# Unsupervised Classification

*A partioning-based clustering algorithm: K-means*

Data Analysis – juliette.chevallier@insa-toulouse.fr

INSA Toulouse, Applied Mathematics, 4th year

---

# K-means type methods

## 1.1 General principle

## 1.2 Choice of hyper-parameters

# Introduction

We observe $n$ individuals described by $p$ variables: $x_i = \left(x_{i1}, x_{i2}, \ldots, x_{ip}\right) \in \mathcal{X}$

$$X = \begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1p} \\ x_{21} & x_{22} & \ldots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \ldots & x_{np} \end{pmatrix}$$

$\mathcal{X} = \mathbb{R}^p\ ,\ ]-\pi, \pi]^p\ , \ldots$ Quantitative variables!

- Initial measurements
- Transformed measurements
- Coordinates after dimension reduction

We observe $n$ individuals described by $p$ variables: $x_i = \left(x_{i1}, x_{i2}, \ldots, x_{ip}\right) \in \mathcal{X}$

$$X = \begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1p} \\ x_{21} & x_{22} & \ldots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \ldots & x_{np} \end{pmatrix}$$

$\mathcal{X} = \mathbb{R}^p$ , $]-\pi, \pi]^p$ , ... Quantitative variables!

- Initial measurements
- Transformed measurements
- Coordinates after dimension reduction

- Let $d$ be the Euclidean distance: $d(x, y) = \|x - y\|^2$

- **Goal** of $k$-means algorithm: Find a partition of the individuals that minimizes the intra-class inertia, *i.e.* the within-cluster sum of squares (WCSS)

$$\widehat{\mathcal{P}}_K^{k-means} \in \underset{\mathcal{P}_K}{\operatorname{argmin}} \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} d(\mu_k, x_i)^2 \qquad \text{where} \quad \mu_k = \frac{1}{|\mathcal{C}_k|} \sum_{i \in \mathcal{C}_k} x_i$$

$\mathcal{P}_K = \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$ partition of $[\![1, n]\!]$

# $k$-means algorithm [MacQueen, 1967, Steinhaus, 1957]

**Initialization:**

- Choice of the number of classes $K$
- Choice of $K$ initial centroids $\mu_1^{(0)}, \ldots, \mu_K^{(0)}$

**Iteration $t$:** Repeat:

**Allocation update:** Point $i$ allocated to the nearest centroid

$$i \in \mathcal{C}_k^{(t)} \quad \text{such that} \quad d\big(x_i, \mu_k^{(t-1)}\big) = \min_{\ell \in [\![1,K]\!]} d\big(x_i, \mu_\ell^{(t-1)}\big)$$

**Centroids update:** Centroid as the new class mean

$$\mu_k^{(t)} = \frac{1}{|\mathcal{C}_k^{(t)}|} \sum_{i \in \mathcal{C}_k^{(t)}} x_i$$

# $k$-means algorithm [MacQueen, 1967, Steinhaus, 1957]

**Initialization:**

- Choice of the number of classes $K$
- Choice of $K$ initial centroids $\mu_1^{(0)}, \ldots, \mu_K^{(0)}$

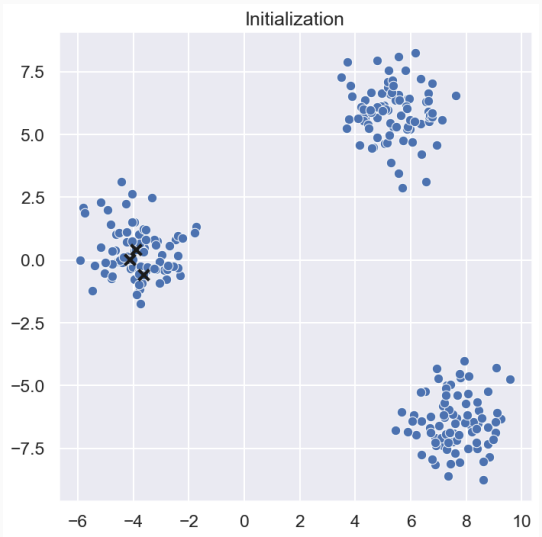
**Iteration $t$:** Repeat:

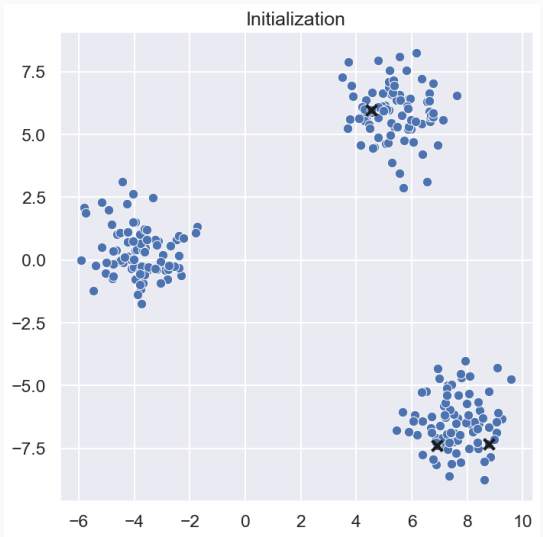
**Allocation update:** Point $i$ allocated to the nearest centroid

$$i \in \mathcal{C}_k^{(t)} \quad \text{such that} \quad d\left(x_i, \mu_k^{(t-1)}\right) = \min_{\ell \in [\![1,K]\!]} d\left(x_i, \mu_\ell^{(t-1)}\right)$$

**Centroids update:** Centroid as the new class mean

$$\mu_k^{(t)} = \frac{1}{|\mathcal{C}_k^{(t)}|} \sum_{i \in \mathcal{C}_k^{(t)}} x_i$$



# $k$-means algorithm [MacQueen, 1967, Steinhaus, 1957]

**Initialization:**

- Choice of the number of classes $K$
- Choice of $K$ initial centroids $\mu_1^{(0)}, \ldots, \mu_K^{(0)}$

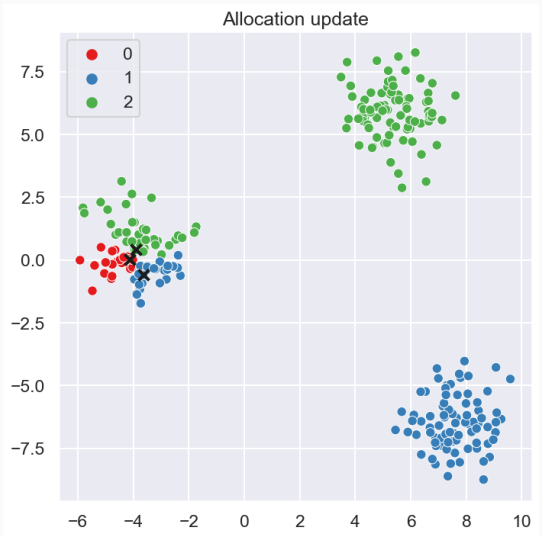
**Iteration $t$:** Repeat:

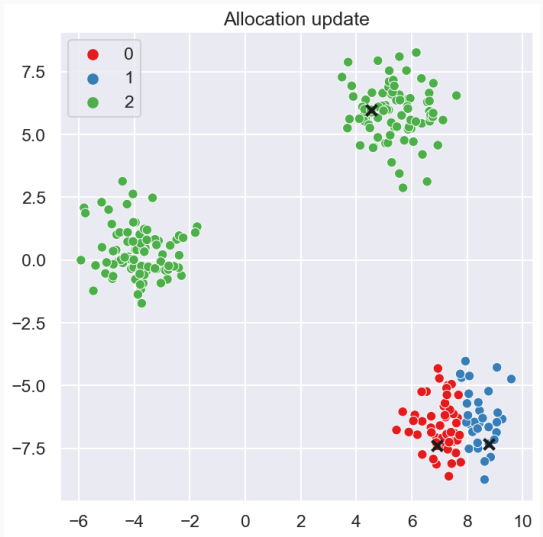
**Allocation update:** Point $i$ allocated to the nearest centroid

$$i \in \mathcal{C}_k^{(t)} \quad \text{such that} \quad d\left(x_i, \mu_k^{(t-1)}\right) = \min_{\ell \in [\![1,K]\!]} d\left(x_i, \mu_\ell^{(t-1)}\right)$$

**Centroids update:** Centroid as the new class mean

$$\mu_k^{(t)} = \frac{1}{|\mathcal{C}_k^{(t)}|} \sum_{i \in \mathcal{C}_k^{(t)}} x_i$$



# $k$-means algorithm [MacQueen, 1967, Steinhaus, 1957]

**Initialization:**

- Choice of the number of classes $K$
- Choice of $K$ initial centroids $\mu_1^{(0)}, \ldots, \mu_K^{(0)}$

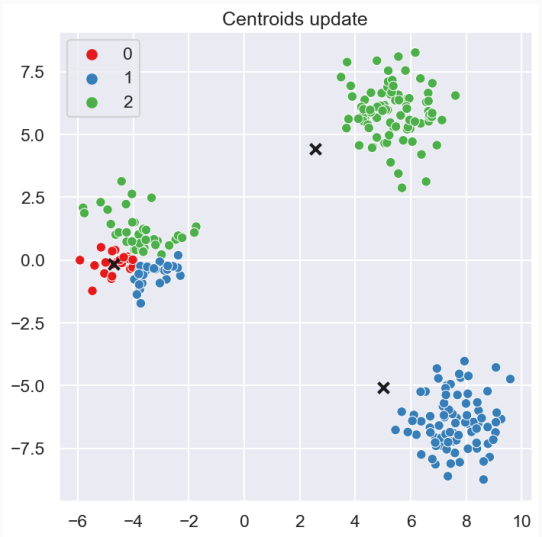
**Iteration $t$:** Repeat:

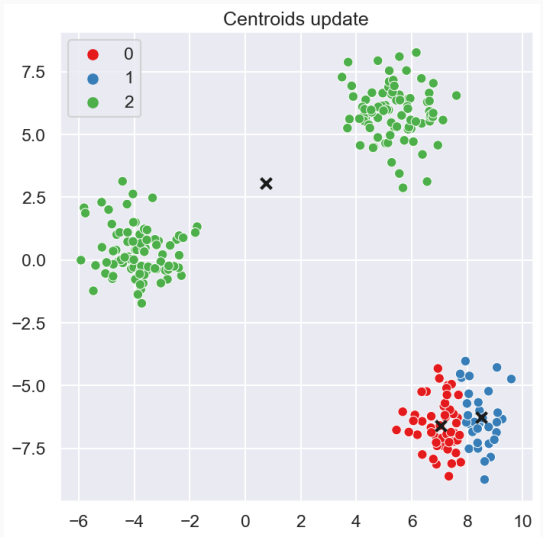
**Allocation update:** Point $i$ allocated to the nearest centroid

$$i \in \mathcal{C}_k^{(t)} \quad \text{such that} \quad d\left(x_i, \mu_k^{(t-1)}\right) = \min_{\ell \in [\![1,K]\!]} d\left(x_i, \mu_\ell^{(t-1)}\right)$$

**Centroids update:** Centroid as the new class mean

$$\mu_k^{(t)} = \frac{1}{|\mathcal{C}_k^{(t)}|} \sum_{i \in \mathcal{C}_k^{(t)}} x_i$$



# $k$-means algorithm [MacQueen, 1967, Steinhaus, 1957]

**Initialization:**

- Choice of the number of classes $K$
- Choice of $K$ initial centroids $\mu_1^{(0)}, \ldots, \mu_K^{(0)}$

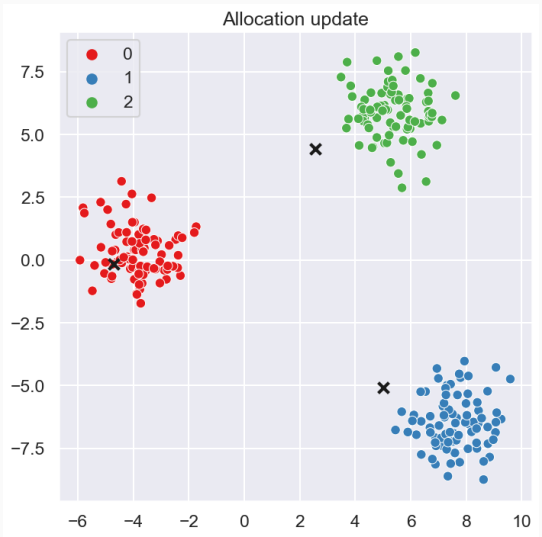
**Iteration $t$:** Repeat:

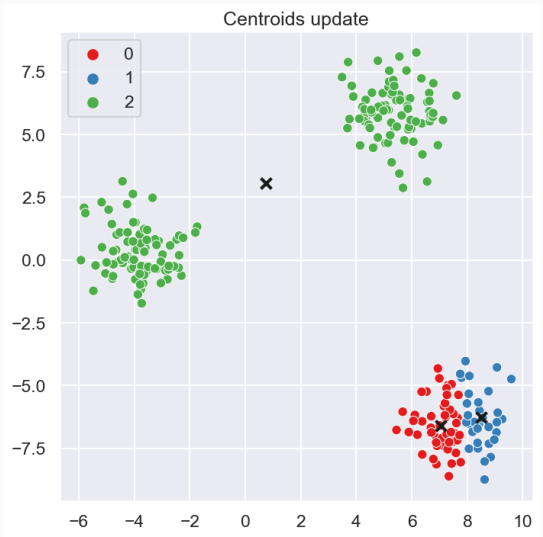
**Allocation update:** Point $i$ allocated to the nearest centroid

$$i \in \mathcal{C}_k^{(t)} \quad \text{such that} \quad d\left(x_i, \mu_k^{(t-1)}\right) = \min_{\ell \in [\![1,K]\!]} d\left(x_i, \mu_\ell^{(t-1)}\right)$$

**Centroids update:** Centroid as the new class mean

$$\mu_k^{(t)} = \frac{1}{|\mathcal{C}_k^{(t)}|} \sum_{i \in \mathcal{C}_k^{(t)}} x_i$$



# $k$-means algorithm [MacQueen, 1967, Steinhaus, 1957]

**Initialization:**

- Choice of the number of classes $K$
- Choice of $K$ initial centroids $\mu_1^{(0)}, \ldots, \mu_K^{(0)}$

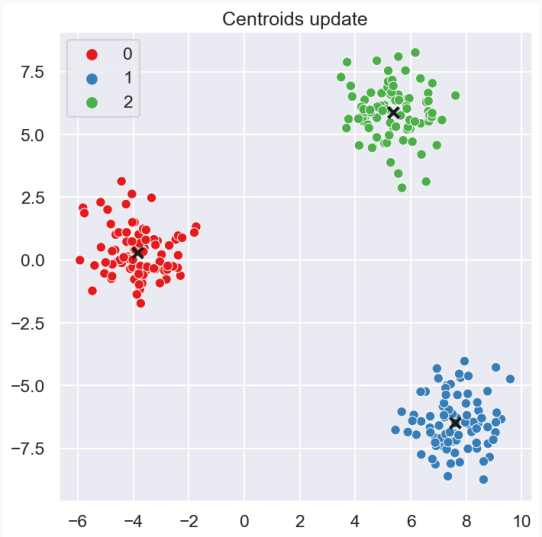
**Iteration $t$:** Repeat:

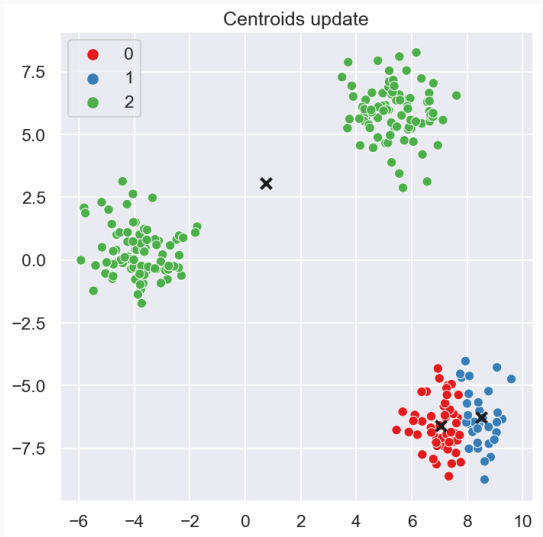
**Allocation update:** Point $i$ allocated to the nearest centroid

$$i \in \mathcal{C}_k^{(t)} \quad \text{such that} \quad d\left(x_i, \mu_k^{(t-1)}\right) = \min_{\ell \in [\![1,K]\!]} d\left(x_i, \mu_\ell^{(t-1)}\right)$$

**Centroids update:** Centroid as the new class mean

$$\mu_k^{(t)} = \frac{1}{|\mathcal{C}_k^{(t)}|} \sum_{i \in \mathcal{C}_k^{(t)}} x_i$$



# Assymptotic Behavior of the $k$-means Algorithm

**Proposition**

The intra-class inertia $I_{Intra}\left(\mathcal{P}_K^{(t)}\right)$ decreases with each step.

$\rightsquigarrow$ Convergence of the $k$-means algorithm towards a local minimum of the intra-class inertia.

**Sketch of the proof**: (Demonstration will be covered in tutorials)

Two key arguments:

1. If point $i$ goes from $\mathcal{C}_k^{(t-1)}$ to $\mathcal{C}_\ell^{(t)}$, then:

$$d\left(x_i, \mu_\ell^{(t)}\right)^2 \leqslant d\left(x_i, \mu_k^{(t-1)}\right)^2$$

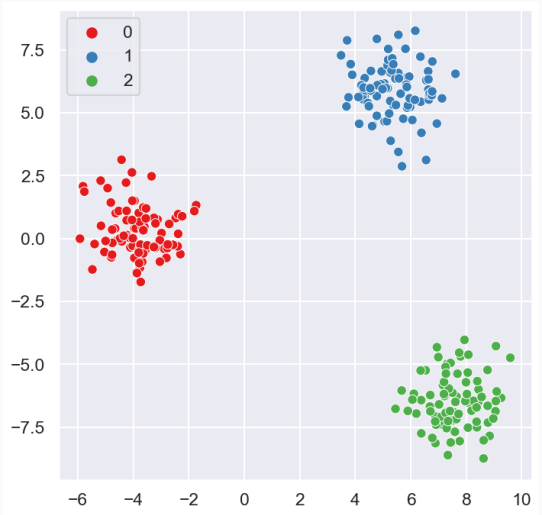
2. $\mu_k^{(t)}$ being the gravity center of $\mathcal{C}_k^{(t)}$,

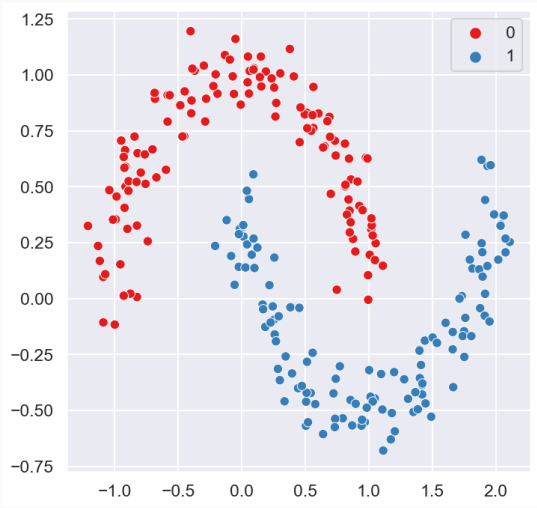
$$\sum_{i \in \mathcal{C}_k^{(t)}} d\left(x_i, \mu_k^{(t)}\right)^2 \leqslant \sum_{i \in \mathcal{C}_k^{(t)}} d\left(x_i, \mu_k^{(t-1)}\right)^2$$

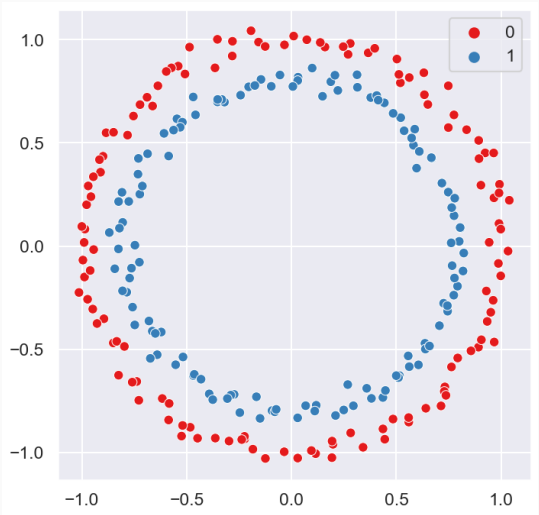
# Strengths and Weaknesses

**Pros:**

- Relatively efficient (fast),
- Tends to reduce intra-class inertia at each iteration,
- Forms compact and well-separated classes.



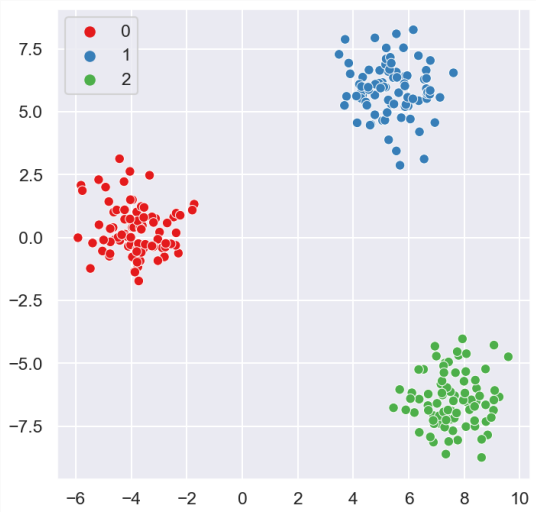
# Strengths and Weaknesses

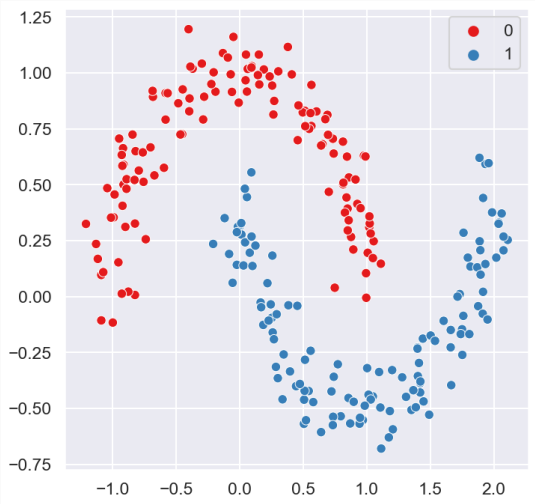
**Pros:**

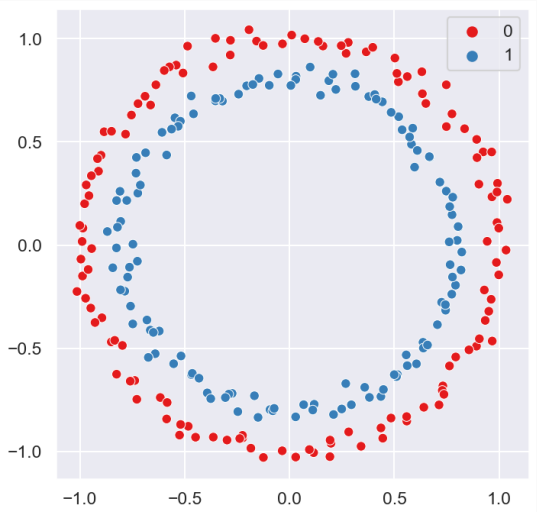
- Relatively efficient (fast),
- Tends to reduce intra-class inertia at each iteration,
- Forms compact and well-separated classes.

**Cons:**

- Influence of the choice of initial centroids,
- Convergence to a local minimum,
- Requires the notion of center of gravity,
- Influence of outliers (due to averaging),
- Not suitable for non-convex classes.



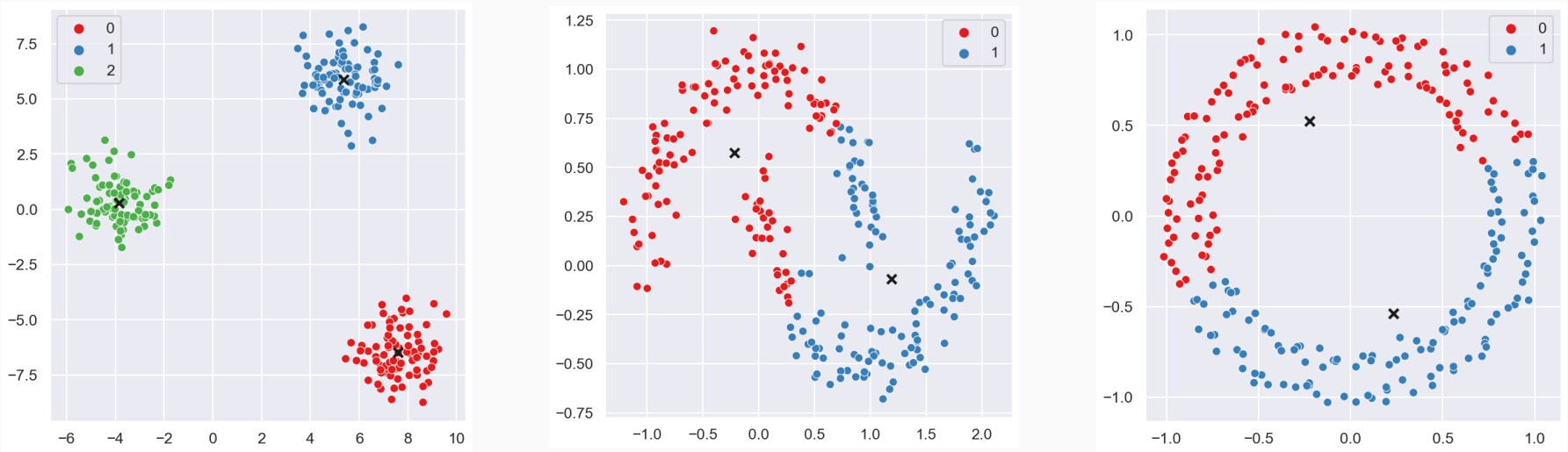
# Strengths and Weaknesses

**Pros:**

- Relatively efficient (fast),
- Tends to reduce intra-class inertia at each iteration,
- Forms compact and well-separated classes.

**Cons:**

- Influence of the choice of initial centroids,
- Convergence to a local minimum,
- Requires the notion of center of gravity,
- Influence of outliers (due to averaging),
- Not suitable for non-convex classes.




Best result obtained by the k-means algorithm, on 10 runs.

# Variants of the $k$-means

**$k$-medoids:** More efficient on **small dataset**, more robust in the presence of **noise** or **outliers**.

*Idea*: Use medoids instead of centroids, *i.e.* points from $X$.

$$\nu_k \in \operatorname*{argmin}_{y \in X} \sum_{i \in \mathcal{C}_k} d(y, x_i)^2$$

# Variants of the $k$-means

**$k$-medoids:** More efficient on **small dataset**, more robust in the presence of **noise** or **outliers**.

*Idea*: Use medoids instead of centroids, *i.e.* points from $X$.

$$\nu_k \in \underset{y \in X}{\operatorname{argmin}} \sum_{i \in \mathcal{C}_k} d(y, x_i)^2$$

**$k$-modes:** For **qualitative** data.

(i) Modify the dissimilarity measure to handle qualitative data.

$$d(x_a, x_b) = \sum_{j=1}^{p} \frac{n_{aj} + n_{bj}}{n_{aj} \times n_{bj}} \mathbb{1}_{\{x_{aj} \neq x_{bj}\}}$$

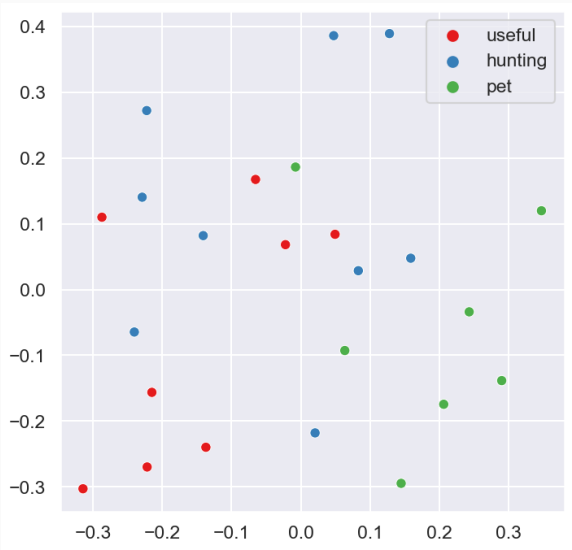
where $n_{aj} = \#\{i \in [\![1, n]\!] \mid x_{aj} = x_{ij}\}$.

(ii) Use of modes instead of centers of gravity.

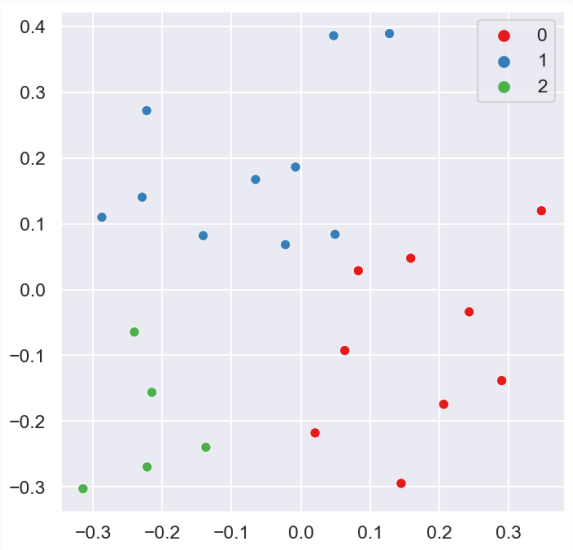
***Remark***: For **qualitative** data, we can also use $k$-means algorithm on multiple correspondence analysis (MCA).

# Dogs Breeds: $k$-modes vs $k$-means+MCA

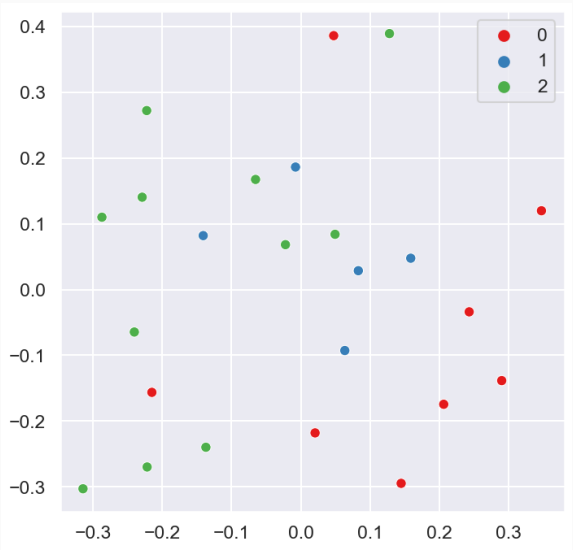
Analysis of 27 dog breeds based on 6 descriptive qualitative: size (3), weight (3), velocity (3), intelligence (3), affection (2) and aggressiveness (2).




MCA



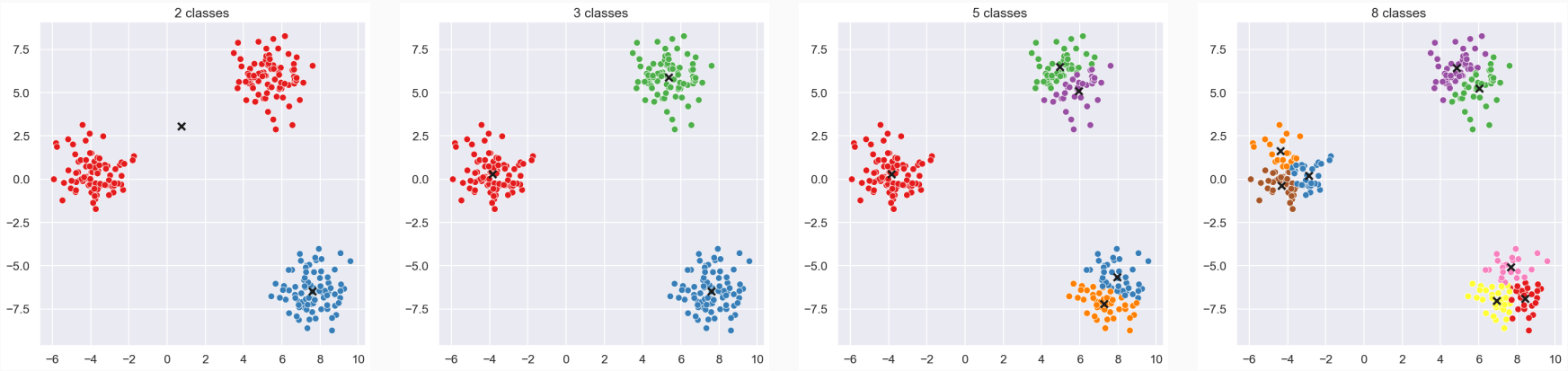

MCA + $k$-means




MCA + $k$-modes

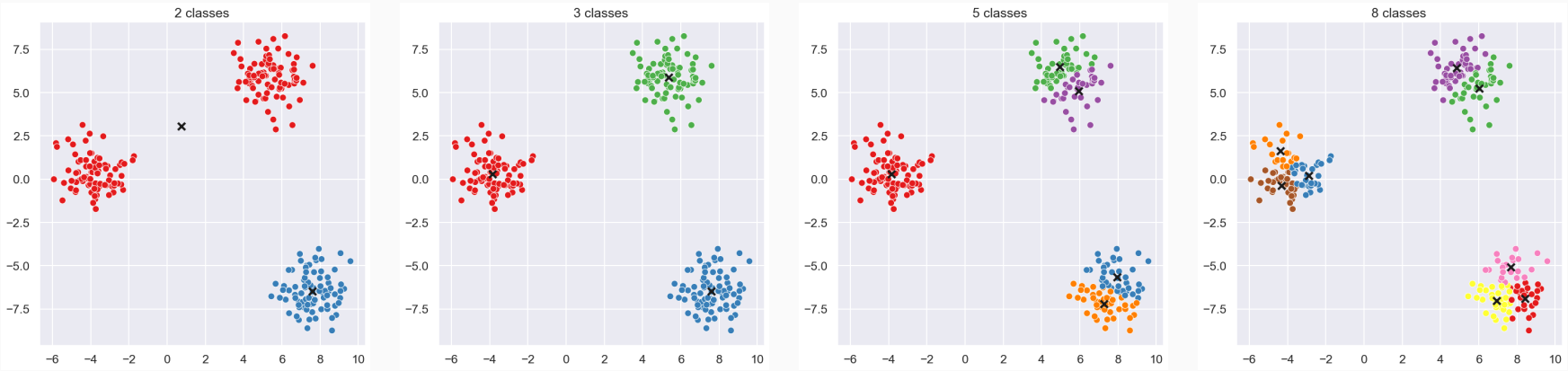
# K-means type methods

# Choice of the Number of Classes $K$




- **Elbow method** for the intra-class inertia $I_{Intra}$
  - For each value of $K \in \{2, \ldots, K_{max}\}$, we obtain a classification $\mathcal{P}_K$,
  - We select the one where we observe a significant jump in intra-class inertia.

# Choice of the Number of Classes $K$




- **Elbow method** for the intra-class inertia $I_{Intra}$
  - For each value of $K \in \{2, \ldots, K_{max}\}$, we obtain a classification $\mathcal{P}_K$,
  - We select the one where we observe a significant jump in intra-class inertia.

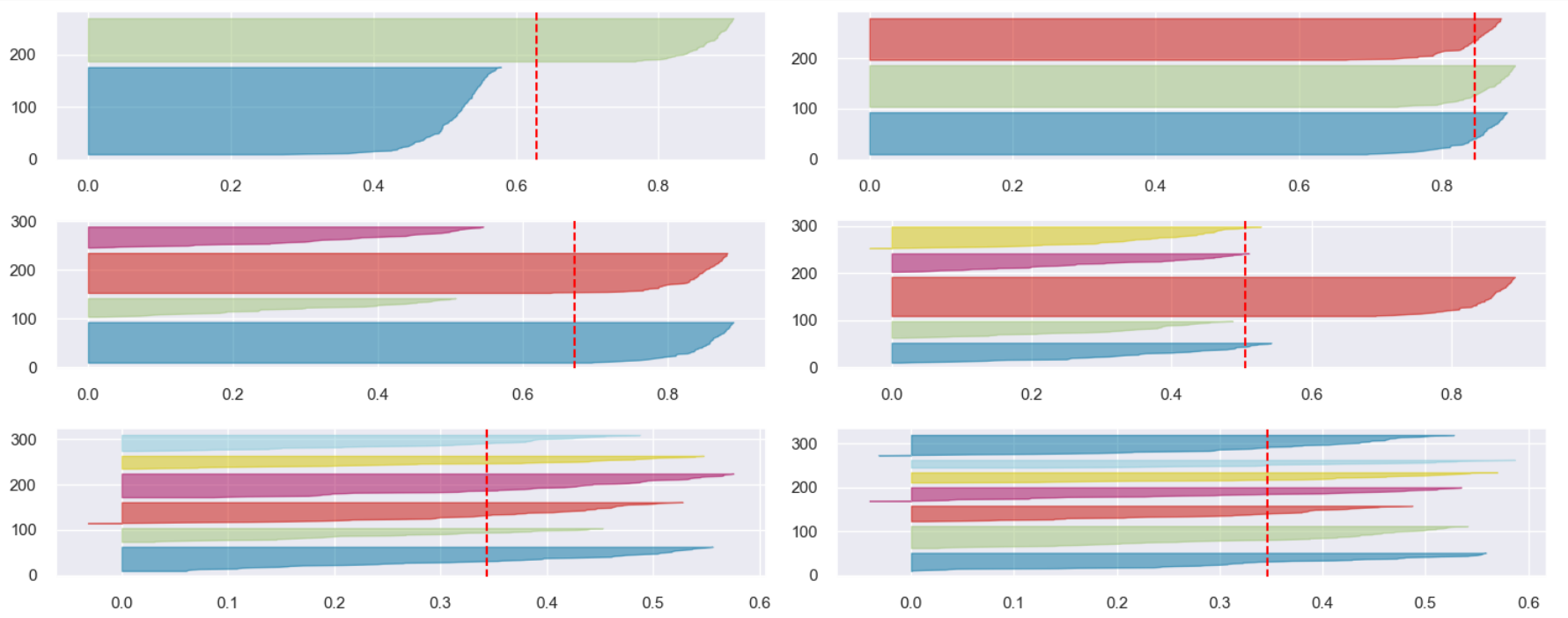
- Other criteria based on inertia
  - **R-Square**: $RSQ(K) = \dfrac{I_{Inter}(\mathcal{P}_K)}{I_{Tot}} = 1 - \dfrac{I_{Intra}(\mathcal{P}_K)}{I_{Tot}}$

    $K$: *Elbow on the* $RSQ$ *curve.*
  - **Semi-Partial R-Square**: $SPRSQ(K) = \dfrac{I_{Inter}(\mathcal{P}_K) - I_{Inter}(\mathcal{P}_{K-1})}{I_{Tot}}$

    $K$: *largest reduction of the* $QSPRS$.
  - **Calinski-Harabasz** (CH): $PseudoF(K) = \dfrac{I_{Inter}(\mathcal{P}_K)}{I_{Intra}(\mathcal{P}_K)} \times \dfrac{n-K}{K-1}$

    $K$: *Peak on the* $CH$ *curve.*

# Silhouette Score



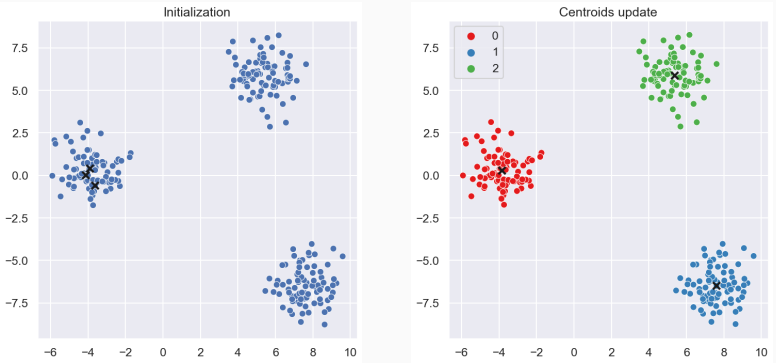

$$s(i) = \frac{b(i) - a(i)}{\max\big(a(i), b(i)\big)}$$

where

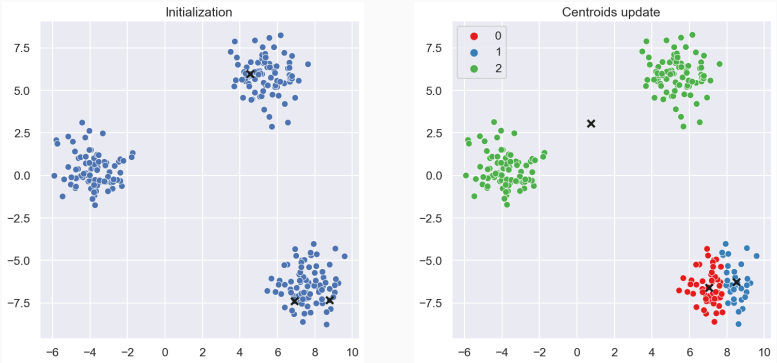
$$\begin{cases} a(i) = \dfrac{1}{|\mathcal{C}_k| - 1} \displaystyle\sum_{j \in \mathcal{C}_k, j \neq i} d(x_i, x_j) \quad \text{(cohesion)}\,, \\ b(i) = \min\limits_{\ell \neq k} \dfrac{1}{|\mathcal{C}_\ell|} \displaystyle\sum_{j \in \mathcal{C}_\ell} d(x_i, x_j) \quad \text{(separation)}\,. \end{cases}$$

- The better the classification, the closer the silhouette score is to 1
- Negative score in case of bad classification

# Choice of Initial Centroids



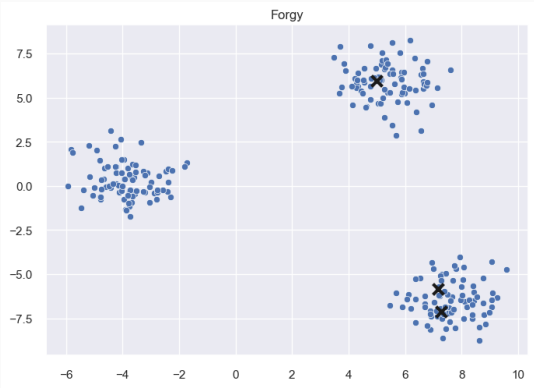
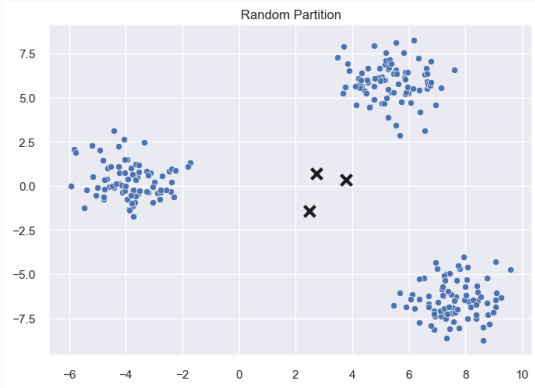

Initialization 1




Initialization 2

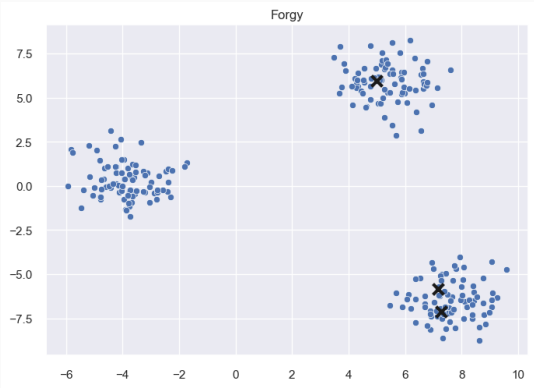
- A judicious choice can favour the convergence towards a global minimum!
- Selection based on additional knowledge,
  or on a preliminary study of the data: histograms, *etc.*
- Repeat the method $N$ times, and select the partition $\mathcal{P}_K$ with the lowest intra-class inertia.

# Choice of Initial Centriods [Arthur and Vassilvitskii, 2006]

| Forgy Initialization | Random Partition Method | $k$-means++ |
|---|---|---|
|  |  |  |
| Any $K$ points from the data, at random. | • Random assignment of a cluster ID to each data point,<br>• Average by ID of the points. | • Choose a random point,<br>• Next centroid so that it lies at a large distance from the first one, with high probability: Sample a point from a probability distrib. proportional to the distance to the first centroid,<br>• Remaining pts generated by a probability disctrib. proportional to the distance of each point from its nearest |

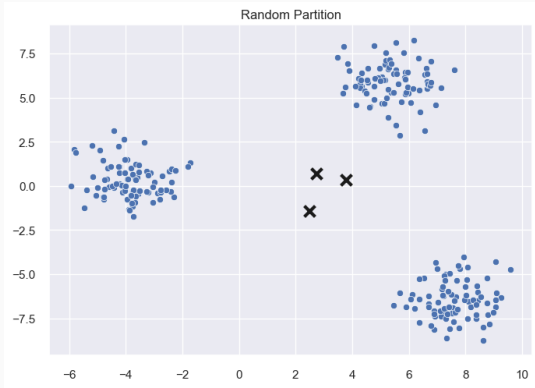
# Choice of Initial Centriods [Arthur and Vassilvitskii, 2006]

## Forgy Initialization




Any $K$ points from the data, at random.

## Random Partition Method




- Random assignment of a cluster ID to each data point,
- Average by ID of the points.

*Not a good choice for $k$-means!*

## $k$-means++


- Choose a random point,
- Next centroid so that it lies at a large distance from the first one, with high probability: Sample a point from a probability distrib. proportional to the distance to the first centroid,
- Remaining pts generated by a probability disctrib. proportional to the distance of each point from its nearest

# DBSCAN: Density-Based Spatial Clustering of Applications with Noise

## 2.1 Principle of DBSCAN methods

## 2.2 Choice of hyper-parameters

# DBSCAN Algorithm [Ester et al., 1996]

- Two **key parameters**:

  $\varepsilon$: The distance that specifies the neighborhoods.
  Two points are considered to be neighbors if the distance between them are less than or equal to $\varepsilon$.

  $MinPts$: Minimum number of data points to define a cluster.

- **Algorithmic steps** for DBSCAN clustering:
  1. Arbitrarily picking up a point in the dataset (until all points have been visited).
  2. If there are at least $MinPts$ points within a radius of $\varepsilon$ to the point then we consider all these points to be part of the same cluster.
  3. The clusters are then expanded by recursively repeating the neighborhood calculation for each neighboring point

See aaronscotthq.com/2020-05-28-scott_dbscan for an animation.

# DBSCAN Algorithm

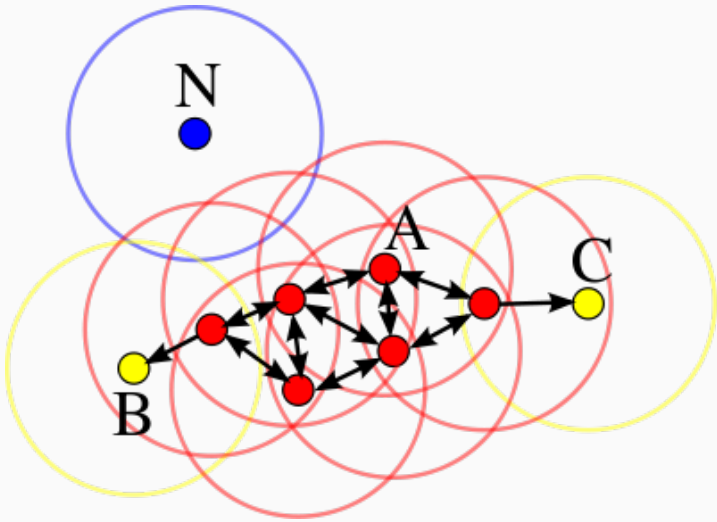
Points classified as **core** point, **border** point, or **outlier**:

**Core point:** There are at least $MinPts$ number of points (including itself) in its $\varepsilon$-neighborhood:

$$\#\{i \in [\![1, n]\!] \mid d(x_a, x_i) < \varepsilon\} \geqslant MinPts\,.$$

**Border point:** Belongs to the $\varepsilon$-neighborhood of a core point, but is not a central point (not enough dense neighborhood).

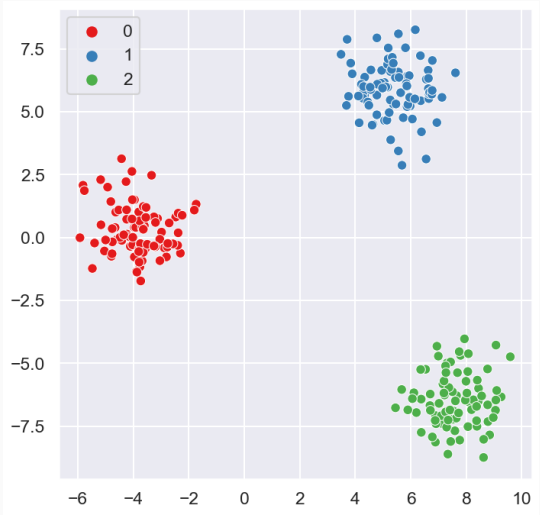
**Outlier:** Neither a central point nor a border point
(In particular, not classified).

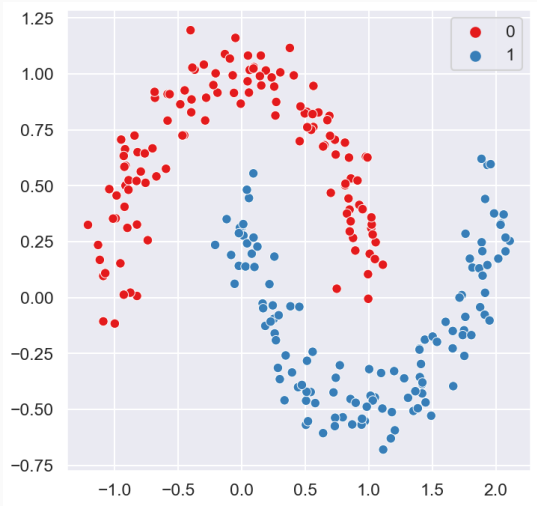


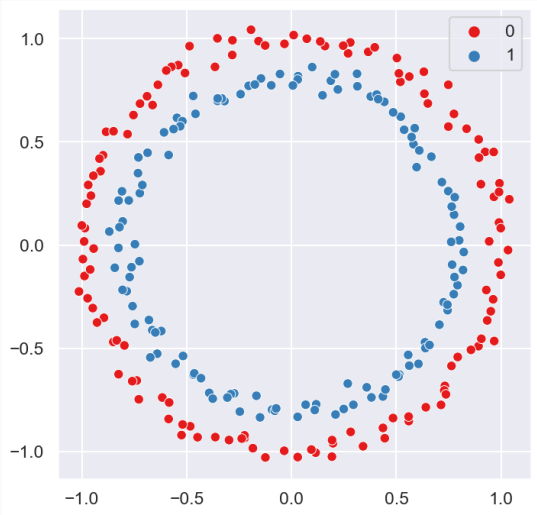
# Strengths and Weaknesses

**Pros:**

- Does not require to specify number of clusters beforehand,
- Performs well with clusters of arbitrary shapes,
- Robust to outliers and able to detect them.



# Strengths and Weaknesses

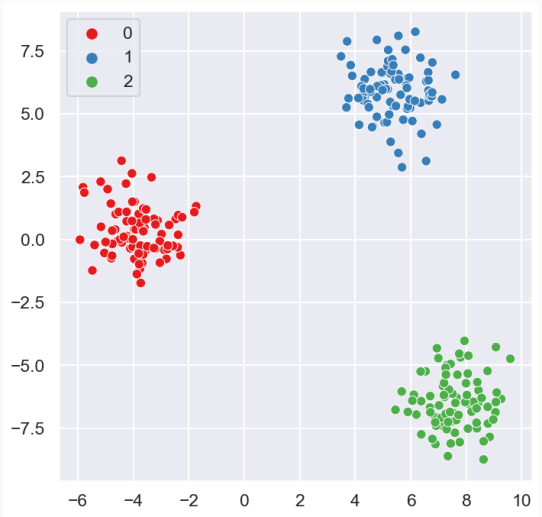
**Pros:**

- Does not require to specify number of clusters beforehand,
- Performs well with clusters of arbitrary shapes,
- Robust to outliers and able to detect them.

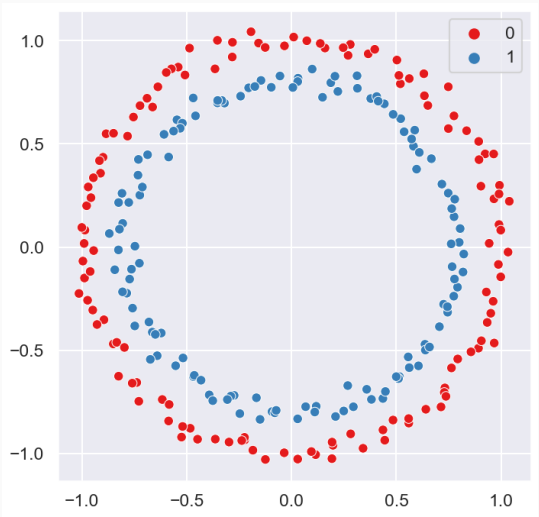
**Cons:**

- In some cases, determining an appropriate neighborhood distance $\varepsilon$ is not easy and requires domain knowledge,
- Not well suited if the clusters are very different from each other in terms of intra-cluster densities.

  Characteristics of the clusters defined by the combination $\varepsilon - MinPts$, and we pass only one couple $\varepsilon - MinPts$ to the algorithm.







# Strengths and Weaknesses

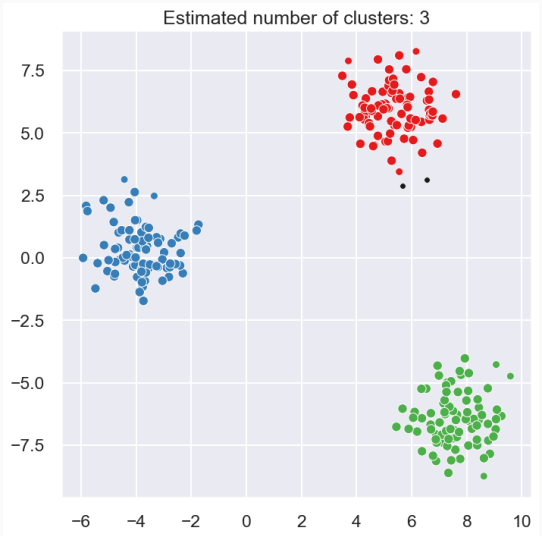
**Pros:**

- Does not require to specify number of clusters beforehand,
- Performs well with clusters of arbitrary shapes,
- Robust to outliers and able to detect them.

**Cons:**

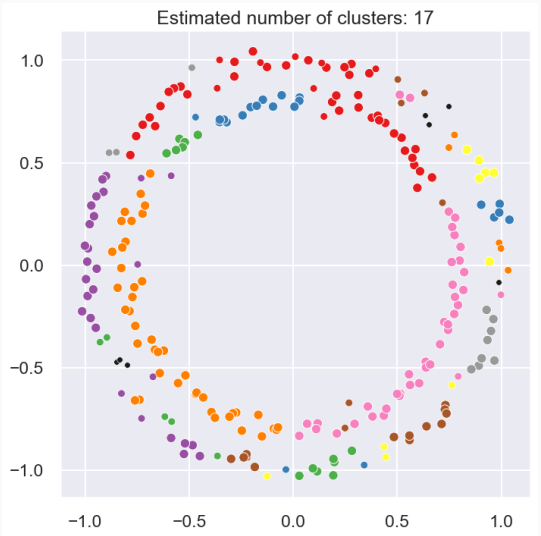
- In some cases, determining an appropriate neighborhood distance $\varepsilon$ is not easy and requires domain knowledge,
- Not well suited if the clusters are very different from each other in terms of intra-cluster densities.

  Characteristics of the clusters defined by the combination $\varepsilon - MinPts$, and we pass only one couple $\varepsilon - MinPts$ to the algorithm.




$MinPts = 4,\ \varepsilon = 1$

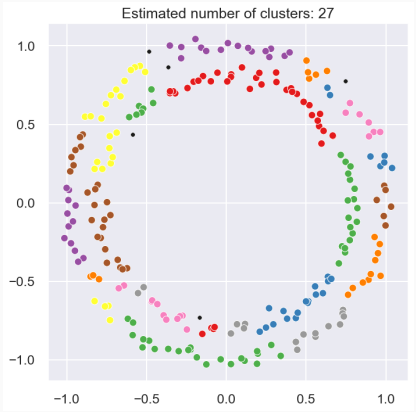

$MinPts = 4,\ \varepsilon = 0.3$



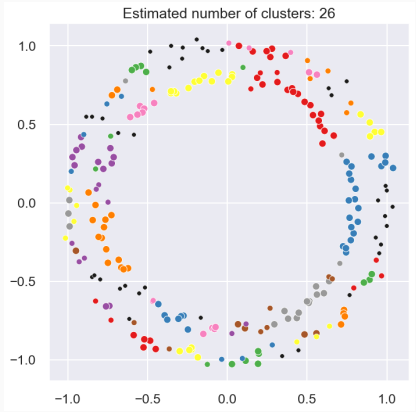

$MinPts = 4,\ \varepsilon = 0.11$

# DBSCAN: Density-Based Spatial Clustering of Applications with Noise

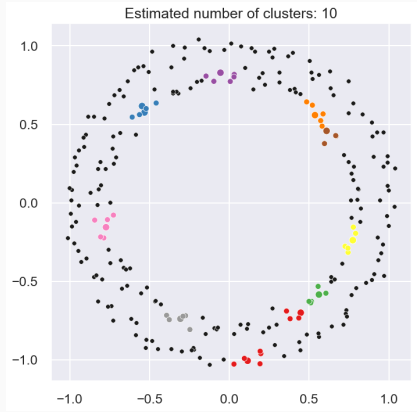
# Determining Minimum Samples $MinPts$



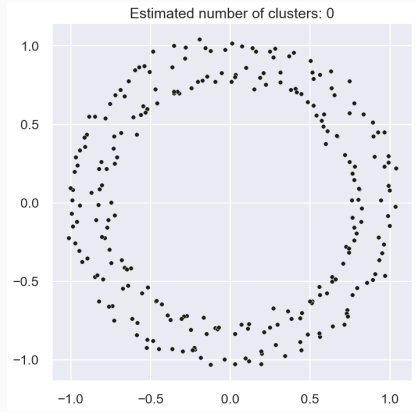

$MinPts = 2,\ \varepsilon = 0.1$




$MinPts = 4,\ \varepsilon = 0.1$




$MinPts = 6,\ \varepsilon = 0.1$

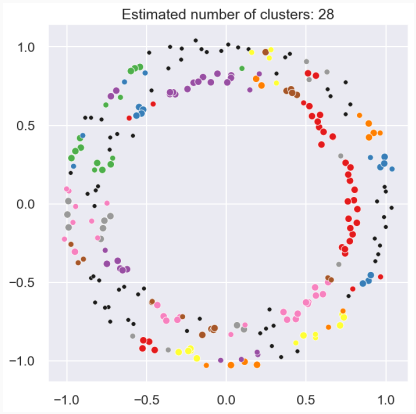



$MinPts = 8,\ \varepsilon = 0.1$

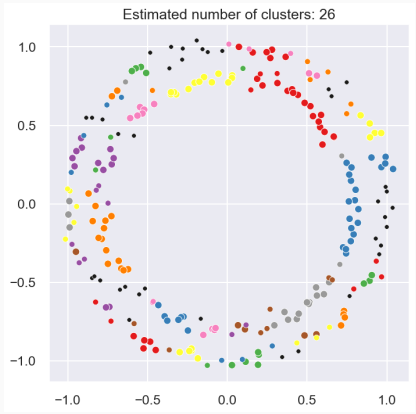
$MinPts$: to be defined using domain knowledge and familiarity with the data set.

- $MinPts \geqslant p + 1$,
- The larger the data set, the larger $MinPts$,
- The noisier the data set, the larger $MinPts$,
- For $2d$ data, use DBSCAN's default value of $MinPts = 4$ [Ester et al., 1996]
- For more than $2d$ data, choose $MinPts = 2p$ [Sander et al., 1998], or $MinPts = \ln(n)$.

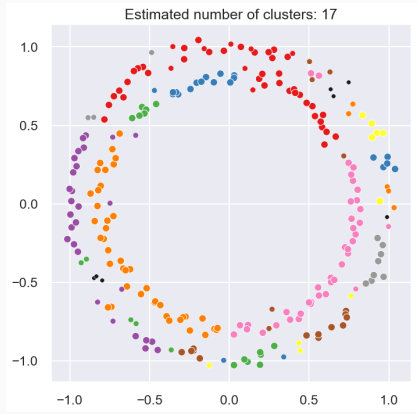
# Determining the distance $\varepsilon$



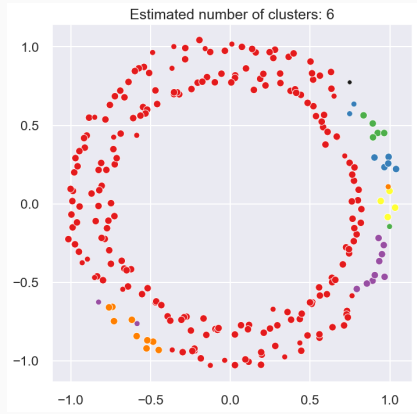

$MPts = 4, \ \varepsilon = 0.09$




$MPts = 4, \ \varepsilon = 0.10$



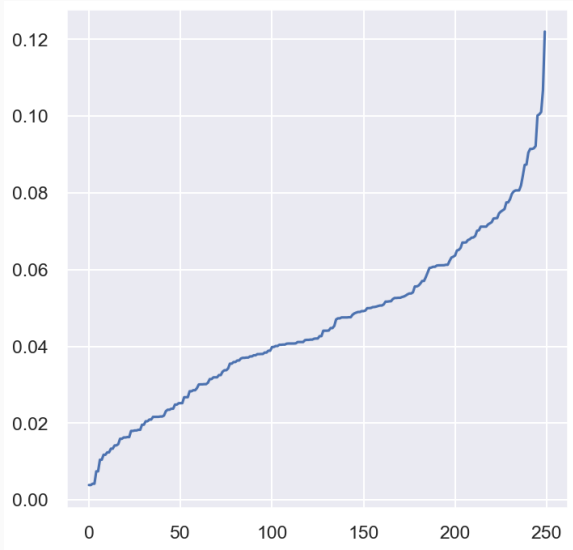

$MPts = 4, \ \varepsilon = 0.11$




$MPts = 4, \ \varepsilon = 0.12$

$\varepsilon$: Based on the average distance between each point and its $MinPts$ nearest neighbors ($MinPts - NN$ distance)

- For *each* point of the dataset, compute its $MinPts - NN$ distance,
- Plot this distances in ascending,
- We choose $\varepsilon$ as the value of the $MinPts - NN$ distance where an "elbow" is observed (maximum curvature).



# References i


Arthur, D. and Vassilvitskii, S. (2006). **k-means++: The advantages of careful seeding.** Technical report, Stanford.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). **A density-based algorithm for discovering clusters in large spatial databases with noise.** In kdd, volume 96, pages 226–231.

MacQueen, J. (1967). **Classification and analysis of multivariate observations.** In 5th Berkeley Symp. Math. Statist. Probability, pages 281–297. University of California Los Angeles LA USA.

Rahmah, N. and Sitanggang, I. S. (2016). **Determination of optimal epsilon (eps) value on dbscan algorithm to clustering data on peatland hotspots in sumatra.** In IOP conference series: earth and environmental science, volume 31, page 012012. IoP Publishing.

Sander, J., Ester, M., Kriegel, H.-P., and Xu, X. (1998). **Density-based clustering in spatial databases: The algorithm gdbscan and its applications.** Data mining and knowledge discovery, 2:169–194.

Steinhaus, H. (1957). **Sur la division des corps matériels en parties: Bulletin de l'académie polonaise des sciences.**