linkage)



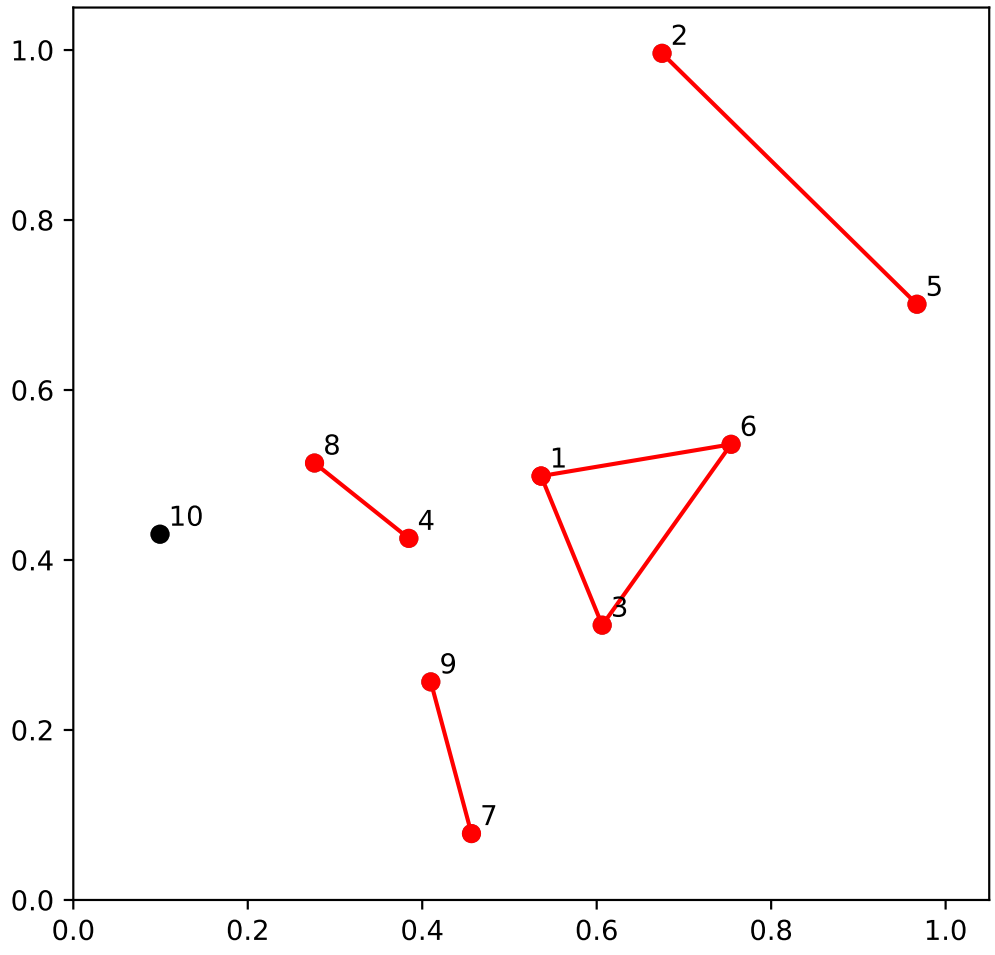
Example (complete linkage)



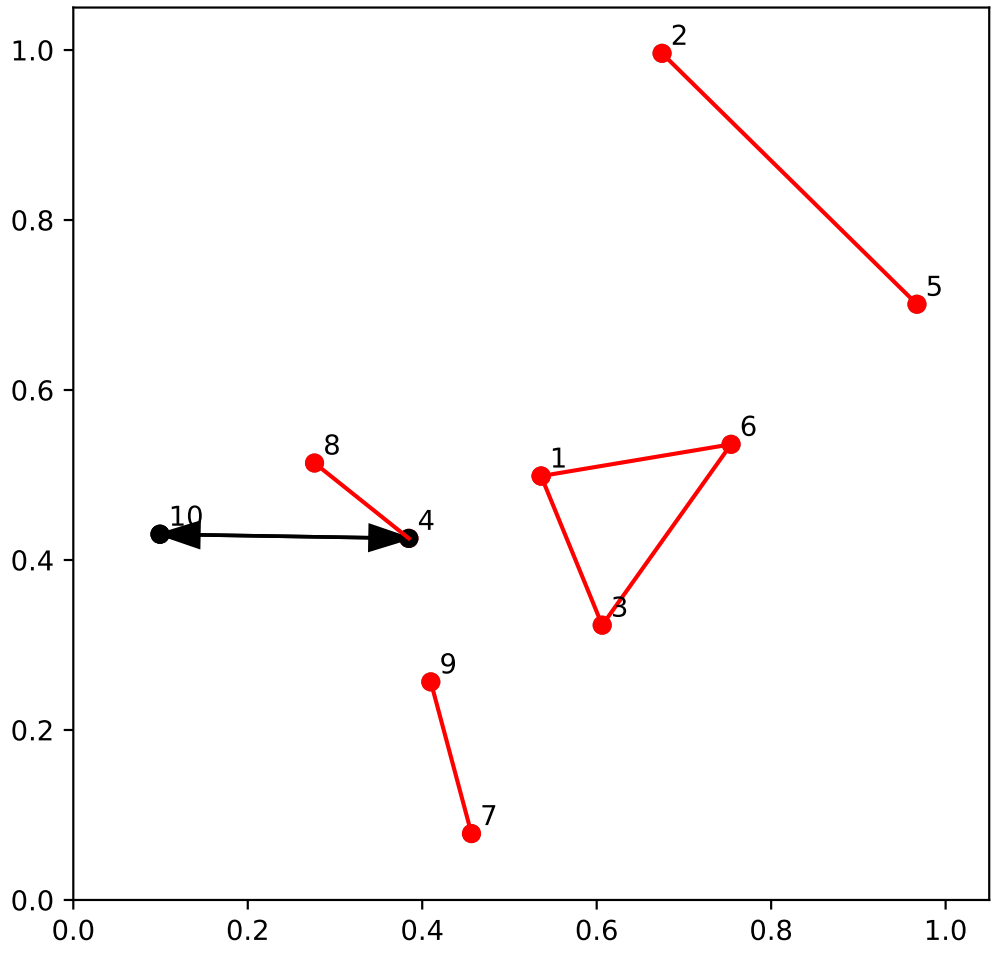
Example (complete linkage)



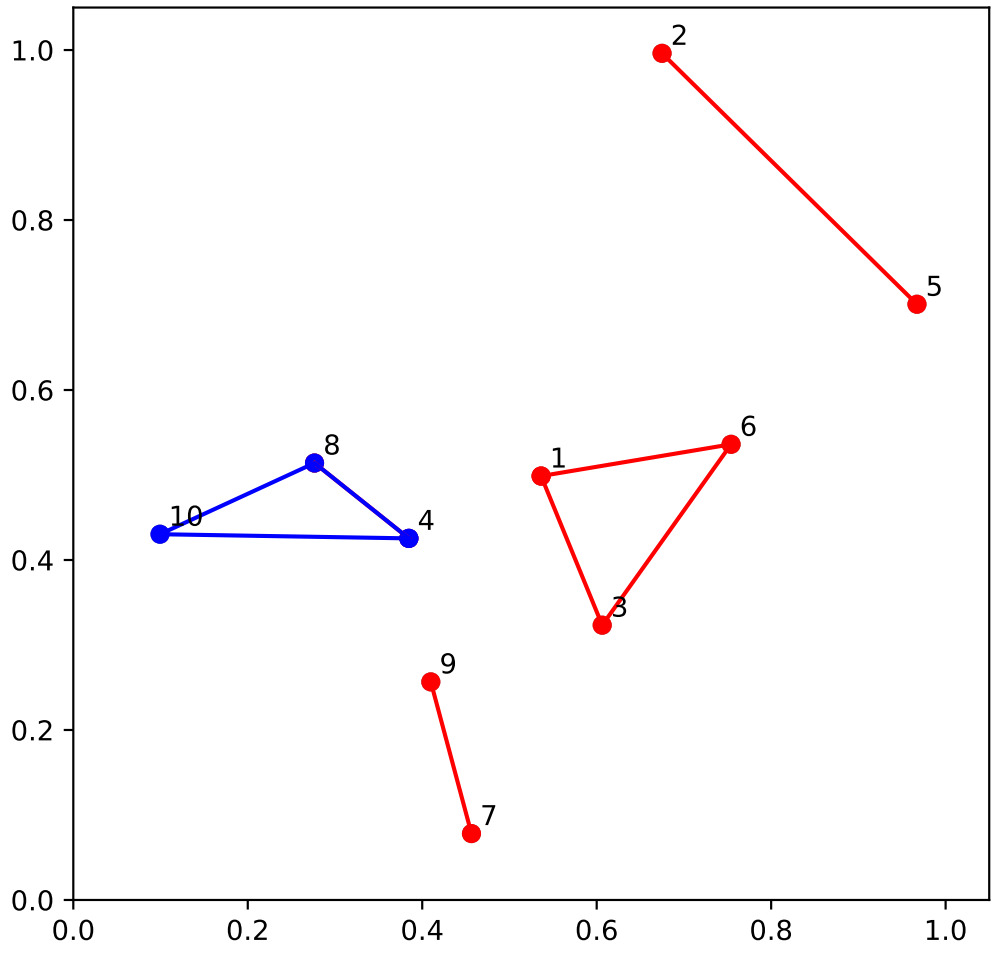
Example (complete linkage)



Example (complete linkage)



Example (complete linkage)



Ward's distance

Proposition

The Ward distance satisfies

$$d(a,b) \leq \min(d(a,c), d(b,c)) \implies d(a \cup b, c) \geq \min(d(a,c), d(b,c))$$

Proof:

$$\begin{aligned} d(a \cup b, c) &= \frac{|a| + |c|}{|a| + |b| + |c|} d(a,c) \\ &\quad + \frac{|b| + |c|}{|a| + |b| + |c|} d(b,c) - \frac{|c|}{|a| + |b| + |c|} d(a,b) \\ &\geq \min(d(a,c), d(b,c)) \end{aligned}$$

Clustering selection

A relevant clustering has a **large distance gap** with the next clustering (i.e., the next cluster merge)

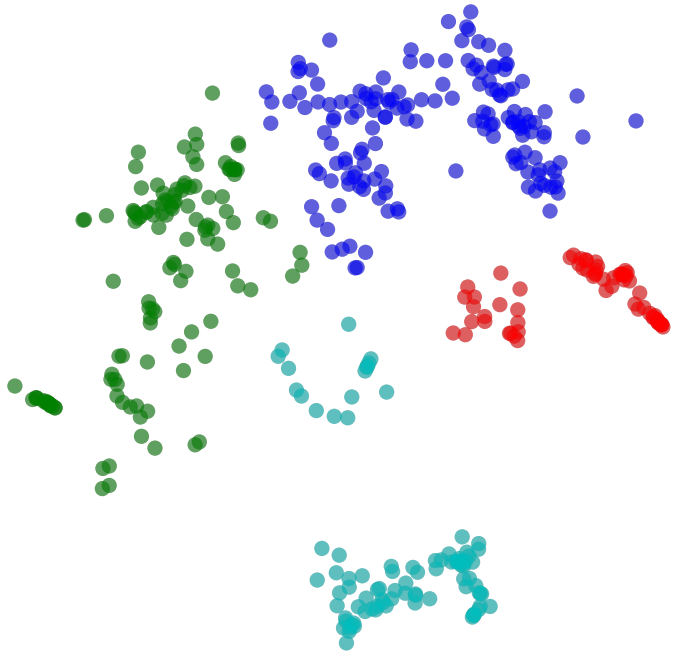
Algorithm

Let $d_1 \leq d_2 \leq \ldots \leq d_{n-1}$ be the distances of the cluster merges

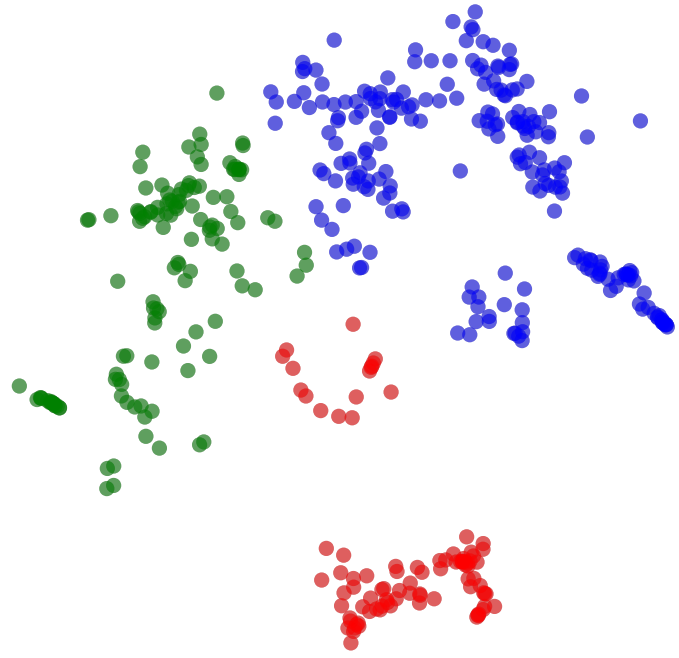
- $\delta_1, \delta_2, \ldots, \delta_{n-1} \leftarrow d_1, d_2 - d_1, \ldots, d_{n-1} - d_{n-2}$
- Sort these values in decreasing order
- $i_1, \ldots, i_{n-1} \leftarrow$ corresponding indices
- Output clusterings $i_1, \ldots, i_{n-1}$

Example (Ward)

Clustering 1

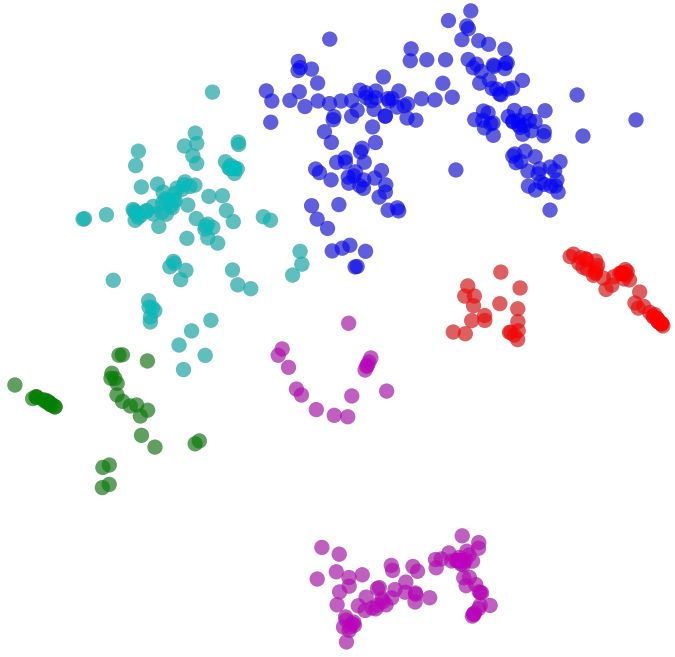


Clustering 2

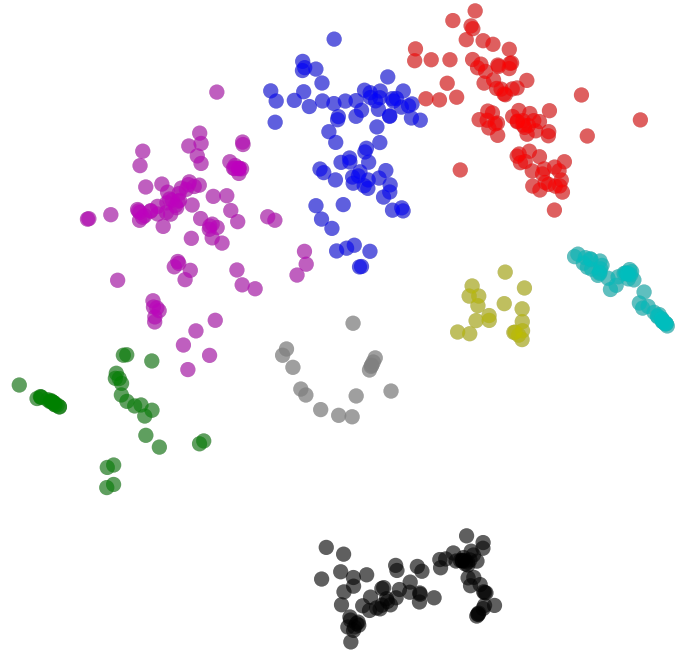


Example (Ward)

Clustering 3



Clustering 4



Performance metrics for clustering

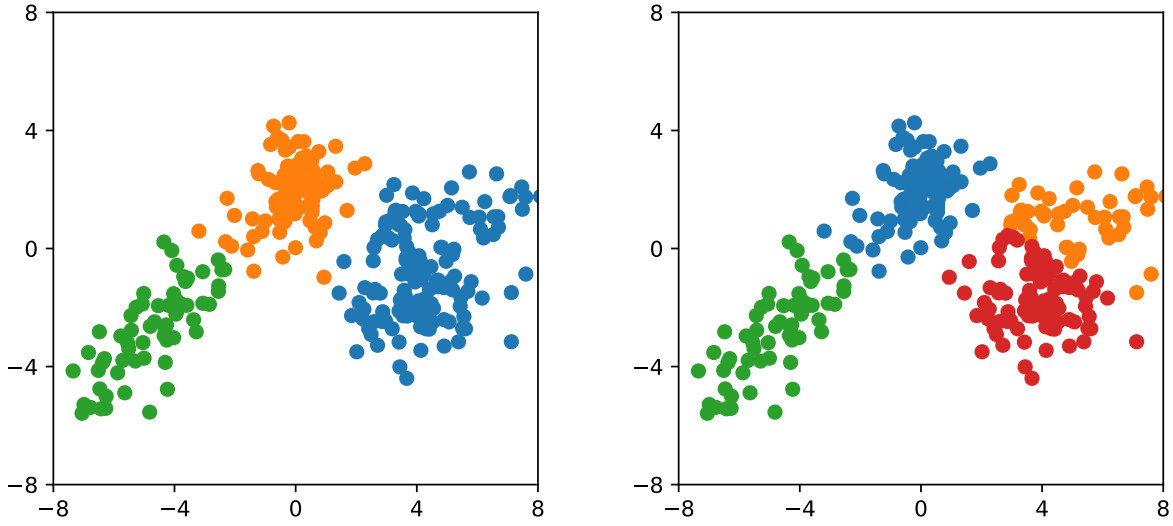
Thomas Bonald

2019 – 2020

Introduction

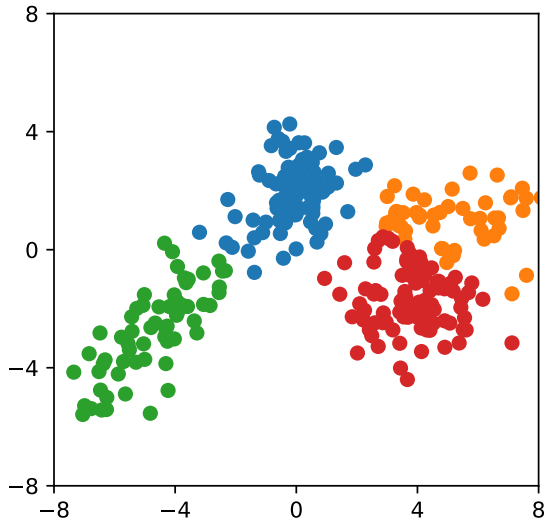
Problem

How to assess the **quality** of a clustering?



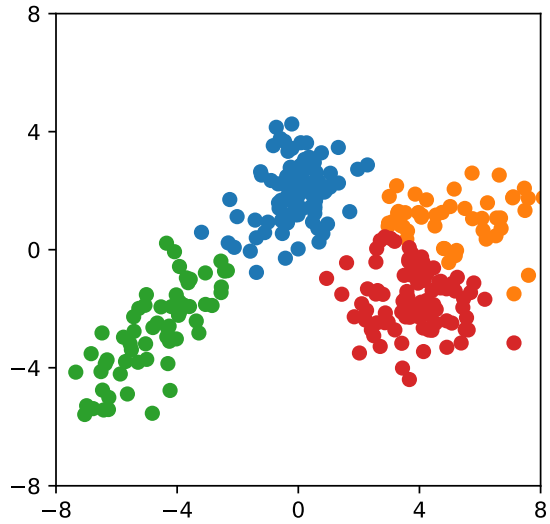
Outline

- Unsupervised metrics
- Supervised metrics
  i.e., with available **ground-truth**



Outline

- **Unsupervised metrics**
- Supervised metrics
  i.e., with available ground-truth



Cluster dispersion

Let $C_1, \ldots, C_k$ be a **partition** of the n samples

Cluster dispersion

Mean **square error**:

$$\frac{1}{n} \sum_j \sum_{i \in C_j} ||x_i - \mu_j||^2$$

where μ_j is the center of cluster j:

$$\mu_j = \frac{1}{n_j} \sum_{i \in C_j} x_i \quad n_j = |C_j|$$

Cluster concentration

Let $C_1, \ldots, C_k$ be a **partition** of the n samples

Cluster concentration

Concentration score:

$$\text{CS} = 1 - \frac{\sum_j \sum_{i \in C_j} ||x_i - \mu_j||^2}{\sum_i ||x_i - \mu||^2}$$

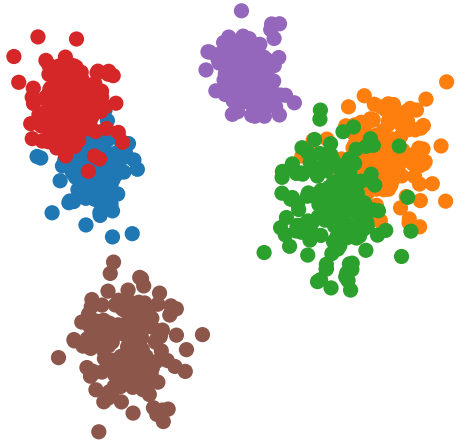
where μ is the center of the dataset:

$$\mu = \frac{1}{n} \sum_i x_i$$

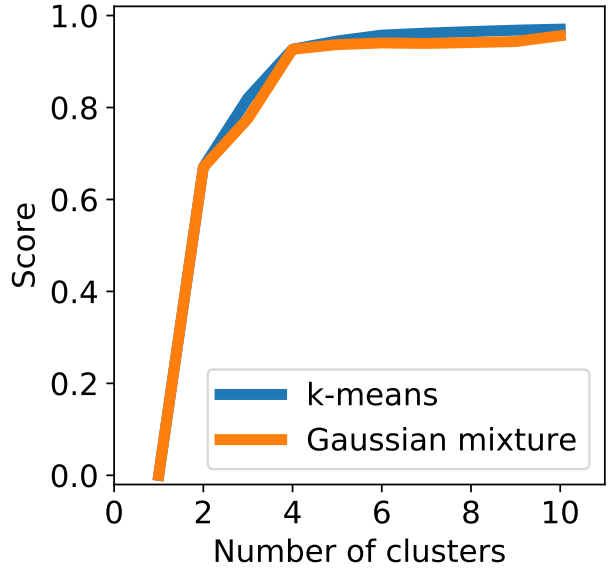
Note: $0 \leq \text{CS} \leq 1$ with $\text{CS} = 0$ for 1 cluster and $\text{CS} = 1$ for n clusters

Example

Data



Concentration score



Other unsupervised metrics

- **Silhouette** coefficient

$$S = 1 - \frac{1}{n} \sum_{i=1}^{n} \frac{d_i^1}{d_i^2}$$

where d_i^1, d_i^2 are the average distances of sample i to samples of the same cluster and to samples of the next nearest cluster.

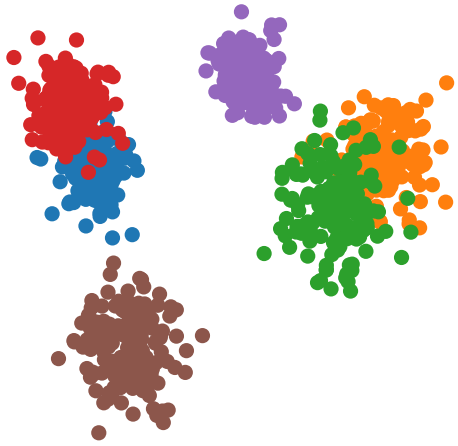
- **Davies-Bouldin** Index

$$DB = 1 - \frac{1}{k} \sum_{j=1}^{k} \max_{j' \neq j} \frac{d_j + d_{j'}}{d_{jj'}},$$

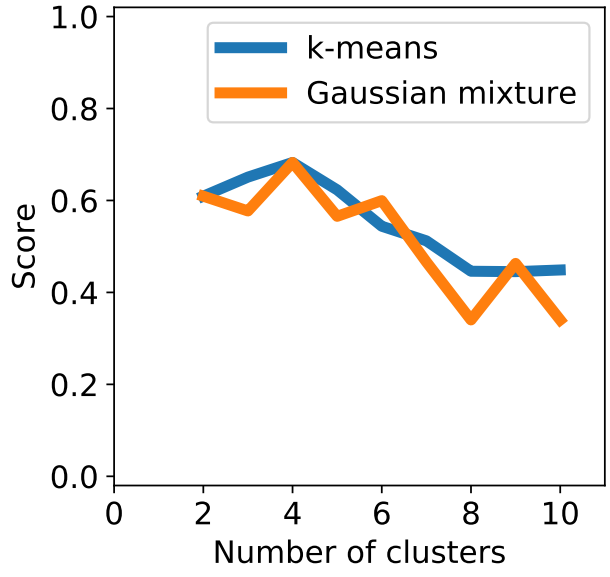
where d_j is the average distance of samples of cluster j to the cluster center μ_j, and $d_{jj'} = ||\mu_j - \mu_{j'}||$ is the distance between the centers of clusters j and j'.

Example

Data

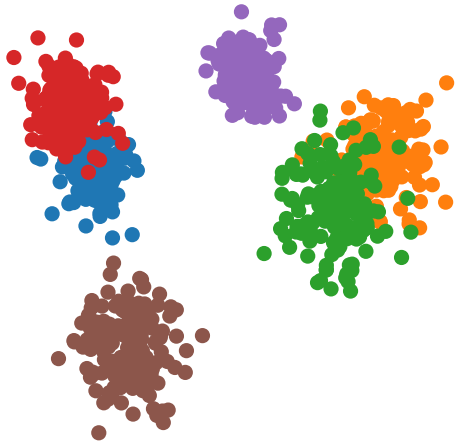


Silhouette score

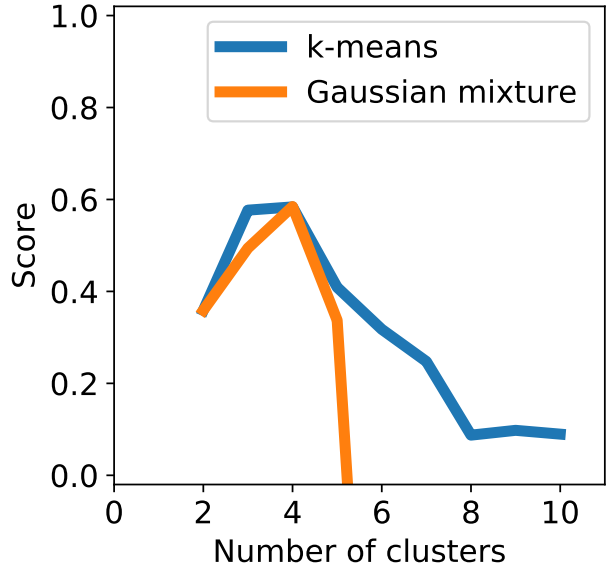


Example

Data

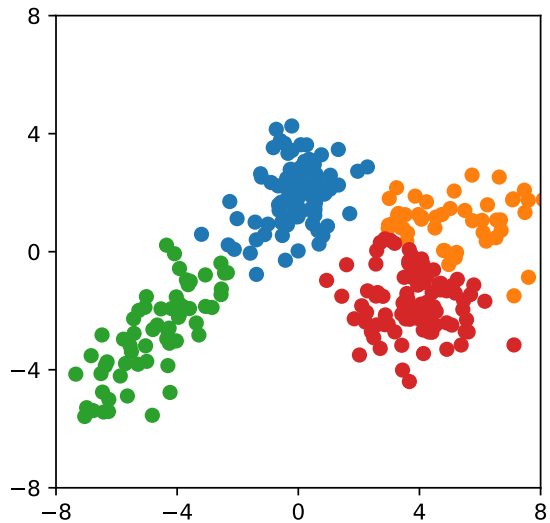


Davies-Bouldin score



Outline

- Unsupervised metrics
- **Supervised metrics**
  i.e., with available **ground-truth**



Ground-truth

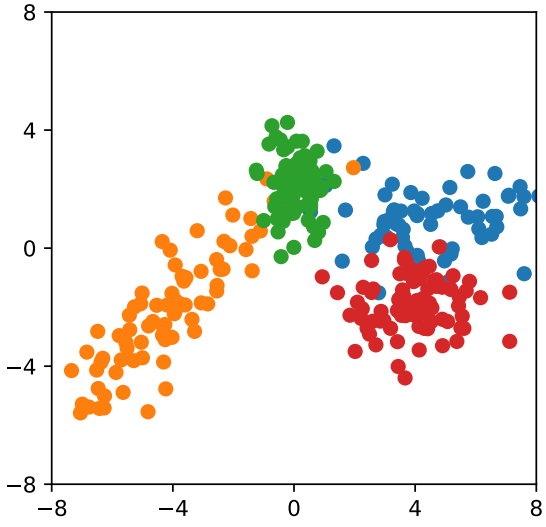
Assume the ground truth is provided in the form of **labels**

Problem

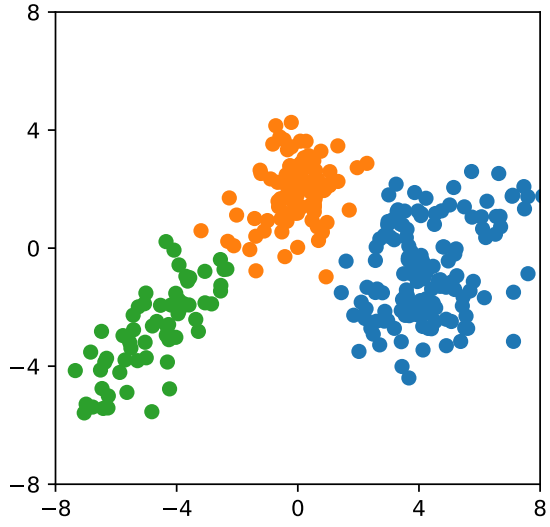
How to assess the **quality** of the clustering?

Need to **align** the predicted labels with the true labels!

True labels



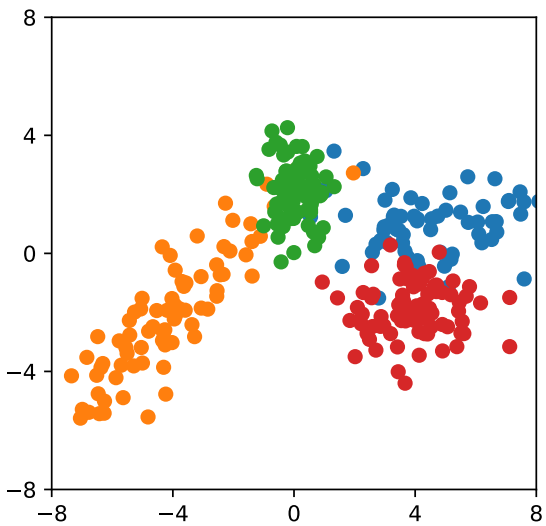
Predicted labels



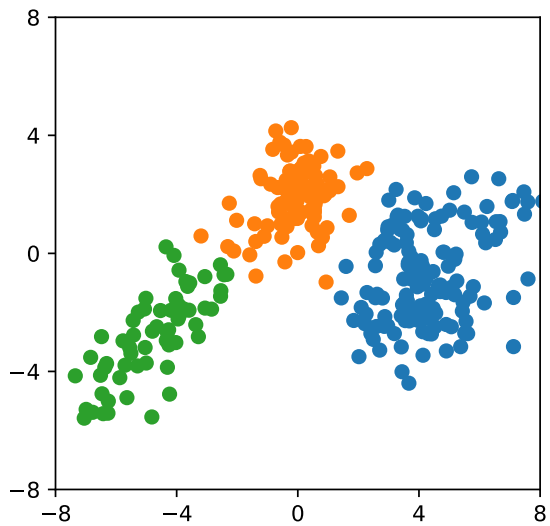
Some supervised metrics

- AF1 (Average F1 Score)
- ARI (Adjusted Rand Index)
- AMI (Adjusted Mutual Information)

True labels



Predicted labels



Counting samples

Let $G_1, \ldots, G_l$ be the **ground-truth** partition of the n samples induced by the l labels

Contingency table

For any clustering $C_1, \ldots, C_k$, let

$$n_{ij} = |G_i \cap C_j|$$

This defines a matrix of size $l \times k$.

Notes:

- The matrix is given by $Z_G^T Z_C$ where Z_G, Z_C are the **membership** matrices, of respective sizes $n \times l$ and $n \times k$
- The size of ground-truth cluster G_i is $n_i = \sum_j n_{ij}$
- The size of predicted cluster C_j is $n_j = \sum_i n_{ij}$
- The number of samples is $n = \sum_{i,j} n_{ij}$

F1 score

Precision, Recall, F1 score

$$\text{precision}(G_i, C_j) = \frac{n_{ij}}{n_j}$$

$$\text{recall}(G_i, C_j) = \frac{n_{ij}}{n_i}$$

$$\text{F1}(G_i, C_j) = \frac{2n_{ij}}{n_i + n_j}$$

Note: The F1 score is the **harmonic mean** of precision and recall
We have:

$$0 \leq \text{F1}(G_i, C_j) \leq 1$$

with $\text{F1}(G_i, C_j) = 1$ iff $G_i = C_j$

F1 score for clustering

Average F1 score

$$\mathrm{AF1} = \frac{1}{2}\left(\sum_i \frac{n_i}{n}\mathrm{F1}(G_i) + \sum_j \frac{n_j}{n}\mathrm{F1}(C_j)\right)$$

with

$$\mathrm{F1}(G_i) = 2\max_j \frac{n_{ij}}{n_i + n_j} \quad \text{and} \quad \mathrm{F1}(C_j) = 2\max_i \frac{n_{ij}}{n_i + n_j}$$

Notes:

- We have $0 \leq \mathrm{AF1} \leq 1$ with $\mathrm{AF1} = 1$ for **perfect clustering**, that is $k = l$ and $\{C_1, \ldots, C_k\} = \{G_1, \ldots, G_k\}$
- **Symmetric** in G and C

Analysis of the Average F1 Score

- Consider k ground-truth clusters with the **same size**
- For the trivial clustering with 1 cluster,

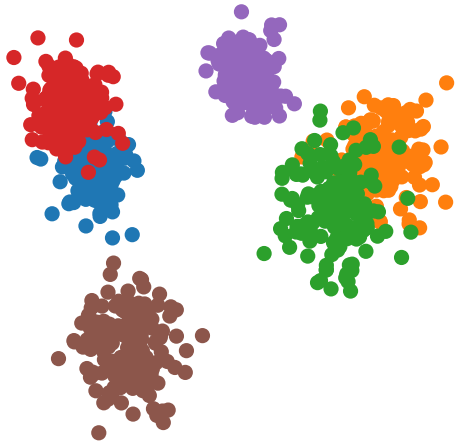
$$\text{AF1} = \frac{2}{k+1}$$

- For the trivial clustering with n clusters,

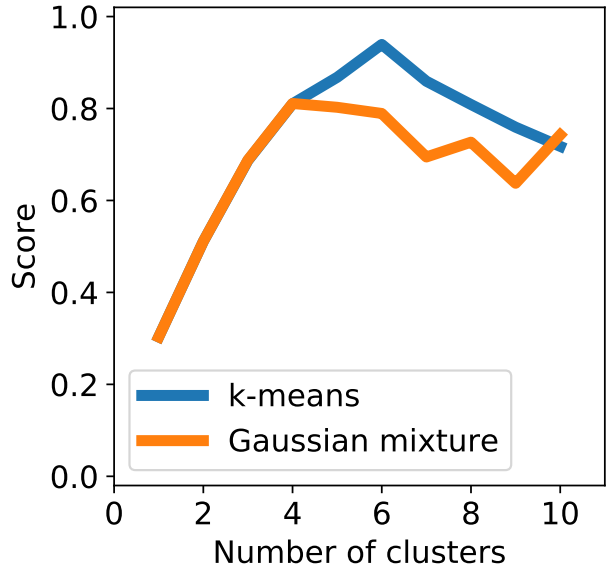
$$\text{AF1} = \frac{2k}{k+n}$$

Example

Data



Average F1 score



Rand Index

Proposed by Rand in 1971

Rand Index

$$\mathrm{RI} = \frac{a+b}{\binom{n}{2}} = 1 - \frac{\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2} - 2\sum_{ij} \binom{n_{ij}}{2}}{\binom{n}{2}}$$

where

$$a = \sum_{ij} \binom{n_{ij}}{2} \quad b = \binom{n}{2} - \sum_i \binom{n_i}{2} - \sum_j \binom{n_j}{2} + \sum_{ij} \binom{n_{ij}}{2}$$

Note:

- $a = \#$ pairs with **same label** and **same cluster**
- $b = \#$ pairs with **different labels** and **different clusters**
- $0 \leq \mathrm{RI} \leq 1$ with $\mathrm{RI} = 1$ for perfect clustering

Proof

Number of pairs with **different ground-truth labels** and in **different clusters**:

$$
\begin{aligned}
b &= \frac{1}{2}\sum_{ij} n_{ij}(n - n_i - n_j + n_{ij}) \\
&= \frac{n^2}{2} - \sum_i \frac{n_i^2}{2} - \sum_j \frac{n_j^2}{2} + \sum_{ij} \frac{n_{ij}^2}{2} \\
&= \binom{n}{2} - \sum_i \binom{n_i}{2} - \sum_j \binom{n_j}{2} + \sum_{ij} \binom{n_{ij}}{2}
\end{aligned}
$$

Analysis of the Rand Index

- Consider k ground-truth clusters with the **same size**
- For the trivial clustering with 1 cluster,

$$\mathrm{RI} = \frac{k\binom{\frac{n}{k}}{2}}{\binom{n}{2}} \approx \frac{1}{k}$$

- For the trivial clustering with n clusters,

$$\mathrm{RI} = 1 - \frac{k\binom{\frac{n}{k}}{2}}{\binom{n}{2}} \approx 1 - \frac{1}{k}$$

Adjusted Rand Index

Adjustment **against chance**: the expected value must be equal to 0 for a random clustering with the same cluster sizes $|C_1|, \ldots, |C_k|$, i.e., random permutation of the rows of the membership matrix Z_C

Adjusted Rand Index

$$\begin{aligned} \mathrm{ARI} &= \frac{\mathrm{RI} - \mathrm{E}(\mathrm{RI})}{1 - \mathrm{E}(\mathrm{RI})} \\ &= \frac{\binom{n}{2}\sum_{ij}\binom{n_{ij}}{2} - \sum_i \binom{n_i}{2}\sum_j \binom{n_j}{2}}{\frac{1}{2}\left(\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2}\right)\binom{n}{2} - \sum_i \binom{n_i}{2}\sum_j \binom{n_j}{2}} \end{aligned}$$

Note: The trivial clusterings with either n clusters or 1 cluster have ARI $= 0$.

Proof

Expected number of pairs with **same label** and **same cluster**:

$$\mathrm{E}(a) = \sum_i \binom{n_i}{2} \sum_j \frac{n_j}{n} \frac{n_j - 1}{n - 1} = \frac{\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}}{\binom{n}{2}}$$

Expected number of pairs **different labels** and **different clusters**:

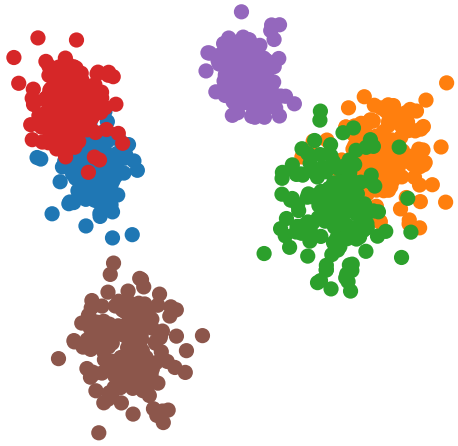
$$\begin{aligned}\mathrm{E}(b) &= \frac{1}{2} \sum_i n_i(n - n_i) \sum_j \frac{n_j}{n} \frac{n - n_j}{n - 1} \\ &= \binom{n}{2} \left(1 - \frac{\sum_i \binom{n_i}{2}}{\binom{n}{2}}\right) \left(1 - \frac{\sum_j \binom{n_j}{2}}{\binom{n}{2}}\right)\end{aligned}$$

Expected RI:

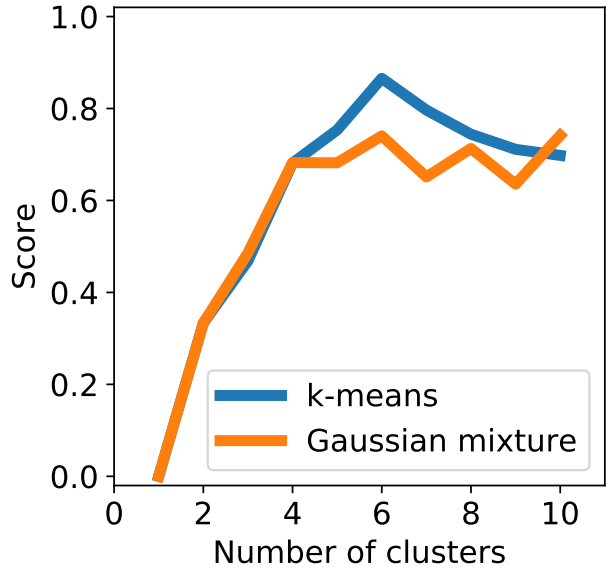
$$\mathrm{E}(\mathrm{RI}) = \frac{\mathrm{E}(a) + \mathrm{E}(b)}{\binom{n}{2}} = 1 - \frac{\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2} - 2 \sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2} / \binom{n}{2}}{\binom{n}{2}}$$

Example

Data



Adjusted Rand Index



Mutual Information

Let X, Y be two discrete random variables

Mutual information

$$I(X, Y) = H(X) + H(Y) - H(X, Y)$$

The quantity of **mutual** information in X and Y:

$$\begin{aligned} I(X, Y) &= -\sum_x p(x) \log p(x) - \sum_y p(y) \log p(y) \\ &\quad + \sum_{x,y} p(x, y) \log p(x, y) \\ &= \sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \\ &= D(p(x, y)||p(x)p(y)) \geq 0 \quad \text{with equality iff } X \perp Y \end{aligned}$$

Mutual Information

Let X, Y be two discrete random variables

Mutual information

$$I(X, Y) = H(X) + H(Y) - H(X, Y)$$

Note:

$$I(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

In particular,

$$I(X, Y) \leq \min(H(X), H(Y))$$

Mutual Information for clustering

Mutual information

Mutual information between labels and clusters:

$$\mathrm{MI} = \sum_{i=1}^{l} \sum_{j=1}^{k} p_{ij} \log \frac{p_{ij}}{p_i p_j}$$

where

$$p_{ij} = \frac{n_{ij}}{n}$$

Note:

$$\mathrm{MI} \leq \min(\log l, \log k)$$

Normalized mutual information

Let X, Y be two discrete random variables

Normalized mutual information

$$\text{NMI} = \frac{2I(X,Y)}{H(X) + H(Y)}$$

We have:

$$0 \leq \text{NMI} \leq 1$$

with $\text{NMI} = 0$ iff $X \perp Y$ and $\text{NMI} = 1$ iff X is determined by Y (and vice versa) .

Normalized mutual information applied to clustering

Normalized mutual information

Normalized mutual information between labels and clusters:

$$\mathrm{NMI} = -\frac{2\sum_{i,j} p_{ij} \log \frac{p_{ij}}{p_i p_j}}{\sum_i p_i \log p_i + \sum_j p_j \log p_j}$$

We have:

$$0 \leq \mathrm{NMI} \leq 1$$

with $\mathrm{NMI} = 1$ for **perfect clustering**, that is $k = l$ and $\{C_1, \ldots, C_k\} = \{G_1, \ldots, G_k\}$

V-measure

Another name / interpretation for NMI.

V-measure

Harmonic mean between **homogeneity** and **completeness**:

$$2\left(\frac{1}{h}+\frac{1}{c}\right)^{-1}$$

with

$$h=-\frac{\sum_{i,j} p_{ij}\log\frac{p_{ij}}{p_i p_j}}{\sum_i p_i \log p_i} \qquad c=-\frac{\sum_{i,j} p_{ij}\log\frac{p_{ij}}{p_i p_j}}{\sum_j p_j \log p_j}$$

Note:

- $h=1$ if and only if $\forall i, \exists j, G_i \subset C_j$
- $c=1$ if and only if $\forall j, \exists i, C_j \subset G_i$

Analysis of Normalized Mutual Information

- Consider k ground-truth clusters with the same size
- For the trivial clustering with 1 cluster,

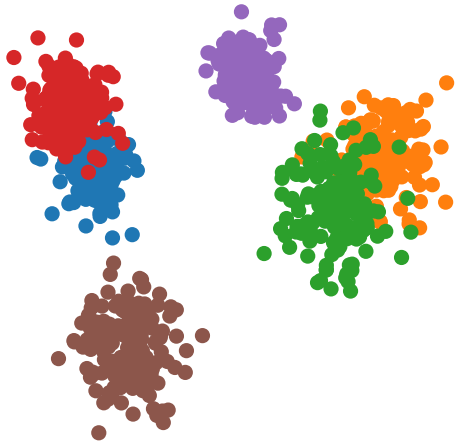
$$\text{NMI} = 0$$

- For the trivial clustering with n clusters,

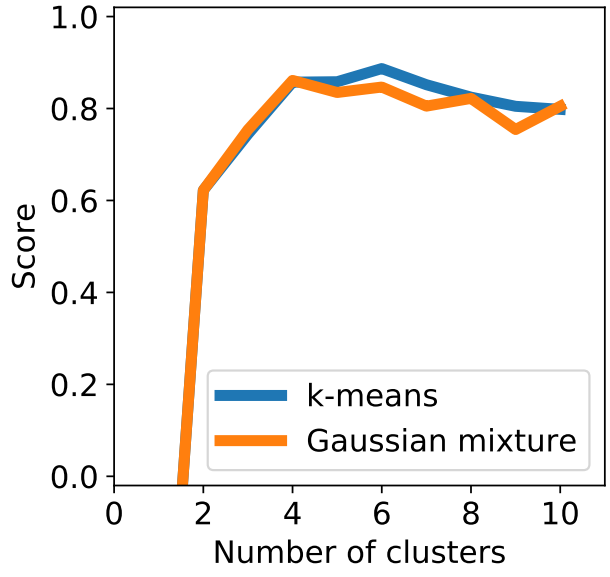
$$\text{NMI} = \frac{2 \log k}{\log k + \log n}$$

Example

Data



NMI



Adjusted mutual information

Adjustment **against chance**: the expected value must be equal to 0 for a random clustering with the same cluster sizes $|C_1|, \ldots, |C_k|$, i.e., random permutation of the rows of the membership matrix Z_C

Adjusted mutual information

$$\mathrm{AMI} = \frac{\mathrm{MI} - \mathrm{E}(\mathrm{MI})}{\max(\mathrm{MI}) - \mathrm{E}(\mathrm{MI})}$$

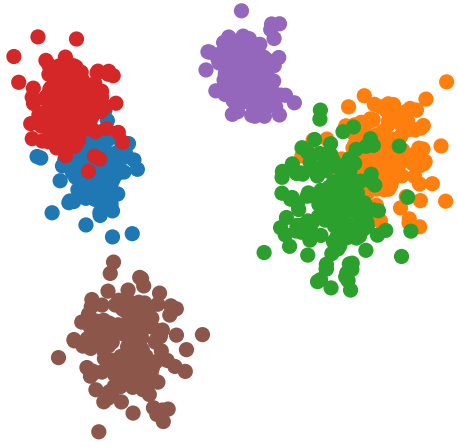
with

$$\mathrm{MI} = \sum_{ij} p_{ij} \log \frac{p_{ij}}{p_i p_j}, \max(\mathrm{MI}) = \max(-\sum_i p_i \log p_i, -\sum_j p_j \log p_j)$$

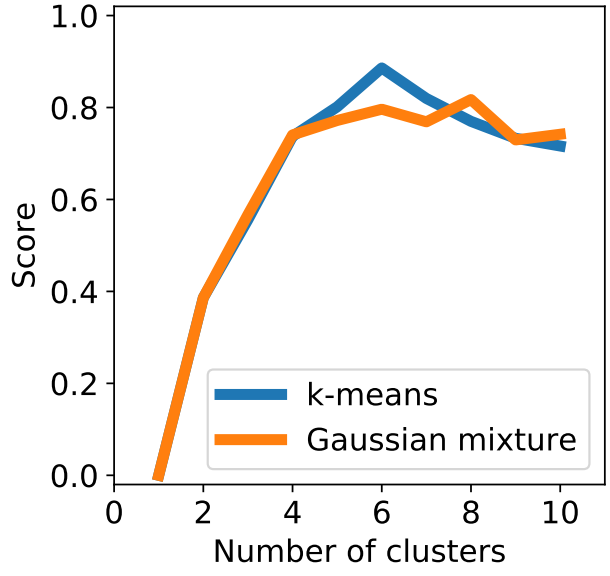
$$\mathrm{E}(\mathrm{MI}) = \sum_{ij} \sum_m \frac{m}{n} \log \frac{mn}{n_i n_j} \frac{\binom{n}{m, n_i - m, n_j - m}}{\binom{n}{n_i}\binom{n}{n_j}}$$

Example

Data



AMI



Impact of granularity

Consider k ground-truth clusters of same size

- **Merge** = clusters merged by groups of m (with $m \leq k$)
- **Split** = clusters splitted in m subclusters of same size (with $m \leq n/k$)

Clustering	AF1	ARI	NMI
Merge	$\frac{2}{m+1}$	$\approx \frac{k-m}{\frac{m+1}{2}k-m}$	$\frac{2(\log k - \log m)}{2\log k - \log m}$
Split	$\frac{2}{m+1}$	$\approx \frac{k-1}{k\frac{m+1}{2}-1}$	$\frac{2\log k}{2\log k + \log m}$

Example

Scores for 1000 samples and 20 ground-truth clusters of same size
Granularity > 1 for splitted clusters and < 1 for merged clusters