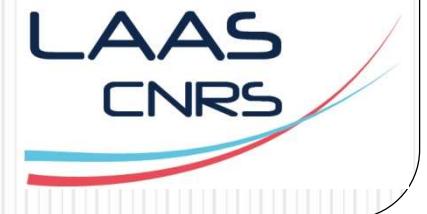


# UF Analyse descriptive et prédictive

M.-J. Huguet

<https://homepages.laas.fr/huguet>

2022-2023



# Analyse descriptive/préditive/prescriptive

## • Analyse descriptive

- Comprendre les données; les décrire; chercher régularités
  - Apprentissage non-supervisé
  - *Ex : pourquoi y-a-t-il des bouchons sur la route ?*

## • Analyse prédictive

- Identifier des règles de décision / données futures
  - Apprentissage supervisé
  - *Ex : quel sera le trafic dans 2 heures ?*
  - Ex : diagnostic médical - Description des patients – Prédiction d'une pathologie

## • Analyse prescriptive

- Quelles actions effectuer ?
  - *Ex : quel est le meilleur trajet en partant à 8h ?*



# Autres UF connexes

---

- **Infrastructures pour la gestion de données massives**
  - Concepts et techniques liés à la distribution et à la parallélisation des traitements
  - Concepts et techniques liés au stockage
  - Concepts et techniques liés à la virtualisation
- **Projet Intégrateur :**
  - IA et Big Data

# UF Analyse descriptive et prédictive

## - Tronc commun

---

- **Partie Analyse exploratoire – G. Trédan**
  - Méthodologie pour explorer/analyser un ensemble de données
  - Visualisation de données
  - TP en R
- **Partie Apprentissage non-supervisé – MJ. Huguet – M. Siala**
  - Principes généraux
  - Quelques méthodes : k-means, approches hiérarchiques, basées graphes
  - Quelques problèmes : fouille de données, réseaux sociaux
  - TP en Python (ScikitLearn)
- **Partie Apprentissage supervisé – MJ. Huguet – C. Labit-Bonis**
  - Suite cours de 4IR : Compléments sur familles de méthodes
  - Deep Learning – Applications sur des données de type Images
  - Enjeux en apprentissage : interprétabilité, équité, ...

# Plan

---

- **Introduction - Brefs rappels sur l'apprentissage**
- **Partie 1 - Apprentissage Non supervisé**
  - Clustering
  - Quelques problèmes en fouille de données
- **Partie 2 - Apprentissage Supervisé**



# L'apprentissage automatique

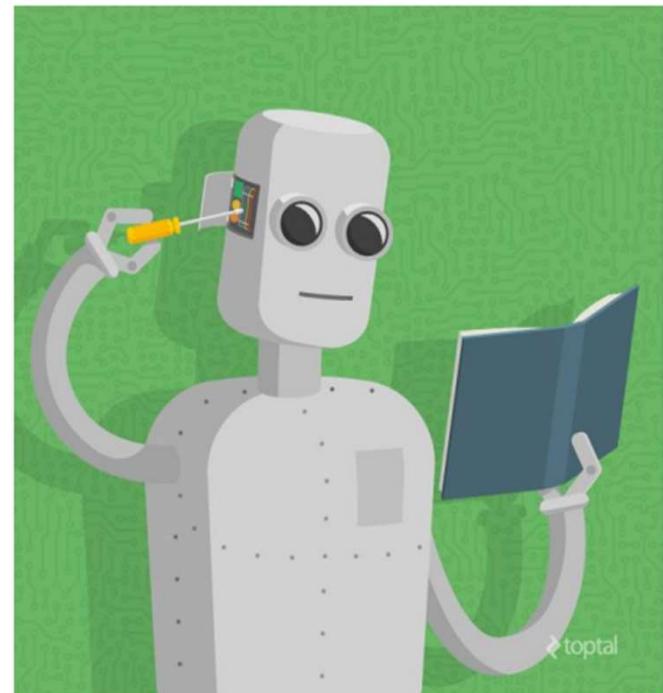
---

- **Brefs rappels sur l'apprentissage**

# Apprentissage automatique

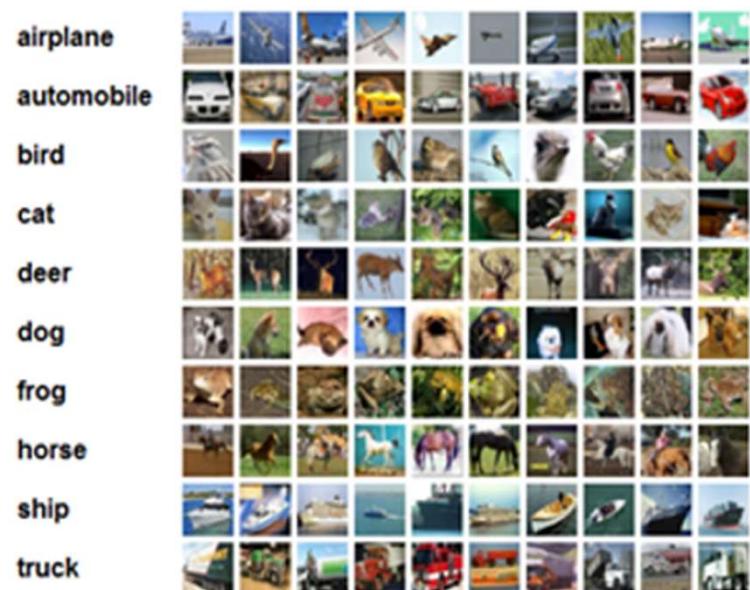
- **Une branche de l'Intelligence Artificielle**

- Capacité des ordinateurs d'apprendre à effectuer des tâches de classification ou de prédiction sans intervention humaine
- Autre dénomination:
  - Apprentissage artificiel
  - Apprentissage machine
  - « Machine Learning »
- **Different types d'apprentissage :**
  - Supervisé
  - Non supervisé
  - Par renforcement



# Apprentissage supervisé

- **Principe :**
  - On dispose d'un ensemble de données **annotées** par une valeur ou une classe **cible**
  - **Prédire** la réponse pour la cible sur de nouvelles données après un processus d'entraînement et de test
- **Illustration :**
  - Données d'entrée : Images
  - Cible : classe de chaque image
  - Mise en œuvre d'une méthode d'apprentissage supervisé
  - Résultat : un **modèle** représentatif des données d'entrée
  - Exploitation de ce modèle sur de nouvelles images pour prédire leur classe



# Vu en 4<sup>e</sup> année IR

<https://moodle.insa-toulouse.fr/course/view.php?id=1822>

---

- **Objectifs de l'apprentissage supervisé**

- Tâche de classification / Tâche de régression

- **Processus d'apprentissage**

- Données labellisés
  - Entrainement – Training set (éventuellement validation set)
  - Evaluation – Testing set
  - Métriques basées sur la matrice de confusion
    - Exemple : accuracy en entraînement / accuracy en test

- **Deux familles de méthodes d'apprentissage**

- Arbres de décision
  - Réseaux de neurones
  - (Hyper)-paramètres spécifiques de chaque méthode
    - Recherche des « bons » paramètres

# Apprentissage supervisé - Données

---

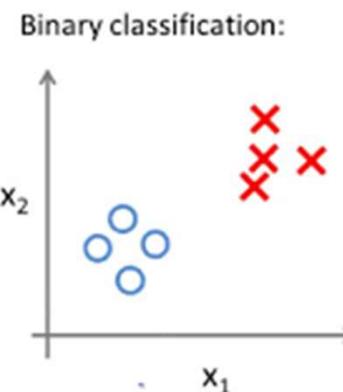
- **Jeux de données :**
  - Exemples utilisés pour l'entraînement et l'évaluation
  - **Attributs (features)** : caractéristiques numériques ou non associées à chaque exemple
  - **Etiquettes (labels)** : valeur ou classe cible associée à chaque exemple
- **Séparer les données :**
  - **Ensemble d'entraînement** (training set) : ensemble d'exemples utilisés pour la phase d'entraînement
  - *Ensemble de validation* (validation set) : ensemble d'exemples (parfois) utilisés pendant la phase d'entraînement
  - **Ensemble de test** (testing set) : ensemble d'exemples utilisés pour la phase d'évaluation
  - **Tous ces ensembles doivent être disjoints**

# Apprentissage supervisé - Tâches

- **Tâches d'apprentissage**

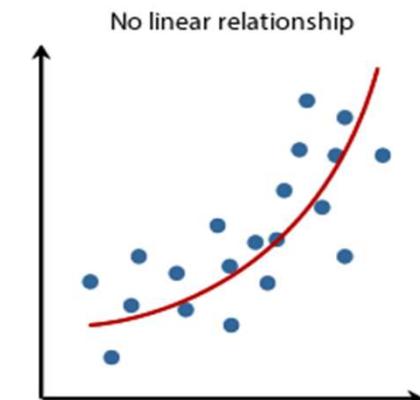
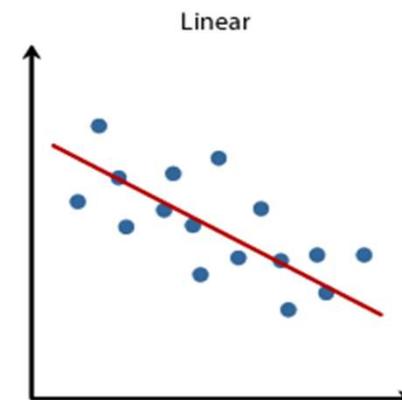
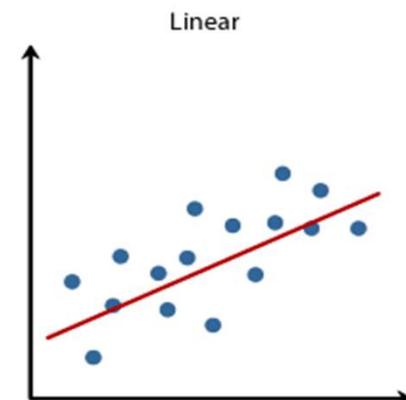
- **Classification**

- Les cibles sont des valeurs discrètes
    - Binaire
    - Multi-classes



- **Régression**

- Les cibles sont des valeurs réelles



# Illustration

## • Jouer au tennis ?

- Jeux de données tabulaires
  - Attributs :
    - Ciel, Température, Humidité, Vent
  - Label : Jouer
- Entrainement/Apprentissage
  - modèle de classification binaire
  - données d'entraînement
- Evaluation/Test
  - quelle classe pour (Soleil, 21.0, 95, Faible) ?
  - données de test
  - Capacité de généralisation du modèle
- Déployer le modèle
  - Nouvelles données

Ciel	Temperature	Humidite	Vent	Jouer
Soleil	27.5	85	Faible	Non
Soleil	25.0	90	Fort	Non
Couvert	26.5	86	Faible	Oui
Pluie	20.0	96	Faible	Oui
Pluie	19.0	80	Faible	Oui
Pluie	17.5	70	Fort	Non
Couvert	17.0	65	Fort	Oui
Soleil	22.5	70	Fort	Oui
Soleil	19.5	70	Faible	Oui
Pluie	22.5	80	Faible	Oui
Couvert	21.0	90	Fort	Oui
Couvert	25.5	75	Faible	Oui
Pluie	20.5	91	Fort	Non
Soleil	21.0	95	Faible	Non

# Apprentissage supervisé – Métriques performance

- **Mesures de performance**

- Matrice de confusion

		Labels calculés	
		True	False
Vrais labels	True	correct TP	erroné FN
	False	erroné FP	correct TN

- Exemple :

- Accuracy :  $\frac{TP+TN}{TP+FP+FN+F}$
- Précision, sensibilité, ....

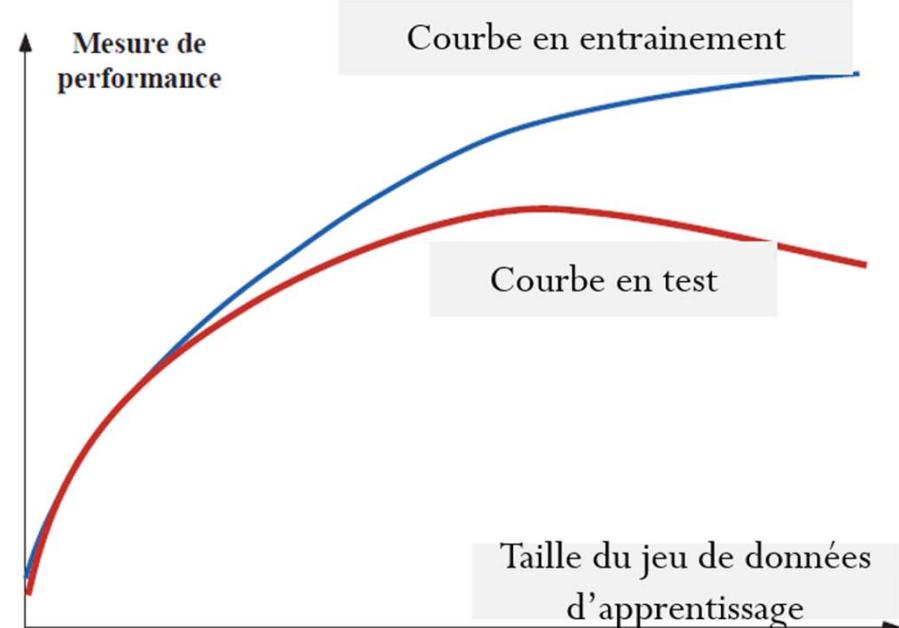
# Apprentissage supervisé – Métriques performance

- **Mesures de performance**

- **sous apprentissage** : le modèle appris est éloigné des données
- **sur-apprentissage** : le modèle se généralise mal sur les données de test

- **Validation croisée**

- Partition du jeu de données en  $k$  sous ensembles
  - $k-1$  pour l'entraînement ;
  - 1 pour l'évaluation
- répéter pour toutes les permutations
- Courbes : moyenne sur les  $k$  sous ensembles



# Apprentissage supervisé - Méthodes

---

- **NOMBREUSES méthodes**

- K plus proches voisins (k-NN) ~1980 -
- Méthodes de séparation linéaire ou non linéaire ~1980 -
- Réseaux de neurones et Deep Learning ~1980-201x
- Règles de décision (Arbres, listes, ensembles) ~1980 - 201x
- Ensemble learning (apprendre et combiner plusieurs modèles) ~1995 - 201x
- Basées sur différentes techniques d'optimisation

- **Exemples d'applications**

- Reconnaissance de formes
- Vision
- Détection de spams
- Traitement du langage

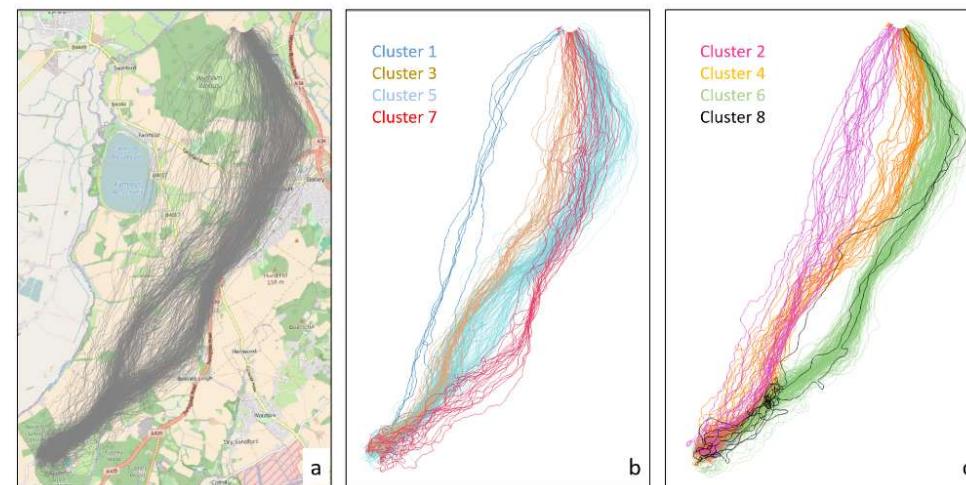
# Apprentissage non supervisé

- **Principe :**

- On dispose d'un ensemble de données représentées par différents attributs (features)
- Déterminer une structuration dans ces données
  - Regrouper des données similaires
  - Séparer des données différentes
  - Associer des informations

- **Illustration :**

- Données d'entrée : Traces mobilité
- Mise en œuvre d'une méthode d'apprentissage non supervisé
- Résultat : identification de trajets similaires (routes / horaires)
- Exploitation de ce modèle pour analyser le trafic, proposer des trajets alternatifs, ...

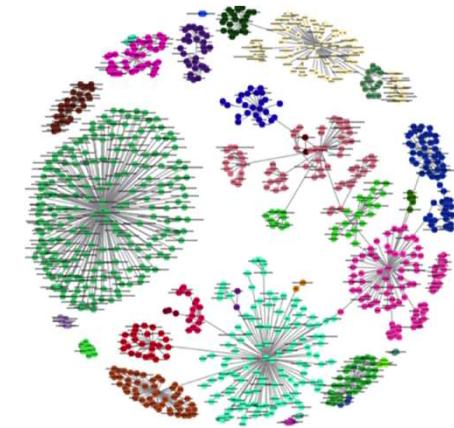
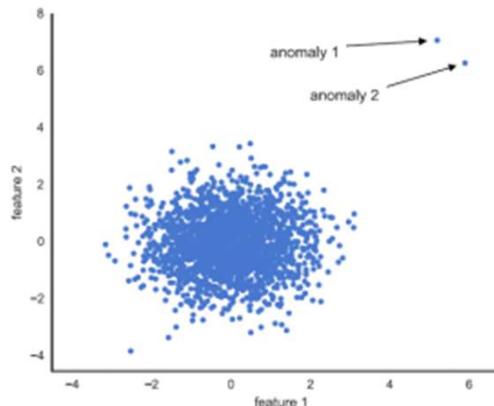


# Apprentissage non supervisé - Données

- **Ensemble de données non annotées**

- Partitionner les données : clustering

- Détection de données anomalies



- Motifs (séquentiels) fréquents

$((0, 0), \langle (1, A), (2, B), (3, C), (4, B), (5, D) \rangle),$   
 $((0, 1), \langle (1, B), (2, A), (3, C), (4, B), (5, B) \rangle),$   
 $((1, 0), \langle (1, D), (2, B), (3, C), (4, B), (5, C) \rangle),$   
 $((1, 1), \langle (1, C), (2, A), (3, C), (4, B), (5, A) \rangle)$



$((0, 0), \langle (1, A), (3, C), (4, B) \rangle),$   
 $((0, 1), \langle (2, A), (3, C), (4, B) \rangle),$   
 $((0, 1), \langle (2, A), (3, C), (5, B) \rangle),$   
 $((1, 1), \langle (2, A), (3, C), (4, B) \rangle)$

# Apprentissage non supervisé - Tâches

## • Clustering

- Illustration joueurs de tennis
  - Comment partitionner les données en 3 classes homogènes ?

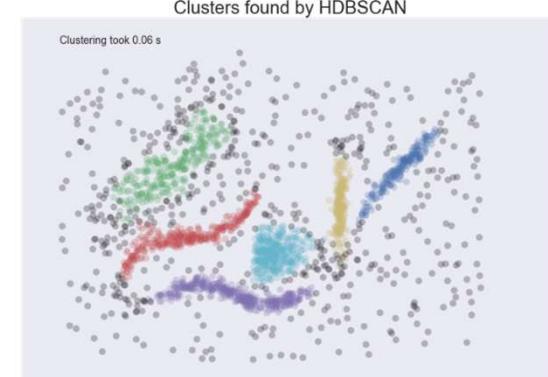
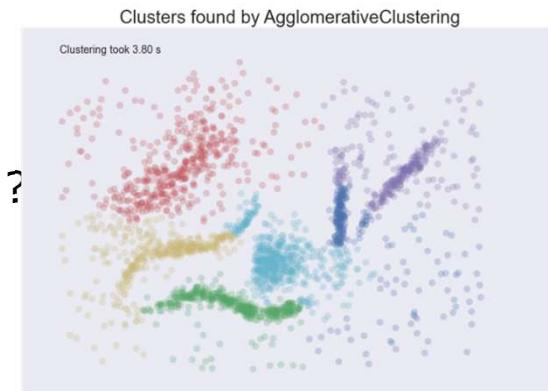
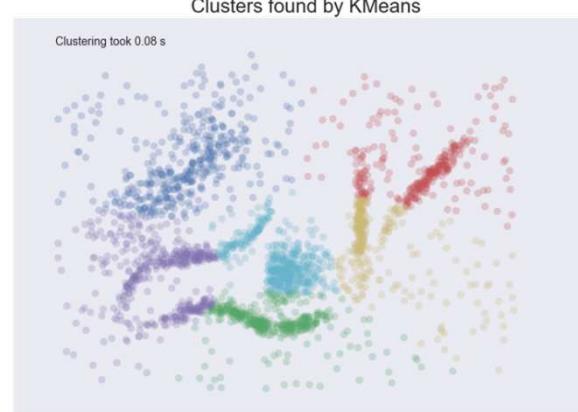
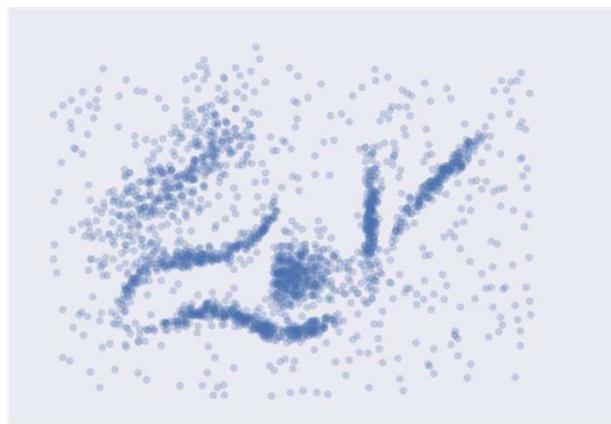
- Métriques de distance
  - Similarité intra cluster
  - Similarité inter-cluster
- Nombre de clusters

Ciel	Temperature	Humidite	Vent	Jouer
Soleil	27.5	85	Faible	Non
Soleil	25.0	90	Fort	Non
Couvert	26.5	86	Faible	Oui
Pluie	20.0	96	Faible	Oui
Pluie	19.0	80	Faible	Oui
Pluie	17.5	70	Fort	Non
Couvert	17.0	65	Fort	Oui
Soleil	22.5	70	Fort	Oui
Soleil	19.5	70	Faible	Oui
Pluie	22.5	80	Faible	Oui
Couvert	21.0	90	Fort	Oui
Couvert	25.5	75	Faible	Oui
Pluie	20.5	91	Fort	Non
Soleil	21.0	95	Faible	Non

# Apprentissage non supervisé – Métriques qualité

## • Mesures d'évaluation d'une solution

- Qualité des clusters obtenus
  - Bonne homogénéité ? Bien séparés ?
- Stabilité des résultats
  - Sensibilité à l'initialisation ?
- Cohérence par rapport à une expertise
  - Est-ce que le résultat fait sens pour l'application ?



# Apprentissage non supervisé - Méthodes

---

- **NOMBREUSES méthodes**

- k-Means ~1960 - 201x
- Méthodes hiérarchiques (agglomératives/divisives) ~1980-201x
- Méthodes basées densité
- Mean Shift ~1975
- DBSCAN ~1990, HDBSCAN ~2017
- etc.

- Basées sur différentes techniques d'optimisation

# Exemples de clustering

---

- **Quelques applications**

- Identifier des communautés dans des réseaux sociaux
- Identifier des clients avec un profil similaire
- Analyser des logs d'applications
- Analyser des textes, des emails
- Segmenter des images
- ....

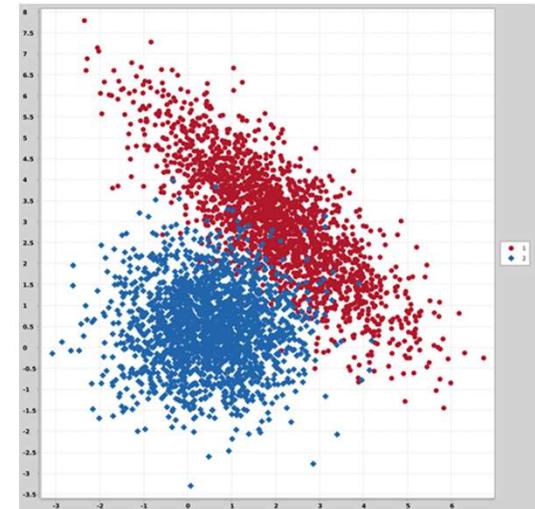
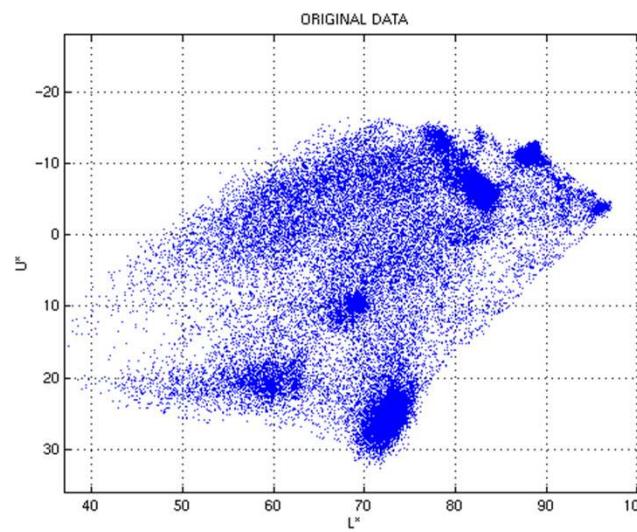
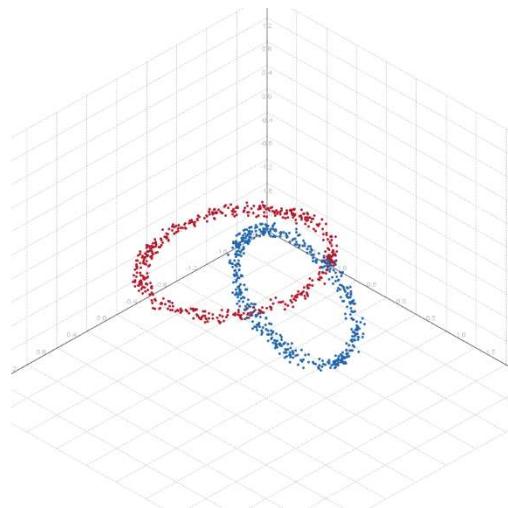
- **Fouille de données**

- Exploration et visualisation de données
- Motifs fréquents, règles d'association,
- Détection de communauté dans des réseaux sociaux
- Analyse de données de spatiales / temporelles
- Détection d'anomalies

# Apprentissage non supervisé

## • Difficultés

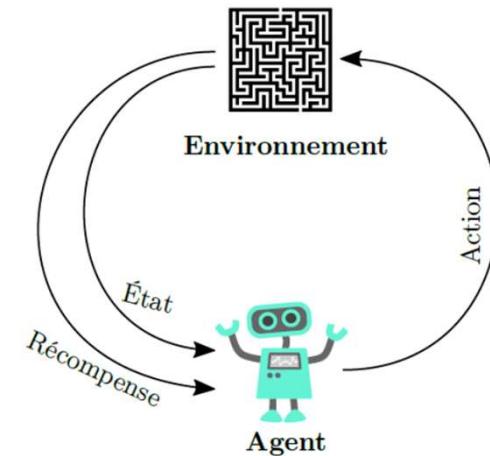
- Formes non convexes
- Présence de bruit dans les données
- Densité variable



# Apprentissage par renforcement

## • Principe

- **Données** : une séquence de perceptions, d'actions et de récompenses
- **Perception** : information sur l'état de l'environnement
- **Action** : décision prise et mise en œuvre
- **Récompense** : valide ou invalide une action



- Problème posé :
  - Dans une situation donnée, quelle action choisir pour maximiser un gain à long terme

# Apprentissage par renforcement

---

- Déterminer un comportement à partir d'expériences
  - Trouver une stratégie (politique) :
    - action à appliquer à partir d'un état
  - Formalisme des processus de décision markovien :
    - L'état suivant ne dépend que de l'état courant et de l'action
      - pas d'historique
- Réaliser un compromis entre exploration et exploitation
  - Exploration : tenter de nouvelles actions
  - Exploitation : refaire des actions car on sait qu'elles apportent de bonnes récompenses
- Méthodes d'évaluation :
  - Exacte ou approchées (échantillonnage)
  - Dimension de l'espace (état, action)

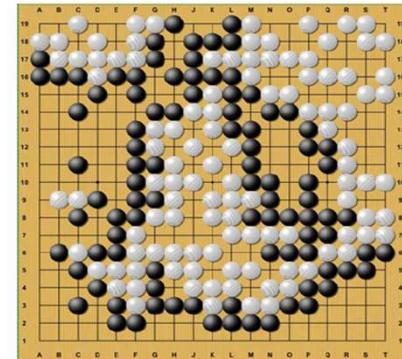
# Apprentissage par renforcement

- Applications :
  - Contrôle optimal
  - Robotique
  - Jeux

- Exemple :
  - 2013 : Jeux Atari
  - 2017 : AlphaGoZero
    - Apprentissage par renforcement : jeu en auto-apprentissage (self play)
    - Réseaux de neurones profonds pour prédire les probabilités de résultats
  - 2017 : AlphaZero : généralisation pour d'autres jeux
  - ... Jeux Atari
  - 2019 : jeu temps réel (StarCraft 2)



$10^{120}$  possibilités



$10^{761}$  possibilités

# Plan

- 
- **Introduction - Brefs rappels sur l'apprentissage**
  - **Partie 1 - Apprentissage Non supervisé**
    - Problème de Clustering
    - // Quelques problèmes en fouille de données
  - **Partie 2 - Apprentissage Supervisé**



# Plan

---

## **1. Caractérisation du problème de clustering**

1. Données
2. Distances
3. Problème de partition
4. Synthèse

## **2. Quelques Méthodes**

1. Méthodes basées centres de masses
2. Méthodes hiérarchiques
3. Méthodes basées voisinage (densité)
4. Méthodes basées graphes

## **3. Bilan Clustering**

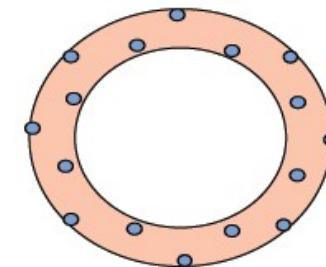
1. Evaluation d'un clustering
2. Application

# Tâche d'apprentissage en non supervisé

- **Apprentissage non supervisé**

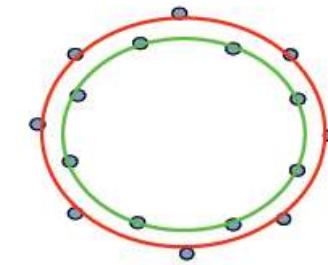
- Ensemble de données non annotées (sans étiquette)
- Nombre et nature des classes inconnus

- Rechercher une structure dans les données
  - Partitionner les exemples en **clusters**/classes
    - Clustering / Partitionnement

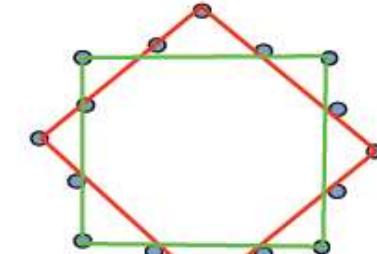


- **Qualité d'un bon clustering ?**

- **Homogénéité** : les éléments d'un même cluster sont similaires
- **Séparation** : les éléments de différents clusters sont différents



- Le résultat dépend beaucoup de la modélisation du problème et de la distance ...



# Nature des données

- **Quantitatives / Numériques**

- Discrètes
- Continues

- **Qualitatives / Catégoriques**

- Binaires
- Nominales (pas de relation d'ordre)
  - Ex : bleu, jaune, rouge
- Ordinales (relation d'ordre)
  - Ex : petit; moyen; grand

Données  
Tabulaires

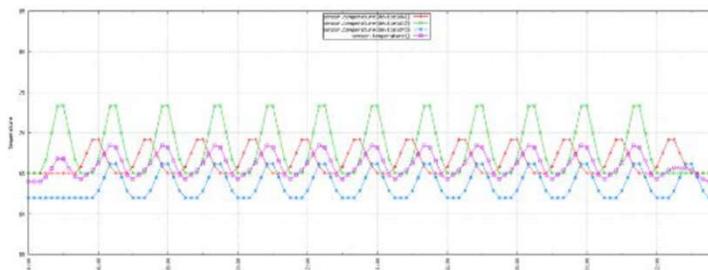
- **Séries Temporelles**

- **Texte**

- **Images**

- **Vidéo**

...



# Données numériques

## • Pré-traitement des données

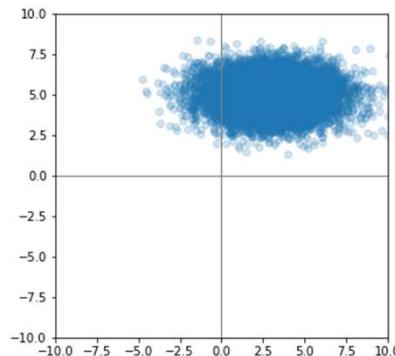
- Redimensionner les valeurs numériques : échelle comparable
- **Centrer** : Ramener les données dans l'intervalle [0,1]

$$\bullet \quad x' = \frac{x - \min}{\max - \min}$$

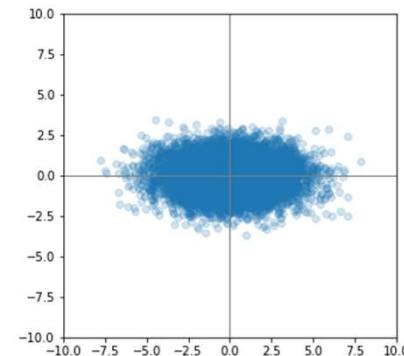
## • Réduire par rapport à l'écart type

$$\bullet \quad x'' = \frac{x' - \text{moy}}{\text{ecart-type}}$$

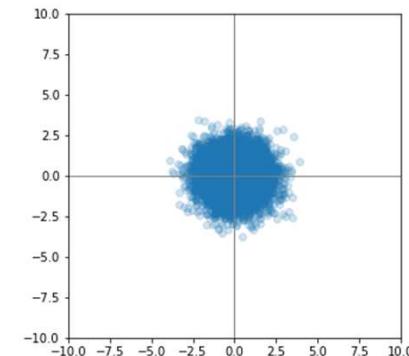
- Données indépendantes de l'échelle de mesure
- Attributs avec même moyenne et dispersion



Données initiales



center  
(soustraire la  
moyenne)



réduire  
(diviser écart  
type)

# Jeux de données

- **Données tabulaires**

- Un ensemble  $X = \{x_i\}$  de  $n$  **exemples**
- Un exemple
  - est composée de  $d$  **attributs** (caractéristiques; features)
- Exemple : extrait du jeu de données « wine »

attributs

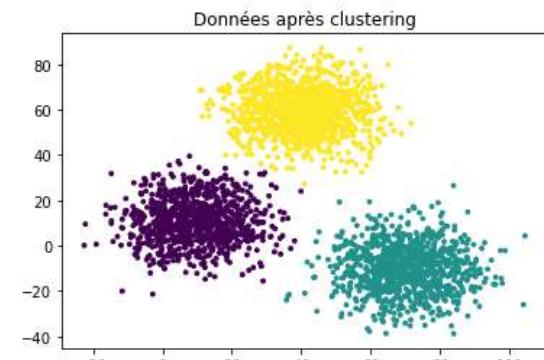
fixed_acidity	density	pH	sulphates	alcohol	quality	style
7.4	0.9978	3.51	0.56	9.4	5	red
7.8	0.9968	3.2	0.68	9.8	5	red
7.8	0.997	3.26	0.65	9.8	5	red
6.2	0.9964	3.33	0.6	9.6	6	white
7.9	0.9992	3.03	0.46	9.2	6	white
6.9	0.9948	3.16	0.72	10.7	7	white
7.3	0.9998	3.36	0.41	9.1	7	white

exemples



# Objectif du clustering

- **Cluster**
  - groupe d'exemples similaires (et faisant sens)
- **Clustering**
  - Déterminer des clusters
  - Les exemples similaires sont regroupés dans un même cluster
  - Les exemples différents appartiennent à des clusters différents
  - Problème d'optimisation
    - Maximiser la similarité intra-cluster
    - Minimiser la similarité inter-cluster
  - S'appuie sur une mesure de similarité / dissimilarité entre exemples d'un jeu de données



# Plan

---

## **1. Caractérisation du problème de clustering**

1. Données
2. Distances
3. Problème de partition
4. Synthèse

## **2. Quelques Méthodes**

1. Méthodes basées centres de masses
2. Méthodes hiérarchiques
3. Méthodes basées voisinage (densité)
4. Méthodes basées graphes

## **3. Bilan Clustering**

1. Evaluation d'un clustering
2. Application

# Mesures de similarité / dissimilarité

## • Distance

- Jeu de données

	att_1	att_2		att_d
1				
2				
3				
n				

- Distance : mesure de dissimilarité entre paire d'éléments :

**Distance** = dissimilarité  
- 0 si éléments proches  
**Similarité**  
- 0 si éléments éloignés

	1	2	3		n
1	0				
2	$\delta(2,1)$	0			
3	$\delta(3,1)$	$\delta(3,2)$	0		
n	$\delta(n,1)$	$\delta(n,2)$	$\delta(n,3)$		0

# Mesures de similarité / dissimilarité

---

- **Distance**

- Fonction calculée entre des paires d'éléments  $\delta: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^+$  vérifiant :
  - Symétrie :  $\delta(x_1, x_2) = \delta(x_2, x_1)$
  - Séparation :  $\delta(x_1, x_2) = 0 \Leftrightarrow x_1 = x_2$
  - Inégalité triangulaire :  $\delta(x_1, x_2) \leq \delta(x_1, z) + \delta(z, x_2)$
- Différentes métriques de distance selon le type des données
- Données en plusieurs dimensions :
  - Pondération potentielle des dimensions (selon jeu de données / applications)
- Première difficulté pour une méthode de clustering :
  - Définir proximité entre éléments d'un jeu de données

# Distances

## • Attributs binaires

### • Indice de Jaccard

- Similarité entre deux ensembles :  $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$
- **Similarité** entre 2 exemples représentés par des attributs binaires
- Pour un attribut binaire dans 2 exemples :
  - 4 combinaisons de valeurs : 00, 01, 10, 11
- Compter le nombre d'apparition de chaque combinaison

	1	0
1	$n_{11}$	$n_{10}$
0	$n_{01}$	$n_{00}$

- Avec  $d = n_{11} + n_{01} + n_{10} + n_{00}$
- Indice de Jaccard :  $J = \frac{n_{11}}{n_{11} + n_{10} + n_{01}} = \frac{n_{11}}{d - n_{00}}$
- **Distance de Jaccard** :  $J_\delta = 1 - J$

### • Exemple

- $x_1 = (1, 1, 0, 1, 0)$
- $x_2 = (1, 0, 0, 0, 1)$ 
  - $n_{11} = 1; n_{10} = 2$
  - $n_{01} = 1; n_{00} = 1$
- $J(x_1, x_2) = \frac{1}{4}$
- $J_\delta = 1 - \frac{1}{4} = \frac{3}{4}$
- $x_1 = (1, 1, 0, 1, 0)$
- $x_3 = (0, 0, 1, 0, 1)$ 
  - $n_{11} = 0; n_{10} = 3$
  - $n_{01} = 2; n_{00} = 0$
- $J(x_1, x_3) = 0$
- $J_\delta = 1 - 0 = 1$

# Distances

## • Attributs binaires

- Variantes (nombreuses) basées sur l'énumération des 4 combinaisons

	1	0
1	n_11	n_10
0	n_01	n_00

- Exemple :  $\frac{n_{11}}{n_{11}+n_{10}+n_{01}+n_{00}}$  (Russel & Dao, 1940)
- Extension à plus de 2 états possibles
  - Données nominales (couleurs)

# Distances

- **Attributs numériques**

- Distance de Minkowski ou Norme  $L_q$

- $\bullet \quad \delta(x_1, x_2) = ||x_2 - x_1||_q = \sqrt[q]{\sum_{i=1}^d |x_{1,i} - x_{2,i}|^q}$

- Si  $q = 2$ , distance euclidienne

- $\circ \quad \delta(x_1, x_2) = ||x_2 - x_1|| = \sqrt{\sum_{i=1}^d |x_{1,i} - x_{2,i}|^2}$

- Si  $q = 1$ , distance de Manhattan

- $\circ \quad \delta(x_1, x_2) = \sum_{i=1}^d |x_{1,i} - x_{2,i}|$

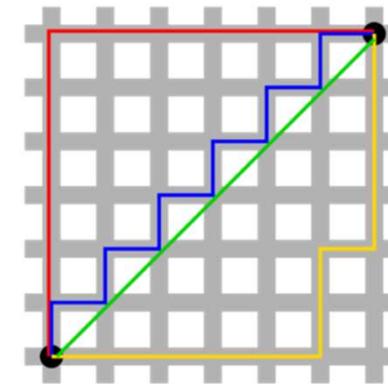


Image:Wikipedia

# Autres distances

## • Distance de Hamming

- Mesurer différence entre deux séquences de symboles
  - Traitement du signal

- Soit  $x_i$  et  $y_i$  deux observations de dimension  $d$

- Distance de Hamming :

- $h(x_i, y_i) = \text{Card}(\{j : x_{ij} \neq y_{ij}\})$

$$x_i = \begin{pmatrix} x_{i,1} \\ x_{i,2} \\ \vdots \\ x_{i,d} \end{pmatrix}$$

- Exemples

- $x_1 = (1, 1, 0, 1, 0)$  et  $x_2 = (1, 0, 0, 0, 1)$  : distance de hamming :  $h(x_1, x_2) = 3$
  - $x_1 = 24836$  et  $x_2 = 34856$  : distance de hamming :  $h(x_1, x_2) = 2$

# Autres distances

---

- **Distance de Levenshtein (distance d'édition)**

- Distance entre deux chaînes de caractères
  - Nombre d'opérations élémentaires (insérer/supprimer/remplacer) pour passer d'une chaîne source à une chaîne destination
  - Exemple :
    - Passer de "a" vers "ab" : distance = 1 (insérer 'b')
    - Passer de "aab" vers "bab" : distance = 1

# Plan

---

## **1. Caractérisation du problème de clustering**

1. Données
2. Distances
3. Problème de partition
4. Synthèse

## **2. Quelques Méthodes**

1. Méthodes basées centres de masses
2. Méthodes hiérarchiques
3. Méthodes basées voisinage (densité)
4. Méthodes basées graphes

## **3. Bilan Clustering**

1. Evaluation d'un clustering
2. Application

# Clustering et problème de partition d'un ensemble

- **Le problème de partition :**
  - Décomposer un ensemble  $X$  en sous-ensembles non vides tel que chaque élément  $x \in X$  se retrouve dans **un et un seul sous ensemble**
  - Soit  $P$  une famille d'ensembles :  $P$  est une **partition** de  $X$ ssi
    - L'ensemble vide n'est pas dans  $P$  :  $\emptyset \notin P$
    - L'union des ensembles de  $P$  vaut  $X$  :  $\bigcup_{A \in P} A = X$
    - Les ensembles de  $P$  sont deux à 2 disjoints :  $\forall A, B \in P : A \neq B \Rightarrow A \cap B = \emptyset$
  - **Exemple**
    - $X = \{a, b, c\}$ . Il existe 5 partitions :
      - $\{\{a\}, \{b\}, \{c\}\}; \quad \{\{a, b\}, \{c\}\}; \quad \{\{a, c\}, \{b\}\}; \quad \{\{b, c\}, \{a\}\}; \quad \{\{a, b, c\}\}$
    - Ne sont pas des partitions de  $X$  :
      - $\{\{\}, \{a, b\}, \{c\}\}; \quad \{\{a, b\}, \{b, c\}\}; \quad \{\{a\}, \{b\}\};$

# Dénombrer le nombre de partitions

- **Nombre de partitions d'un ensemble  $X$  en  $k$  sous-ensembles :**
  - Exemple : pour un ensemble  $X = \{a, b, c\}$  de taille 3
    - Il existe 5 partitions différentes
    - *1 partition en 3 sous-ensembles (3 clusters) :*
      - $\{\{a\}, \{b\}, \{c\}\}$
    - *3 partitions en 2 sous ensembles (2 clusters) :*
      - $\{\{a, b\}, \{c\}\}$
      - $\{\{a, c\}, \{b\}\}$
      - $\{\{b, c\}, \{a\}\}$
    - *1 partition en 1 sous-ensemble (1 cluster) :*
      - $\{\{a, b, c\}\}$
  - Il y a donc 5 solutions de clustering pour cet exemple
  - Et, pour un nombre de clusters fixés, il y a plusieurs solutions

# Nombre de solutions de clustering : Dénombrer le nombre de partitions

---

- Si on connaît la valeur de  $k$  (**clusters**) :
  - Combien de partitions d'un ensemble de taille  $n$  en  $k$  sous-ensembles ?
    - On note  $n$  le nombre d'éléments de  $X$  et  $k$  le nombre de clusters
    - Nombre de Stirling :  $S(n, k)$  : nombre de partitions en  $k$  sous-ensembles
      - Equations de récurrence :
        - $S(n, k) = S(n - 1, k - 1) + k \times S(n - 1, k)$
        - $S(n, 1) = 1$
        - $S(n, n) = 1$
      - Application :
        - $S(3,1) = 1 \rightarrow$  1 seule partition en  $k = 1$  cluster (sous-ensemble)
        - $S(3,2) = S(2,1) + k \times S(2,2)$
        - $S(3,2) = 1 + 2 \times 1 = 3 \rightarrow$  3 partitions en  $k = 2$  clusters (sous-ensembles)

# Nombre de solutions de clustering : Dénombrer le nombre de partitions

---

- **Si on ne connaît pas la valeur de  $k$  (clusters) :**

- **Nombre total de partitions**

- **Nombre de Bell :**

- Somme du nombre de Stirling pour toutes les valeurs possibles de  $k$

- $B(n) = \sum_{k=1}^n S(n, k)$

- Application :

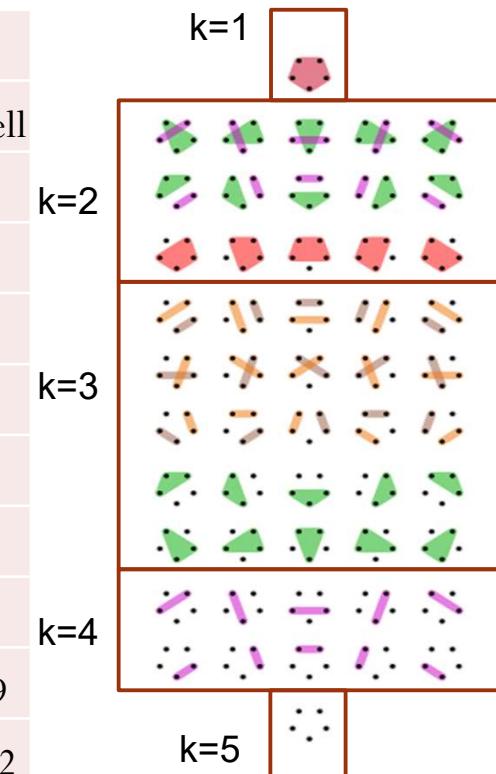
- $B(3) = S(3,1) + S(3,2) + S(3,3) = 1 + 3 + 1 = 5$

# Nombre de solutions de clustering : Dénombrer le nombre de partitions

- Illustration des calculs nombre de Stirling  $S(n, k)$  et nombre de Bell  $B(n)$

n	k										Nb Bell
	1	2	3	4	5	6	6	8	9	10	
1	1										1
2	1	1									2
3	1	3	1								5
4	1	7	6	1							15
5	1	15	25	10	1						52
6	1	31	90	65	15	1					203
7	1	63	301	350	140	21	1				877
8	1	127	966	1701	1050	266	27	1			4139
9	1	255	3025	7770	6951	2646	428	35	1		21112
10	1	511	9330	34105	42525	22827	5214	708	44	1	115266

$$S(5,3) = S(4,2) + 3 \times S(4,3) = 7 + 3 \times 6 = 25$$



# Clustering : trouver une « bonne » partition

- **Clustering :**

- Pour une valeur de  $k$  fixé :

- Comment déterminer la meilleure partition ?

- **Difficultés :**

- Nombre de solutions : nombre de Stirling – Combinatoire élevée !
    - Quelles sont les métriques de qualité pour comparer 2 partitions ?
    - Est-ce que la valeur fixée est pertinente ?



- Pour une valeur de  $k$  non fixé :

- Comment déterminer la meilleure partition ?

- **Difficultés :**

- Nombre de solutions : nombre de Bell – Combinatoire élevée !
    - Quelles sont les métriques de qualité pour comparer 2 partitions ?



# Plan

---

## **1. Caractérisation du problème de clustering**

1. Données
2. Distances
3. Problème de partition
4. **Synthèse**

## **2. Quelques Méthodes**

1. Méthodes basées centres de masses
2. Méthodes hiérarchiques
3. Méthodes basées voisinage (densité)
4. Méthodes basées graphes

## **3. Bilan Clustering**

1. Evaluation d'un clustering
2. Application

# Problème de clustering : synthèse

---

- **Problème :**
  - Un ensemble  $X = \{x_i\}$  de  $n$  exemples avec  $d$  attributs
  - Déterminer une métrique de distance / similarité entre exemples
- **Sélectionner une méthode pour obtenir un clustering**
  - Problème d'optimisation : un ou deux objectifs
    - Maximiser la similarité entre exemples intra-cluster
    - Minimiser la similarité inter-cluster
- **Evaluer le résultat de clustering**

# Méthodes de clustering

---

- **Méthodes exactes**

- Optimum global : Enumération et évaluation des partitions pour déterminer la plus pertinente
  - Explosion combinatoire du nombre de partitions
- Problème d'optimisation posé en général NP-difficile

- **Méthodes approchées ou heuristiques**

- Optimum local : exploration d'un sous-ensemble de partitions
- Très nombreuses méthodes/variantes dans la littérature
- Méthodes générales ou spécifiques pour un domaine d'application
- Exploitent une fonction objectif +/- complexe

- **Méthodes paramétriques / non paramétriques**

- Nombre de clusters imposés / non imposés

# Méthodes de clustering

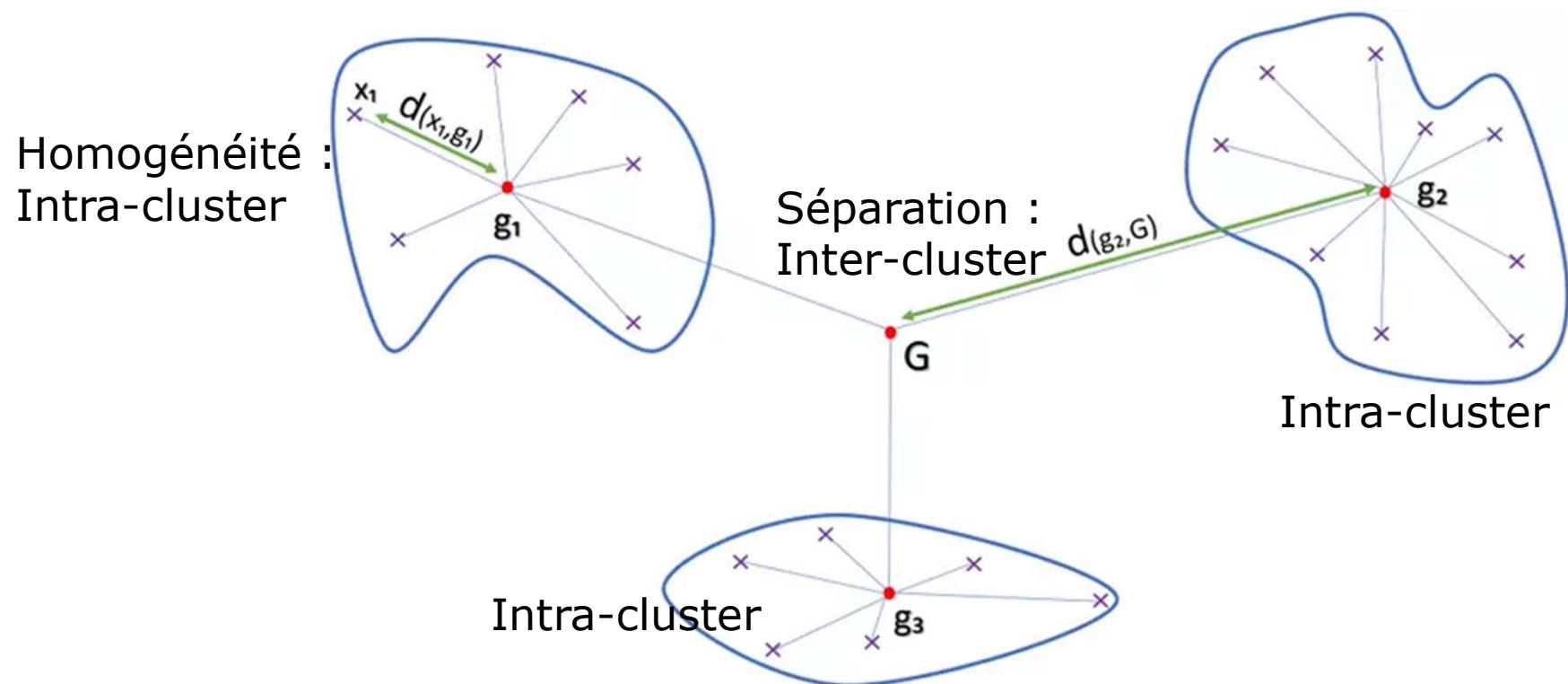
- 
- **Différents types de méthodes approchées ou heuristiques**
    - Basées sur centres de « masse »
      - Chercher des centres de masse et regrouper les exemples autour
    - Basées hiérarchies
      - Agglomération : regrouper par proximité (similarité)
      - Division : Séparer par éloignement (di-similarité)
    - Basées densité
      - Regrouper les exemples en zones denses séparées par des zones de plus faible densité

# Evaluation d'un clustering

- 
- **Forme des clusters**
    - Évaluation de la **qualité** des clusters / distance
    - Attention : **il n'y a pas de labels** pour vérifier !
  - **Differentes mesures permettant d'exprimer la similarité**
    - **Homogénéité** : les éléments d'un même cluster sont similaires
    - **Séparation** : les éléments de différents clusters sont différents
  - **Stabilité des clusters**
    - Insensibilité à l'ordre des traitement des données
    - Les mêmes points sont-ils toujours dans le même cluster ?
    - Aide pour fixer le nombre de clusters
  - **Cohérence / expertise**
    - Évaluation par expert humain ...

# Evaluation d'un clustering

- Exemple : inertie intra et inter clusters



# Plan

---

## **1. Caractérisation du problème de clustering**

1. Données
2. Distances
3. Problème de partition
4. Synthèse

## **2. Quelques Méthodes**

1. Méthodes basées centres de masses
2. Méthodes hiérarchiques
3. Méthodes basées voisinage (densité)
4. Méthodes basées graphes

## **3. Bilan Clustering**

1. Evaluation d'un clustering
2. Application

# Méthode k-means (1)

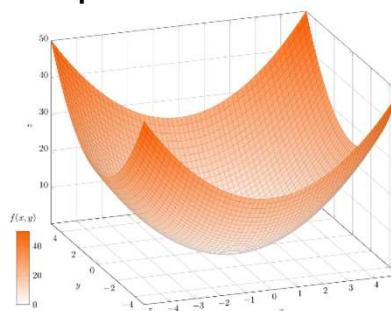
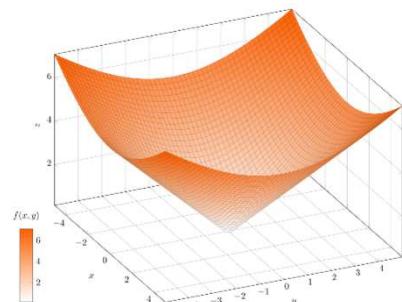
- 
- **Méthode approchée (Lloyd, 1982)**
    - Objectif : Minimiser la distance intra-cluster uniquement
      - distance moyenne des exemples au centre de leur cluster
        - Données numériques : carré de la distance euclidienne → aparté
    - **Entrée** : le nombre de clusters à obtenir
  - **Principe**
    - Placer  $K$  centres de gravité qui vont correspondre aux centres des différents clusters
    - Placer les données dans les clusters / centres de gravité choisis
    - Mettre à jour les centres de gravités
    - Recommencer jusqu'à stabilisation
    - Méthode des k-means ou méthode des centres mobiles

# Apparté

## • Carré de la distance euclidienne

- Euclidian Distance :  $\delta(x_1, x_2) = \|x_2 - x_1\| = \sqrt{\sum_{i=1}^d |x_{1,i} - x_{2,i}|^2}$
- Squared Euclidian Distance :  $\delta(x_1, x_2)^2 = \|x_2 - x_1\|^2 = \sum_{i=1}^d (x_{1,i} - x_{2,i})^2$
- Métrique fréquente en mathématique et en optimisation
  - Méthode des moindres carrés (régression)
  - En clustering : donne plus d'importance aux distances les plus grandes
- N'est pas un espace métrique:
  - Propriété d'inégalité triangulaire non valide
- Bonne propriété pour les méthodes d'optimisation convexe

A cone, the graph of Euclidean distance from the origin in the plane

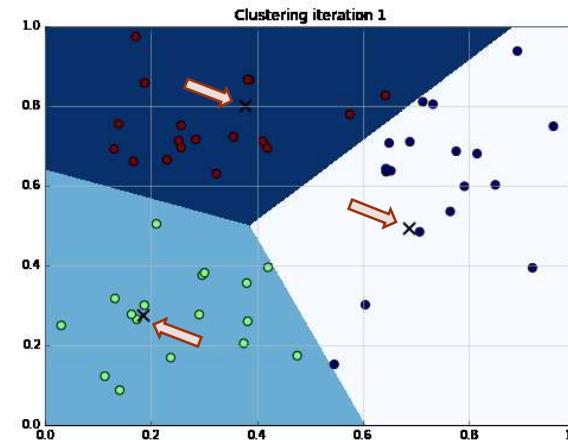
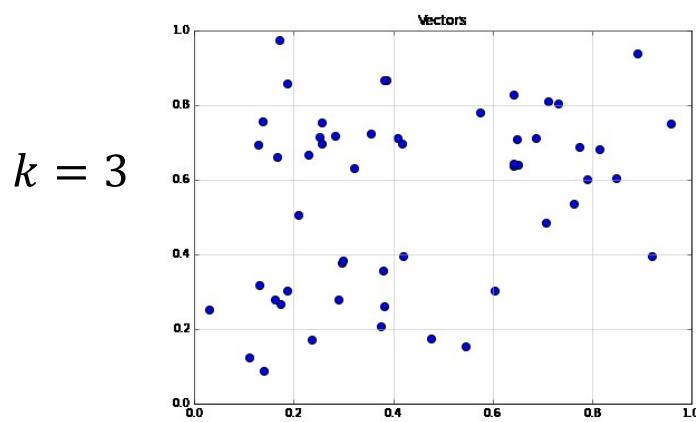


A paraboloid, the graph of squared Euclidean distance from the origin

# Méthode k-means (2)

## • Principale (1)

- Placer  $K$  centres de gravité (les centres des différents clusters)
  - Un centre n'est pas forcément un exemple du jeu de données
- Placer les données dans les clusters / centres de gravité choisis



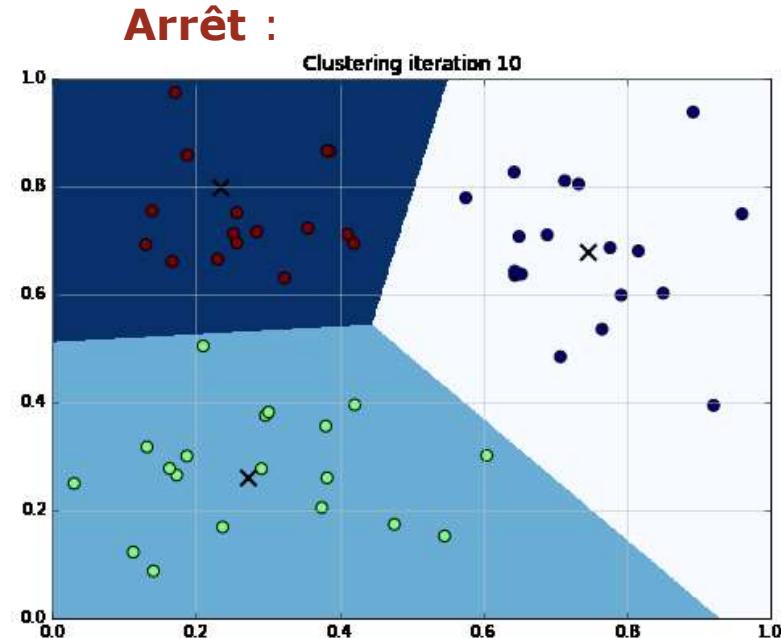
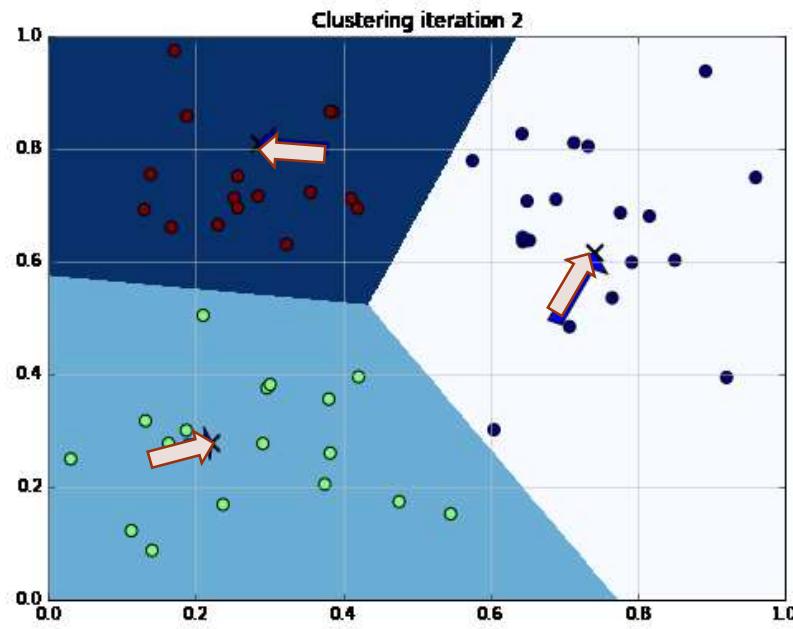
- Cluster d'un exemple donné : celui qui correspond au centre le plus proche
- Principe d'une méthode gloutonne : obtention d'un clustering
- Différentes solutions en fonction de l'initialisation des centres

La méthode ne s'arrête pas là ...

# Méthode k-means (3)

## • Principale (2)

- Faire évoluer la solution initiale obtenue pour améliorer l'inertie du clustering (recherche locale)
  - Mettre à jour les centres de gravité
  - Déterminer nouvelle allocation des exemples aux nouveaux centres
- Arrêt quand stabilisation des centres (et donc des clusters)



# Algorithme k-means

- **Mesure de distance : inertie intra-cluster**

- $\sum_{k=1}^K \frac{1}{n_k} \sum_{i \in C_k} \| (x_i - \mu_k) \|^2$  (min Squared Error – Inertie)

- **Algorithm :**

- Initialisation
  - Choisir  $k$  éléments (centres) :  $\{\mu_1, \dots, \mu_k\}$
  - Les placer dans un cluster :  $C_i \leftarrow \mu_i$

- Répéter
  - Affecter chaque élément au centre le plus proche
    - $C_l = C_l \cup \{x_i\}$  tel que  $l = \operatorname{argmin}_k (d^2(x_i, \mu_k))$
  - Re-évaluer le centre de gravité de chaque cluster
    - $\mu_k = \frac{1}{n_k} \sum_{i \in C_k} x_i$

- Jusqu'à : // Conditions d'arrêt //

- **Arrêt :**

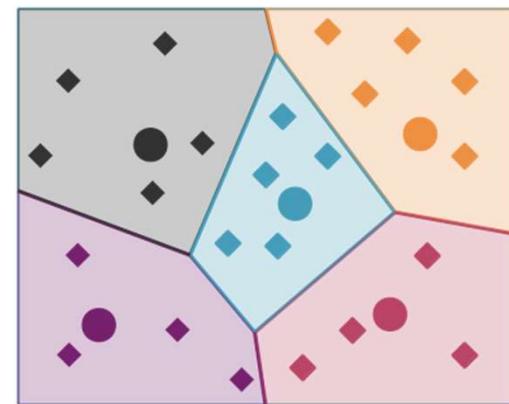
- Nombre itérations ; centres stables ; allocation des exemples stables

- **Convergence :**

- Diminution de la fonction objectif

# Caractéristiques méthode k-means (1)

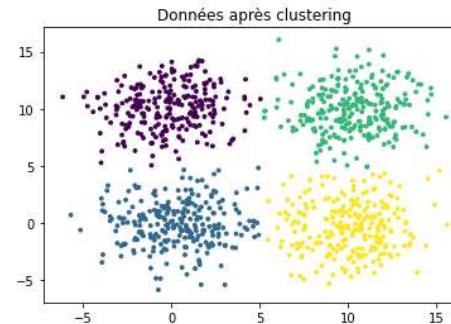
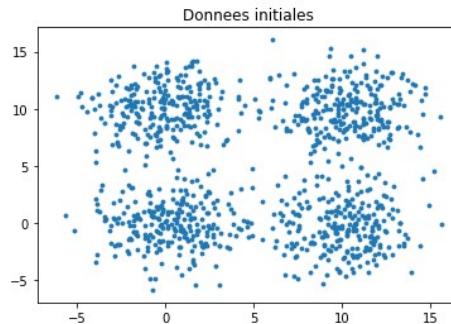
- **En entrée de la méthode**
  - Nécessite de fixer le nombre de clusters
  - Nécessite l'existence d'une distance (euclidienne)
- **Stratégie gloutonne :**
  - Obtention d'un minimum local / objectif
  - Faible complexité : passage à l'échelle
  - Compréhension simple de la méthode
- **Forme des clusters**
  - Formes convexes
    - Chaque point d'un cluster est plus proche de son centre de gravité que des autres centres
  - Pas adapté pour déterminer des clusters avec des formes non convexes



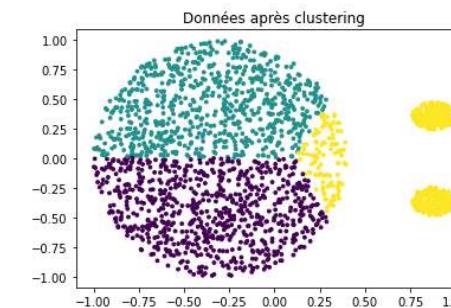
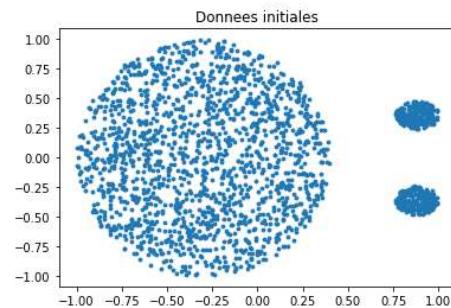
# Caractéristiques méthode k-means (2)

## • Sensibles aux données bruitées et aux anomalies

- Tous les exemples doivent être inclus dans un cluster
- Données non pertinentes pouvant influencer la valeur moyenne
  - 2013 : S. Chawla, A. Gionis (2013) K-means - - A unified approach to clustering and outlier detection. *SIAM International Conference on Data Mining* pp. 189--197



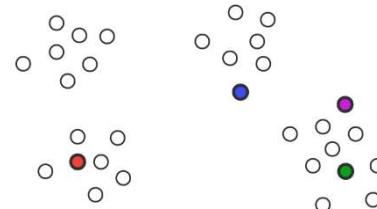
## • Sensibilité aux variations de densité



# Caractéristiques méthode k-means (3)

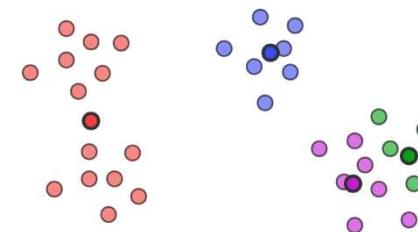
- **Sensible au choix des points initiaux**

- Fort impact sur le résultat



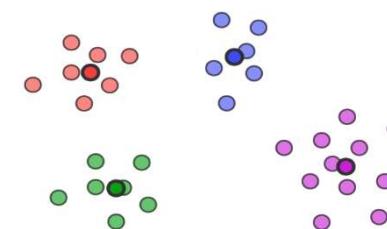
- Aléatoire

- Faire plusieurs exécutions avec différentes initialisations et conserver la meilleure solution



- K-means++

- Choix des centres **avec une probabilité** liée à la distance au carré aux autres centres
    - Garanties / qualité du résultat par rapport à l'aléatoire (article 2007)



# Caractéristiques méthode k-means (3)

---

- **Choisir un nombre de clusters**
  - Post-processing : Split & Merge
    - Découper un cluster quand sa variance est supérieure à un seuil
    - Regrouper deux clusters quand la distance entre leurs centres est inférieure à un seuil
  - Sélection
    - Fixer un nombre supérieur de centres et sélectionner parmi ceux-ci les centres conduisant à des clusters les plus séparables

# Caractéristiques problème k-means

---

- **Complexité**
  - Problème NP-difficile (même pour 2 clusters)
  - Si  $k$  et  $d$  sont fixés : polynôme en puissance de  $d$  et de  $k$
- **Méthodes approchées**
  - **Plusieurs implémentations :**
    - 1982 - algorithme de Lloyd en  $O(n \cdot d \cdot k \cdot nb_{iter})$
    - Sortir des optima locaux : mouvements inter-clusters
    - Autres types de données / distances
    - Métaheuristiques
      - 1999 : K. Krishna, M.N. Murty (1999). "Genetic k-means algorithm". IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics. 29 (3): 433–439
      - 2001 : P. Hansen, N. Mladenovic (2001). "J-Means: A new local search heuristic for minimum sum of squares clustering". Pattern Recognition. 34 (2): 405–413.
      - 2019 : D. Gribel, T. Vidal (2019). "HG-means: A scalable hybrid metaheuristic for minimum sum-of-squares clustering". Pattern Recognition. 88: 569–583.
- **Méthodes exactes**
  - 2022 : V. Piccialli, A. M. Sudoso, A. Wiegle (2022) SOS-SDP: An Exact Solver for Minimum Sum-of-Squares Clustering. INFORMS Journal on Computing 34(4):2144-2162.

# Exemple

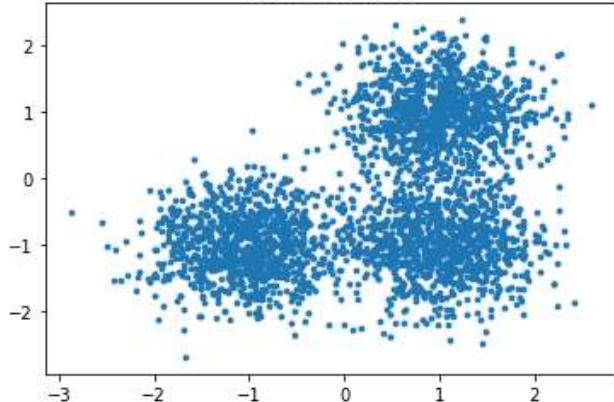
---

- Illustration de la méthode k-means (scikit-learn)
- Evaluation :
  - Fonction objectif : minimiser l'inertie intra-cluster
  - Temps de calcul
  - Taille de jeu de données

# Exemple

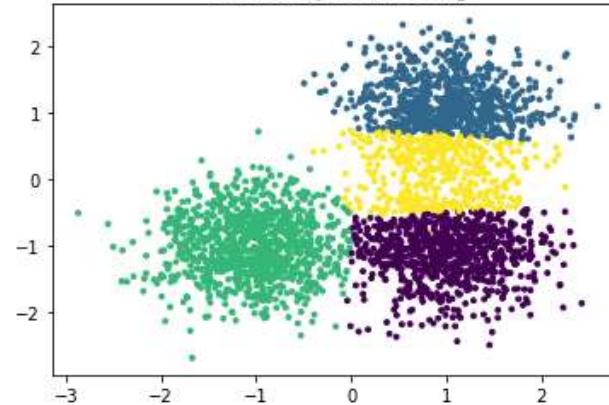
3000 exemples

Données initiales



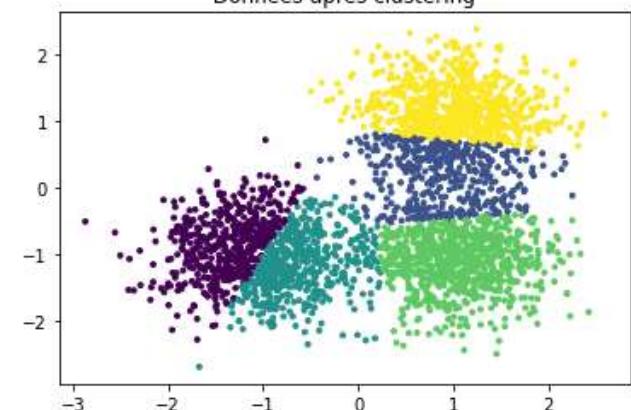
$k = 2$ , Inertie=3317.8132

Données après clustering



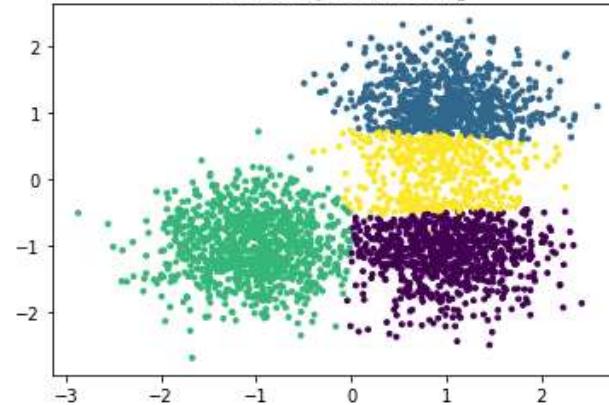
$k = 3$ , Inertie=1395.0552

Données après clustering

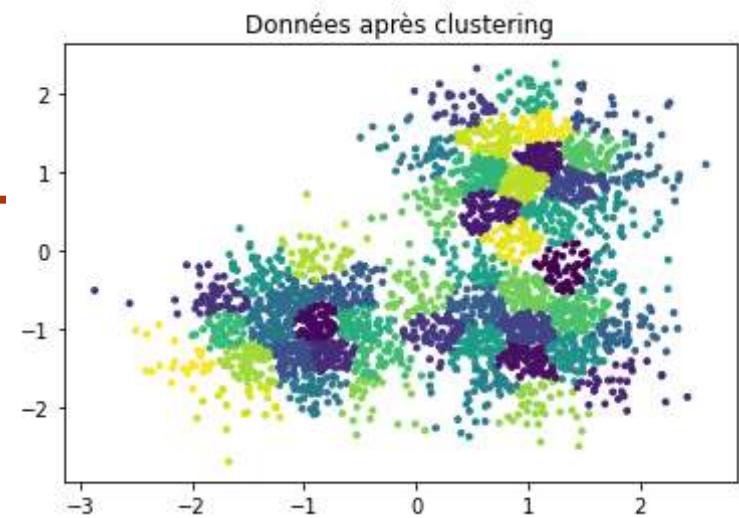
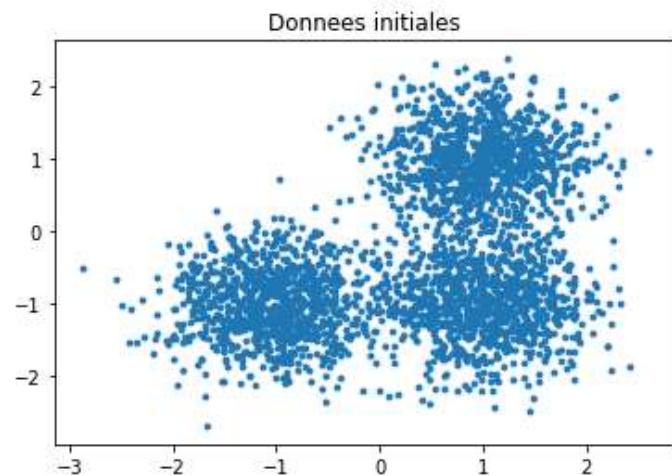


$k = 4$ , Inertie=1217.2132

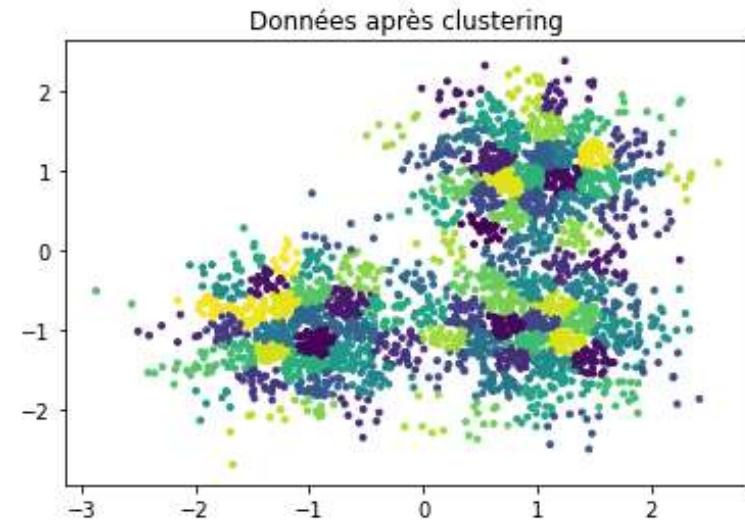
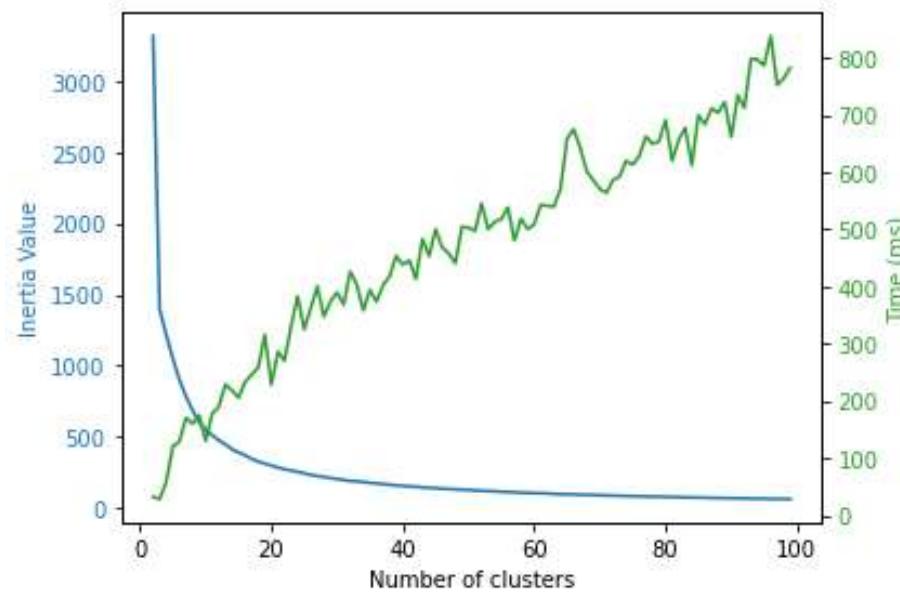
Données après clustering



# Exemple



$k = 50$ , Inertie=124.4352

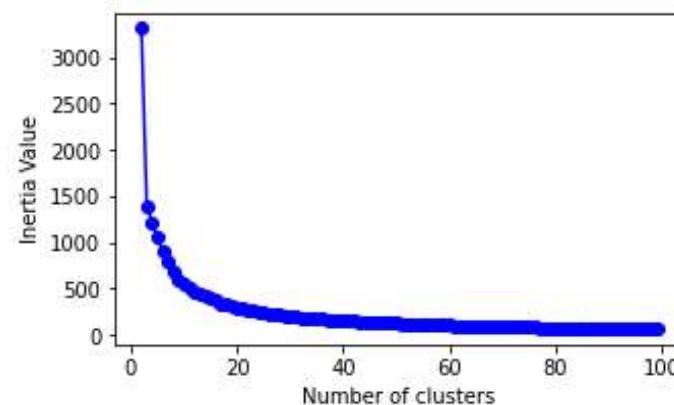
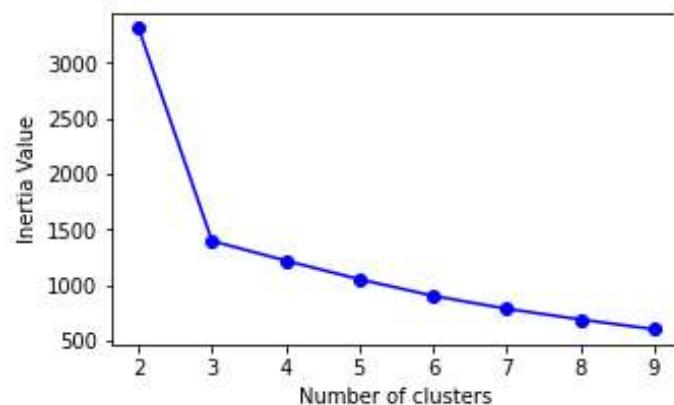


$k = 100$ , Inertie=60.6461

# Exemple

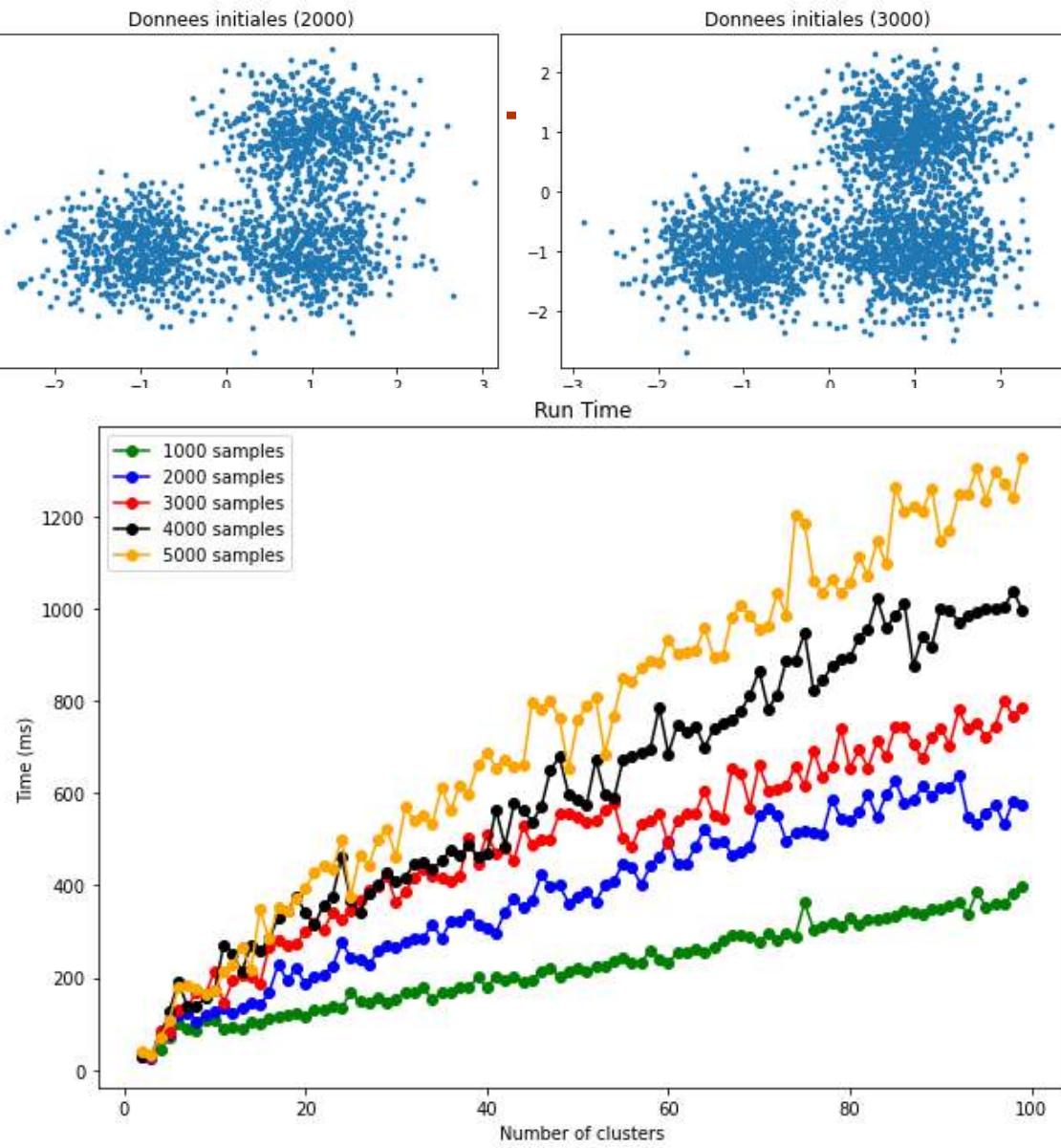
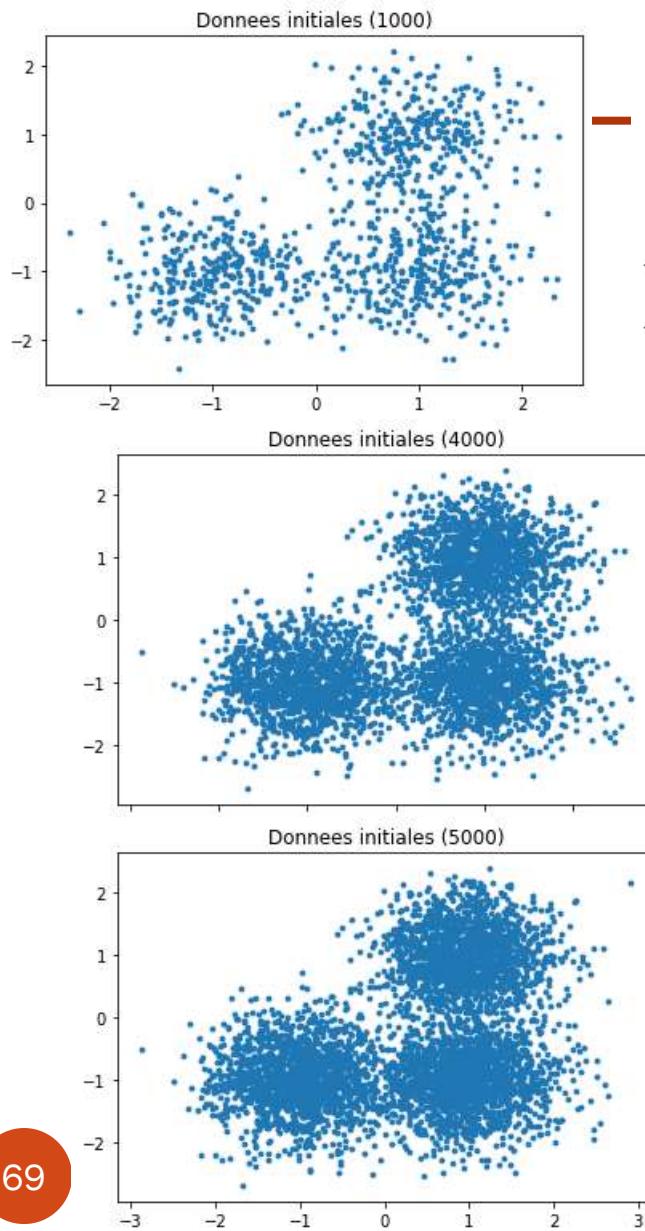
- **Evolution de la fonction objectif en fonction du nombre de clusters**

- Inertie intra-cluster faible : exemples bien regroupés



- La fonction objectif diminue avec le nombre de clusters
  - Plus le nombre de clusters augmente et plus l'inertie intra-cluster diminue
- **Aide pour sélectionner un nombre pertinent de clusters**
  - point d'infexion
  - Méthode du coude (voir plus loin – Evaluation clustering)

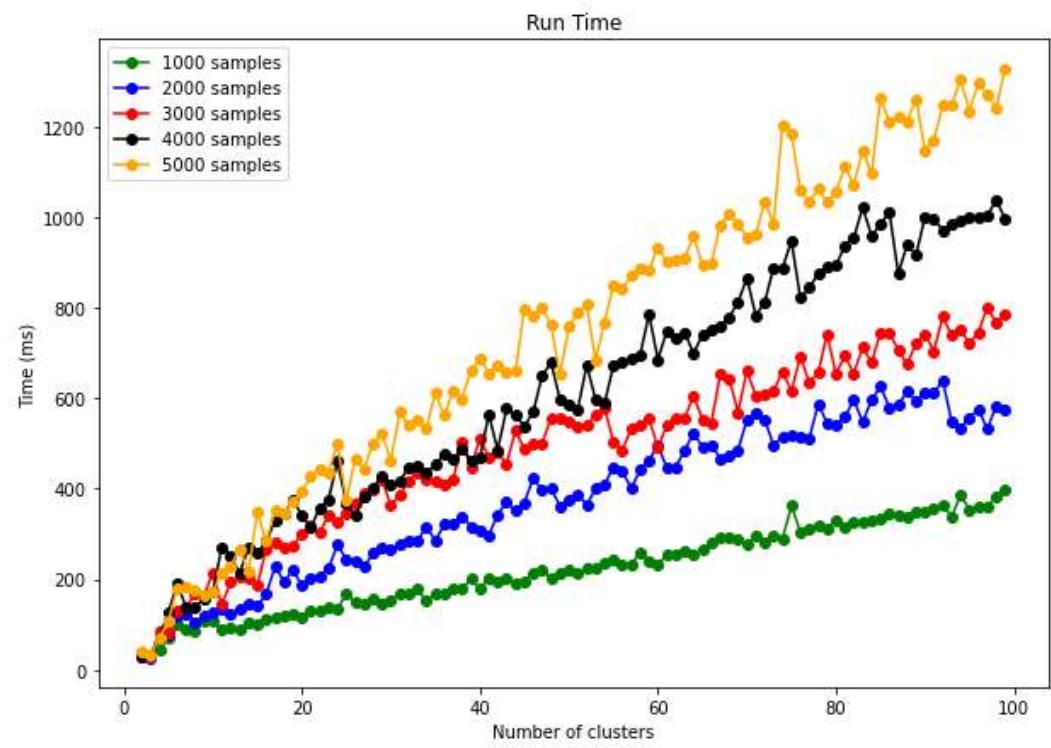
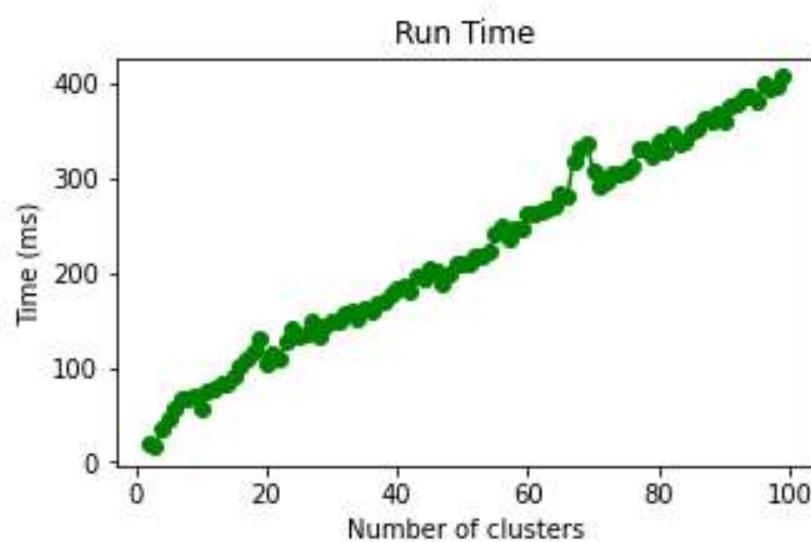
# Exemple



# Exemple

(Fin séance 1 ?)

- **Evolution du temps de calcul**
  - en fonction du nombre de clusters



- et de la taille de jeu de données

# Variantes

---

- ***k Means :***

- les centres ne sont pas forcément des exemples du jeu de données
  - Centres de gravité des exemples de chaque cluster
- En général distance euclidienne au carré

- ***k Medoids :***

- les centres sont les meilleurs représentants de chaque cluster
  - Représentant : exemple le plus « central » de chaque cluster : celui ayant la plus faible dissimilarité par rapport aux autres exemples de chaque cluster
- Sélection initiale des centres
- Permutation des centres avec des exemples du jeu de données pour diminuer la dissimilarité
- Problème NP-difficile :
  - Existence de plusieurs méthodes heuristiques

# K-medoids

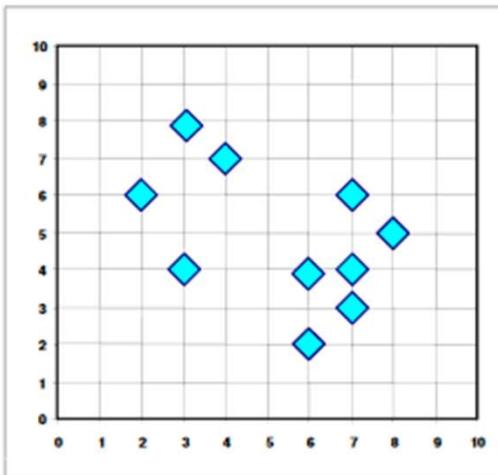
- **Algorithme PAM : permutation around medoïds**

- $k$  : le nombre de clusters est fixé a priori
- Objectif : minimiser la somme des erreurs absolues aux  $k$  medoïds
  - $\sum_{k=1}^K \sum_{i \in C_k} |x_i - o_k|$

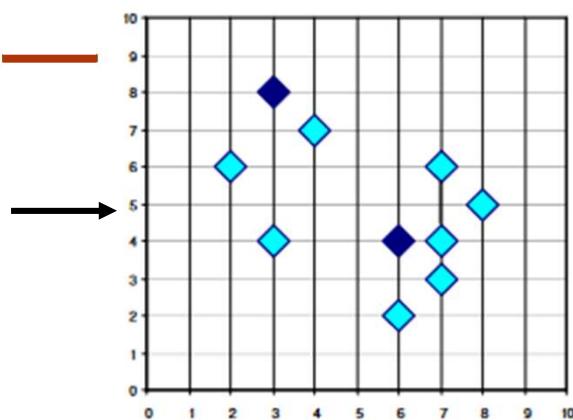
- **Principe de PAM**

- Initialisation :
  - Déterminer  $k$  représentants initiaux : un par un
    - Le premier est le plus central sur tout le jeu de donnée, les suivants permettent de diminuer la fonction objectif
- Répéter
  - Affecter les exemples à leur représentant le plus proche
  - Echange
    - Pour chaque médoïd  $o$  et chaque exemple  $x$  non medoïd :
      - déterminer la variation de coût
      - Si meilleure variation : mémoriser  $o$  et  $x$
    - Réaliser le meilleur échange
  - Jusqu'à plus de variation

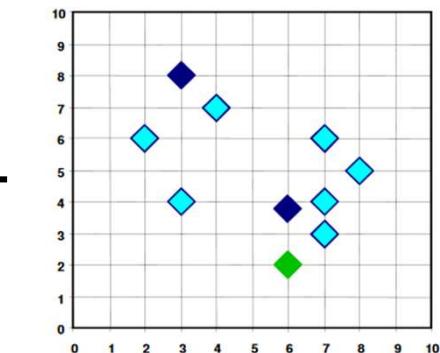
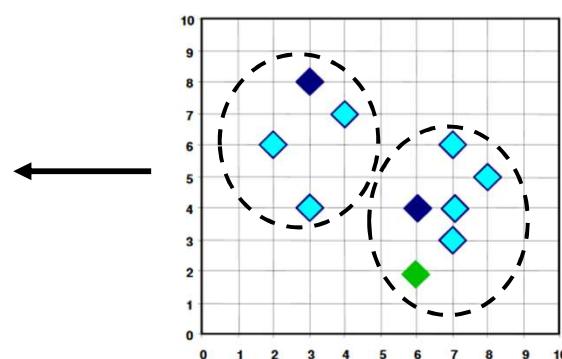
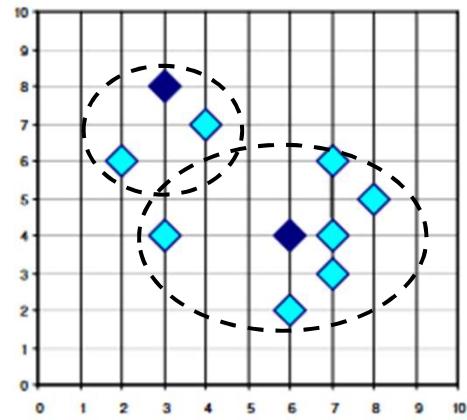
# Exemple



Initialisation



Affection – Cout = 19



# Caractéristiques méthode PAM

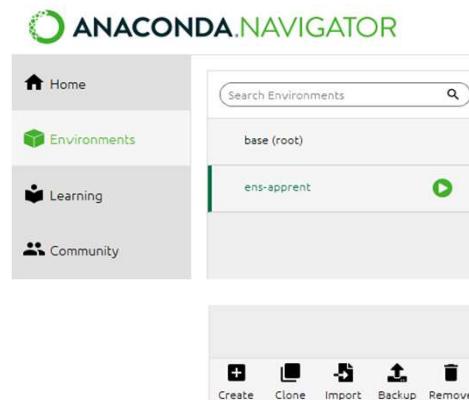
---

- **Stratégie gloutonne**
  - Optimum local
  - Mais temps de calcul important pour des jeux de données de grande taille
  - Existence d'autres variantes (FasterPAM, ...)
- **Métrique de distance**
  - Spécifique pour chaque jeu de données
- **Meilleure interprétabilité du clustering**
  - Les clusters sont basés sur des représentants
- **Moins sensible aux données bruitées et anomalies**
  - Pas de calcul de moyenne

# Pour les TP

- **Anaconda :**

- python – spyder - git
- Conseil :
  - avec Anaconda : créer un environnement virtuel pour les TP de clustering
    - Python 3



- Lancer le terminal de votre environnement virtuel et installer les packages
  - scipy
  - numpy
  - matplotlib
  - scikit-learn

# Pour les TP



- **Scikitlearn**
  - Librairie d'algorithmes d'apprentissage
  - 6 familles de problèmes d'apprentissage
    - Classification / Régression / Clustering
    - Pré-traitement / Reduction de dimension / Sélection de modèles
  - Basé sur (numpy, scipy, matplotlib, pandas, ...)
- Scikit-learn homepage
  - <http://scikit-learn.org>
- Presentations et Tutorials
  - <http://scikit-learn.org/stable/presentations.html>
- **Extension :**
  - Seuls certains algorithmes sont fournis dans scikit-learn
  - Existence de nombreux packages d'algorithmes d'apprentissage respectant le nommage de scikit-learn

# Pour les TP

## • Méthode k-means de scikitlearn

- Sklearn.cluster.Kmeans

<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans>

### • Paramètres principaux

- n\_clusters : 8 par défaut
- init : méthode d'initialisation
  - random ou k-means++ (par défaut)
- n\_init : 10 par défaut. Nombre de graines aléatoires pour l'initialisation. Le meilleur résultat de clustering est retourné
- max\_iter : nombre maximum d'itérations de k-means : 300 par défaut
- tol : par défaut  $10^{-4}$  : condition d'arrêt sur stabilisation des centres
- algorithm : par défaut Lloyd

### • Résultats

- cluster\_centers\_ : coordonnées des centres
- labels\_ : labels de chaque exemple
- inertia\_ : valeur de la fonction objectif
- n\_iter\_ : nombre d'itérations réalisées

### Méthodes :

- fit : pour déterminer le clustering d'un jeu de données
- predict : pour déterminer les clusters de nouveaux exemples

# Pour les TP

---

- **Algorithme k-medoids**
  - n'existe pas dans scikitlearn
- **Package spécifique**
  - pip install kmmedoids
  - <https://python-kmedoids.readthedocs.io/>
- **Distances**
  - Métriques proposées : sklearn.metrics
    - <https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics>
  - Distances implémentées : sklearn.metrics.DistanceMetric
    - <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.DistanceMetric.html?highlight=distance+metrics>

# Plan

---

## **1. Caractérisation du problème de clustering**

1. Données
2. Distances
3. Problème de partition
4. Synthèse

## **2. Quelques Méthodes**

1. Méthodes basées centres de masses
2. **Méthodes hiérarchiques**
3. Méthodes basées voisinage (densité)
4. Méthodes basées graphes

## **3. Bilan Clustering**

1. Evaluation d'un clustering
2. Application

# Méthodes hiérarchiques : Principe général

---

- **Deux types de méthodes hiérarchiques**

- **Clustering ascendant (agglomératif)**

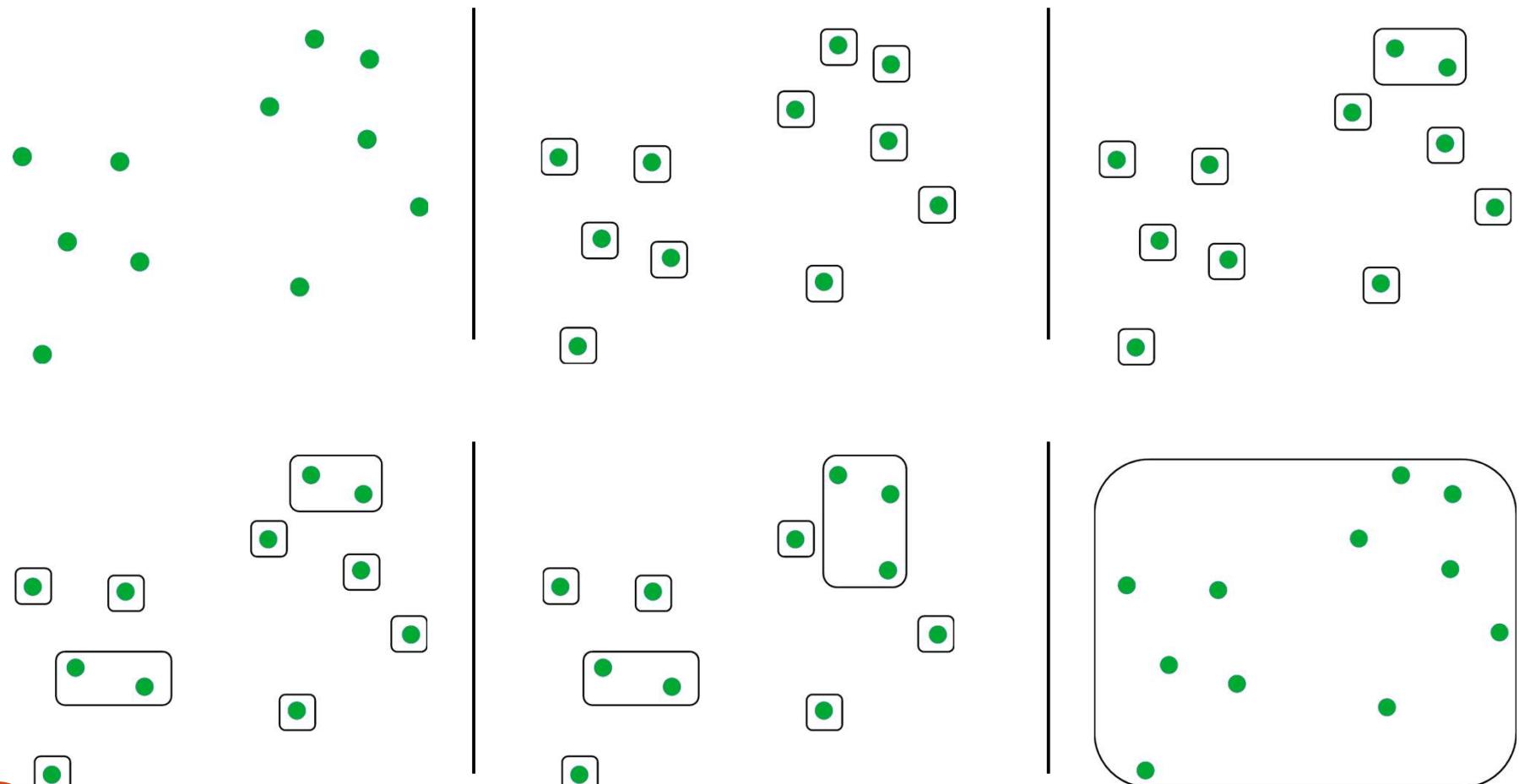
- Initialement chaque exemple (point) est un cluster
    - Fusionner les exemples proches : mesure de similarité (ressemblance)
    - Itérer jusqu'à 1 seul cluster

- **Clustering descendant (divisif)**

- Initialement tous les exemples sont dans le même cluster
    - Le diviser jusqu'à séparer tous les exemples

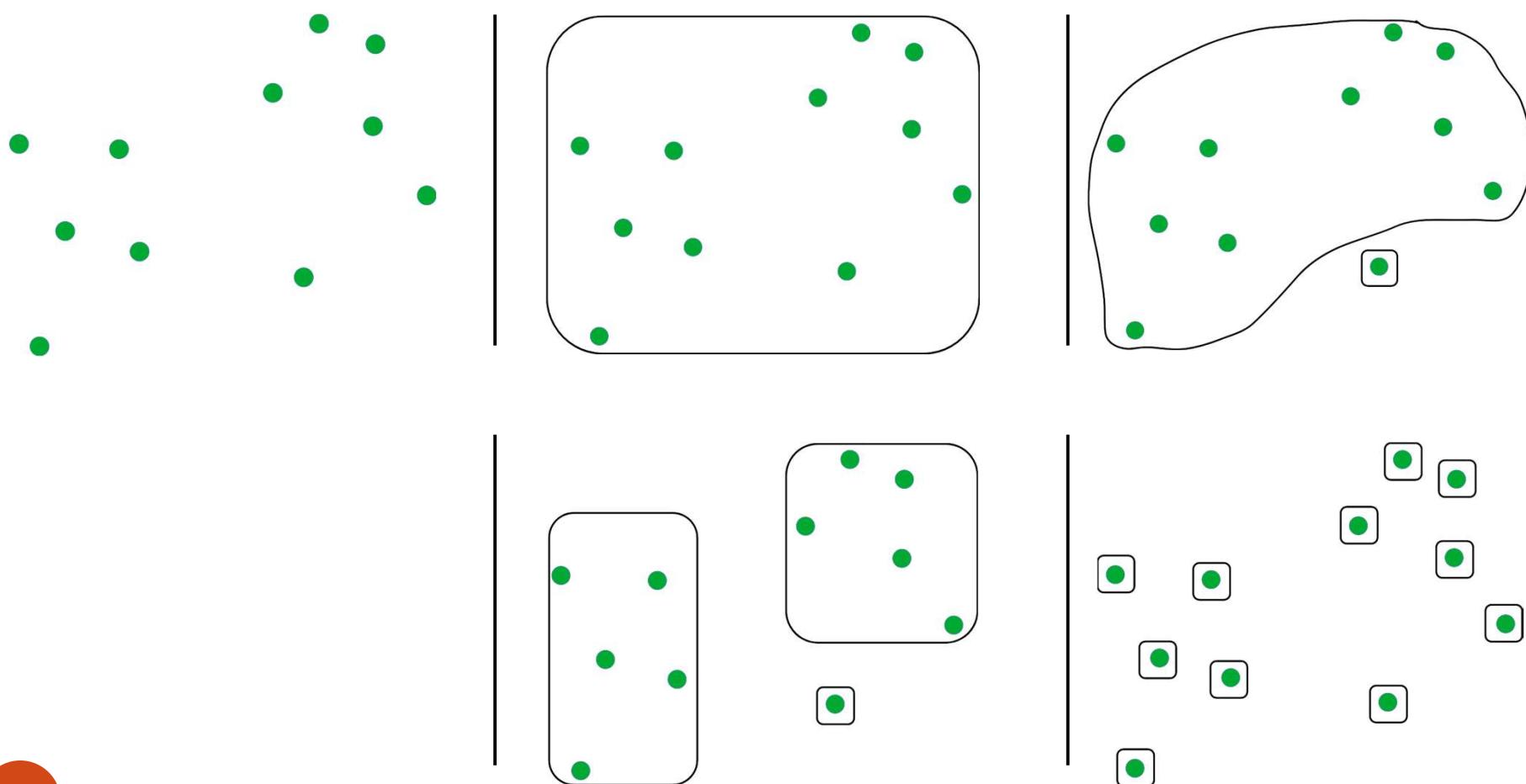
# Clustering ascendant

- **Exemple**



# Clustering descendant

- **Exemple**



# Dendrogramme (1)

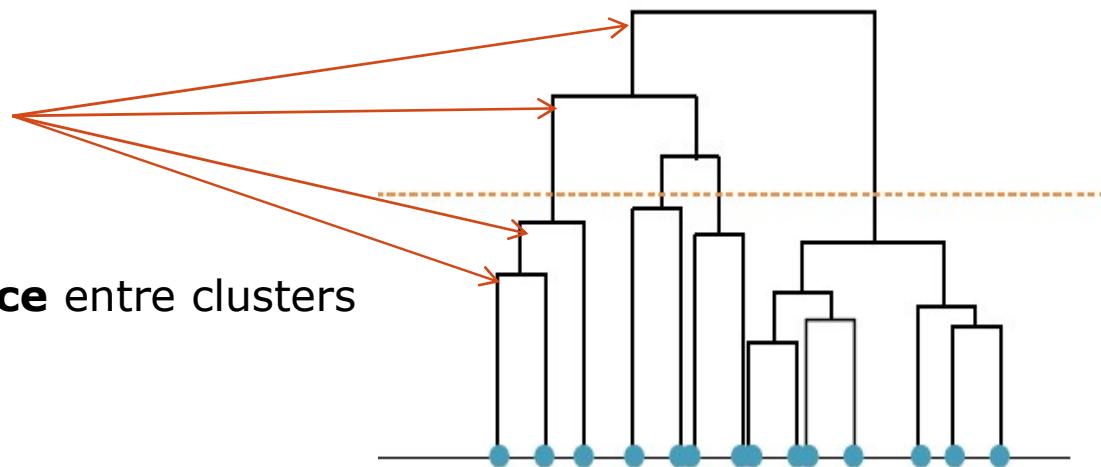
- **Représentation du résultat**

- Dendrogramme = arbre

- Feuilles = exemples
  - Nœuds = cluster

- Hauteur des branches

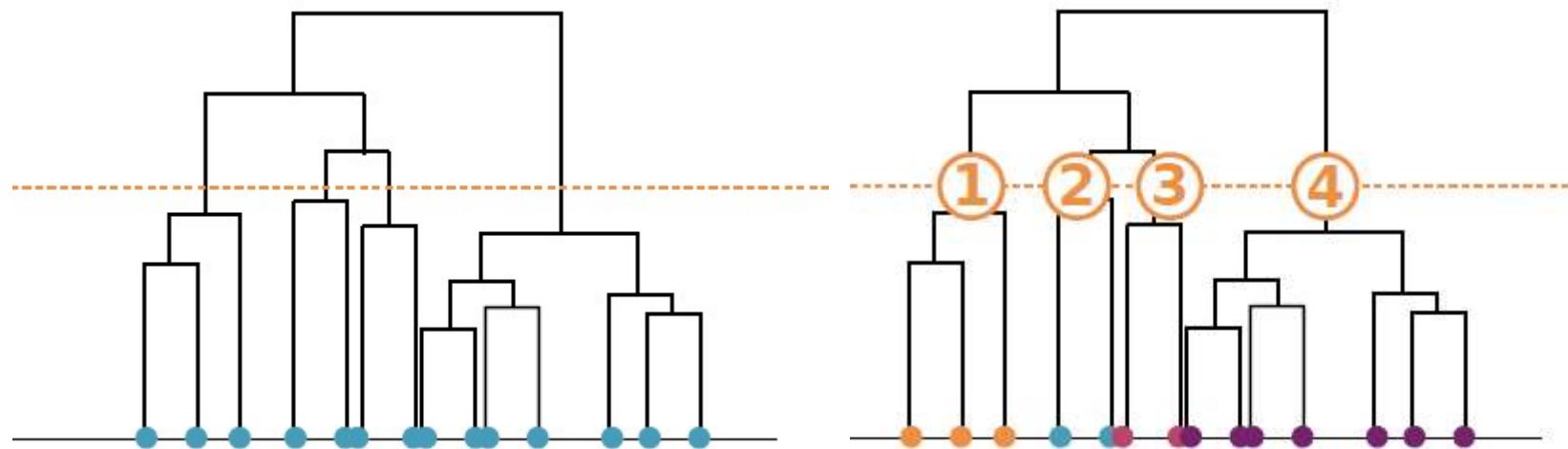
- Proportionnelle **distance** entre clusters



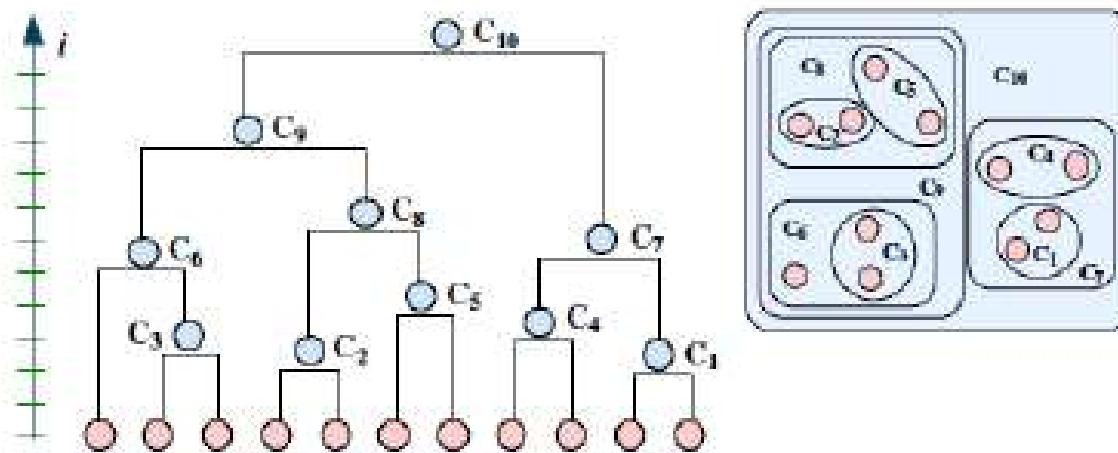
# Dendrogramme (2)

- **Représentation du résultat**

- Couper un dendrogramme
  - Un ensemble de clusters



# Exemple



- Propriété : monotonie
  - Quand on fusionne deux clusters, la similarité avec un autre cluster n'augmente pas
  - Les fusions se font dans l'ordre croissant de similarité
  - Les barres horizontales (fusion/cluster) ne croisent pas les verticales

# Clustering hiérarchique ascendant (CHA)

- **Principe**
  - Chaque exemple ou cluster est fusionné avec le cluster le plus proche

- **Algorithme**
  - Initialisation
    - Chaque exemple est placé dans son propre cluster
    - **Calcul de la matrice  $M$  de « ressemblance » entre exemple**
  - Itérations
    - Sélection dans  $M$  des 2 clusters les plus proches :  $C_i$  et  $C_j$
    - Fusion de  $C_i$  et  $C_j$  pour former un cluster  $C_k$
    - **Mise à jour de  $M$  en calculant « ressemblance » entre le cluster  $C_k$  et les autres clusters**
  - Arrêt : fusion des 2 derniers clusters

**Point clé : avoir une mesure de distance**

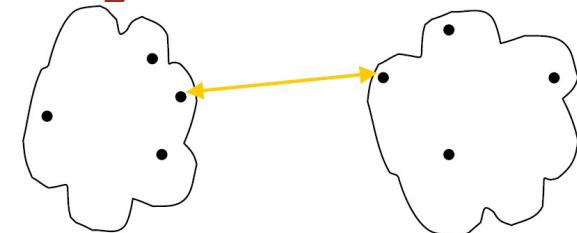
**Point clé : calcul de la similarité entre clusters**

# Similarité entre clusters (1)

- **Exprimer la distance  $\Delta(C_i, C_j)$  entre deux clusters**
  - Sélection des deux clusters les plus proches pour fusionner
- **Plusieurs variantes**

- **Distance minimale entre clusters : single linkage**

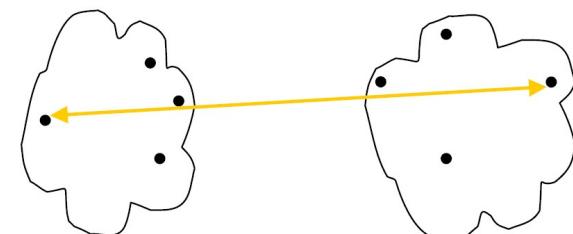
- $\Delta(C_i, C_j) = \min_{x_i \in C_i, x_j \in C_j} \delta(x_i, x_j)$
    - Distance entre les exemples les plus proches



- Fusionner deux clusters si deux de leurs éléments sont proches
    - Effet de chaînage

- **Distance maximale entre clusters : complete linkage**

- $\Delta(C_i, C_j) = \max_{x_i \in C_i, x_j \in C_j} \delta(x_i, x_j)$
    - Distance entre les exemples les plus éloignés

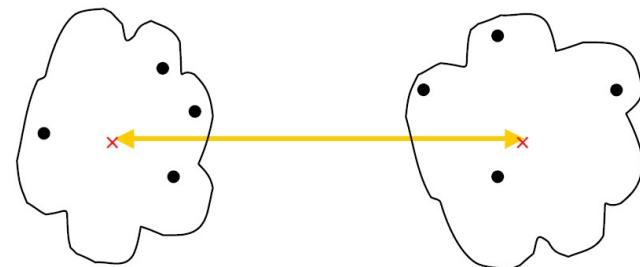


- Fusionner deux cluster si tous leurs éléments sont proches
    - Objectif : minimiser l'accroissement du diamètre de la fusion de 2 clusters

# Similarité entre clusters (2)

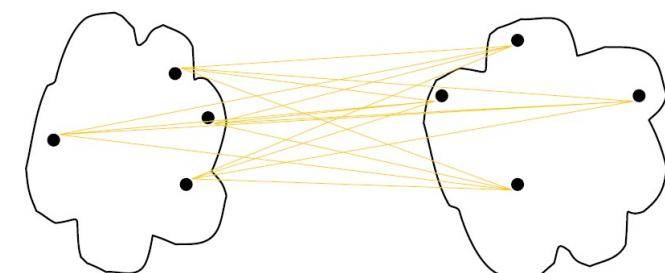
- **Distance moyenne entre clusters**

- Distance entre les centres de chaque cluster : centroïd linkage
- $\Delta(C_i, C_j) = \delta(\mu_i, \mu_j)$ ,  $\mu_i$  et  $\mu_j$  sont les centres



- Moyenne des distances pour toutes les paires d'exemples : average linkage

- $$\Delta(C_i, C_j) = \frac{1}{n_i \times n_j} \sum_{x_i \in C_i} \sum_{x_j \in C_j} \delta(x_i, x_j)$$



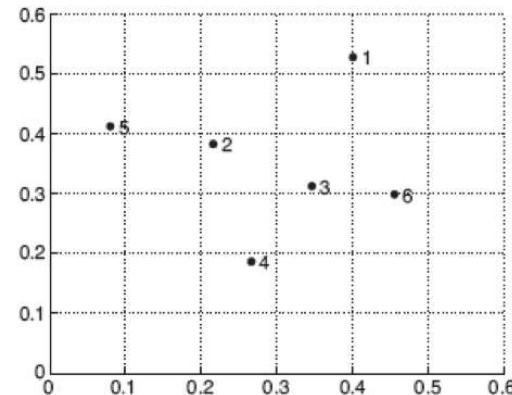
- Fusionner deux clusters si leurs éléments sont proches « en moyenne »

# Similarité entre clusters (3)

- Les distances inter-clusters (minimale / maximale / moyenne) s'intéresse à la séparation des clusters.
  - S'adapte à tout type de distance entre exemple
  - Lien minimal et maximal plus sensible aux bruit et anomalies que lien moyen
- Méthode Ward : considérer l'homogénéité des clusters fusionnés
  - Mesure d'homogénéité : inertie intra-cluster
    - $Inertie(C) = \frac{1}{|C|} \sum_{x \in C} \|x - \mu\|^2$
    - Se base sur distance euclidienne
  - Fusion de deux clusters pour minimiser l'inertie intra-cluster du résultat

# Application (1)

- **Données**



Point	x Coordinate	y Coordinate
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

- **Distance Euclidienne : mesure de « ressemblance »**

- Distance minimale : exemples les plus similaires

	p1	p2	p3	p4	p5	p6
p1	0					
p2	0,23	0				
p3	0,22	0,14	0			
p4	0,37	0,19	0,16	0		
p5	0,34	0,14	0,28	0,28	0	
p6	0,24	0,24	0,10	0,22	0,39	0

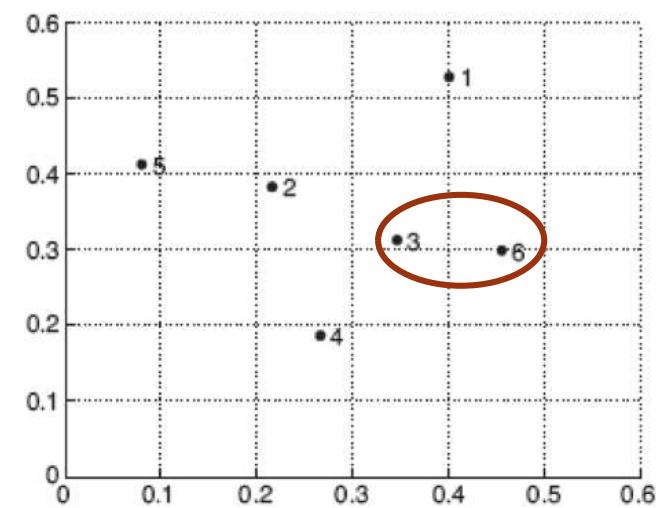
# Application (2)

- **Lien minimal (single linkage)**

	p1	p2	p3	p4	p5	p6
p1	0					
p2	0,23	0				
p3	0,22	0,14	0			
p4	0,37	0,19	0,16	0		
p5	0,34	0,14	0,28	0,28	0	
p6	0,24	0,24	0,10	0,22	0,39	0

	p1	p2	(p3,p6)	p4	p5
p1	0				
p2	0,23	0			
(p3,p6)	0,22	0,14	0		
p4	0,37	0,19	0,16	0	
p5	0,34	0,14	0,28	0,28	0

- Sélection Min  $\rightarrow$  0,10
- Cluster (p3, p6)
- Mise à jour de la matrice de ressemblance
  - o **Distance minimale**



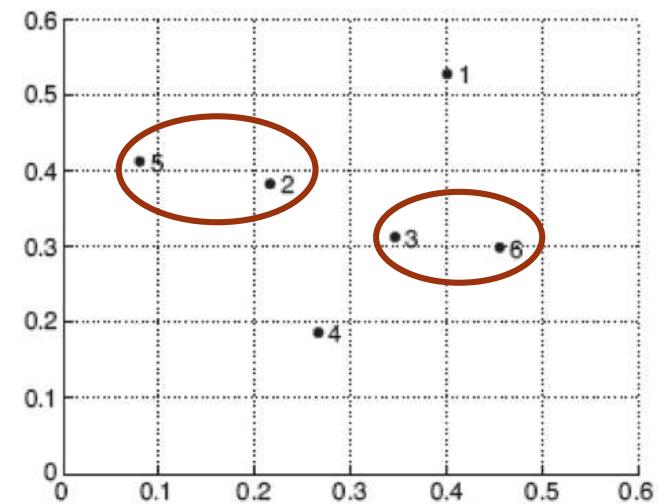
# Application (3)

- Lien minimal (single linkage)

	p1	p2	(p3,p6)	p4	p5
p1	0				
p2	0,23	0			
(p3,p6)	0,22	0,14	0		
p4	0,37	0,19	0,16	0	
p5	0,34	0,14	0,28	0,28	0

	p1	(p2,p5)	(p3,p6)	p4
p1	0			
(p2,p5)	0,23	0		
(p3,p6)	0,22	0,14	0	
p4	0,37	0,19	0,16	0

- Sélection Min  $\rightarrow$  0,14
- Cluster (p2, p5)
- Mise à jour de la matrice de ressemblance
  - Distance minimale



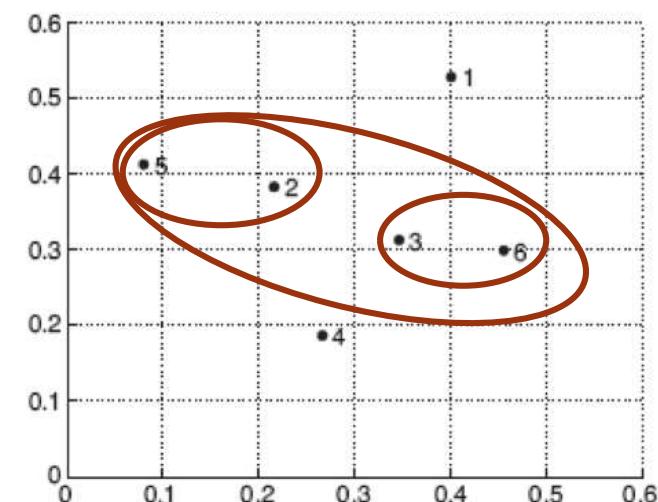
# Application (4)

- Lien minimal (single linkage)

	p1	(p2,p5)	(p3,p6)	p4
p1	0			
(p2,p5)	0,23	0		
(p3,p6)	0,22	0,14	0	
p4	0,37	0,19	0,16	0

	p1	(p2,p5,p3,p6)	p4
p1	0		
(p2,p5,p3,p6)	0,22	0	
p4	0,37	0,16	0

- Sélection Min  $\rightarrow 0,14$
- Cluster (p2, p5, p3, p6)
- Mise à jour de la matrice de ressemblance
  - o Distance minimale



# Application (5)

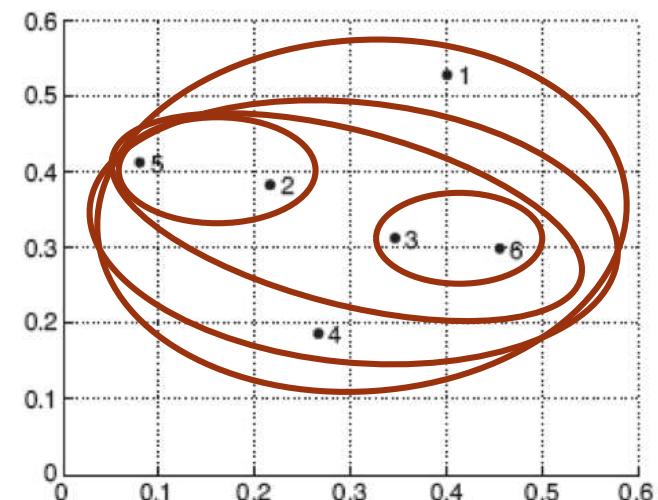
- Lien minimal (single linkage)

	p1	(p2,p5,p3,p6)	p4
p1	0		
(p2,p5,p3,p6)	0,22	0	
p4	0,37	0,16	0

- Sélection Min → 0,16
- Cluster (p2, p5, p3, p6, p4)
- Mise à jour de la matrice de ressemblance
  - Distance minimale

	p1	(p2,p5,p3,p6,p4)
p1	0	
(p2,p5,p3,p6,p4)	0,22	0

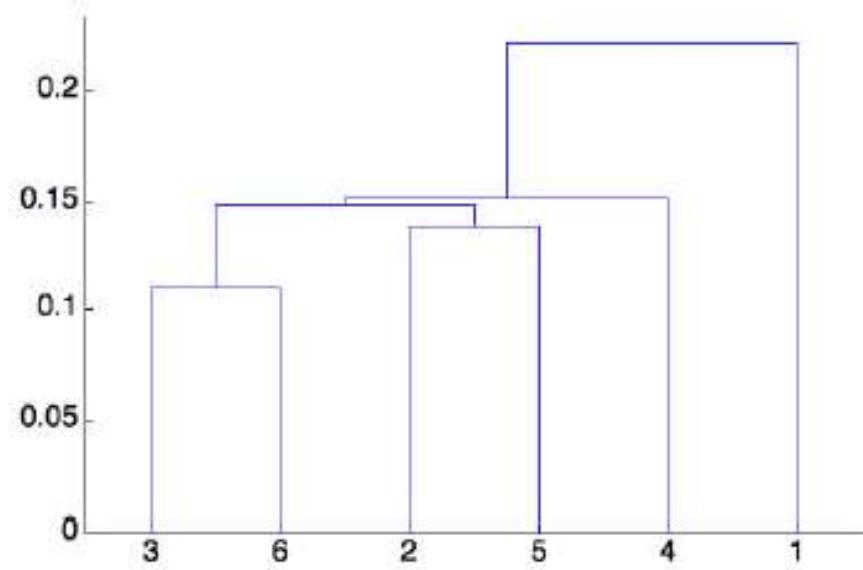
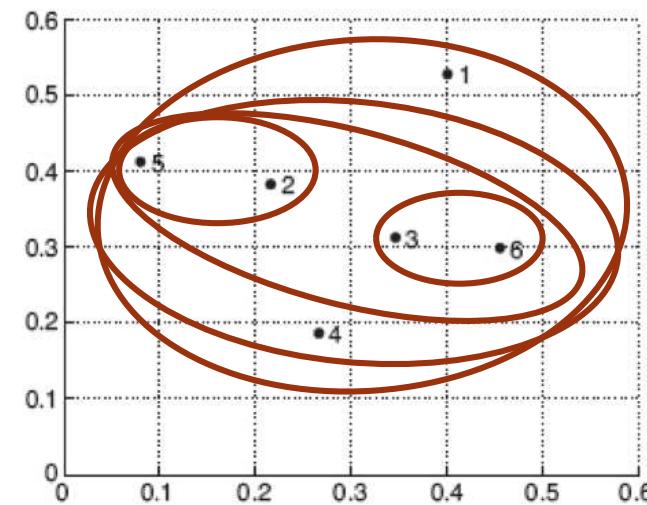
- Sélection Min → 0,22
- Cluster (p2, p5, p3, p6, p4, p1)
- Arrêt



# Résultats différents (1)

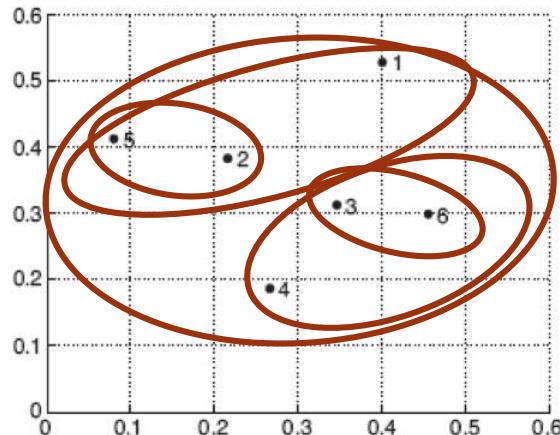
- **Lien minimal (single linkage)**

	p1	p2	p3	p4	p5	p6
p1	0					
p2	0,23	0				
p3	0,22	0,14	0			
p4	0,37	0,19	0,16	0		
p5	0,34	0,14	0,28	0,28	0	
p6	0,24	0,24	0,10	0,22	0,39	0

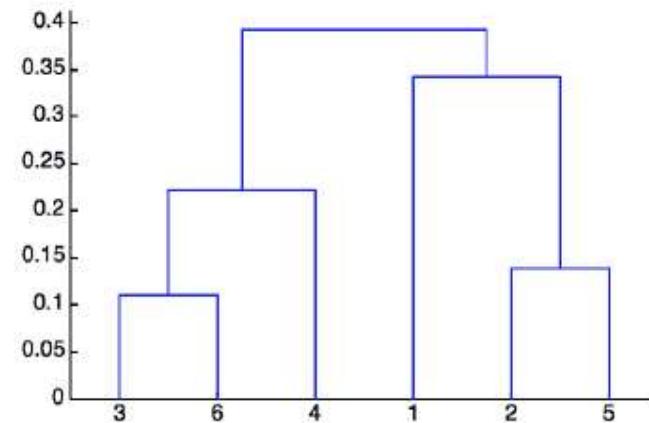


# Résultats différents (2)

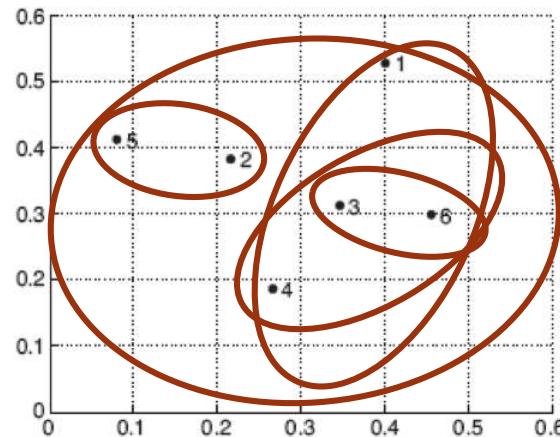
## • Lien maximal (complete linkage)



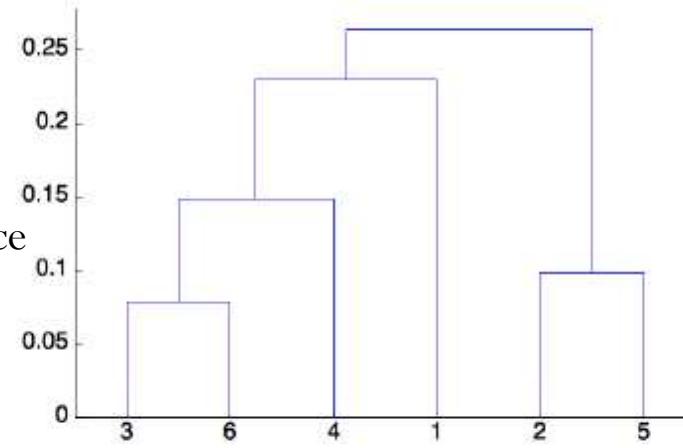
- Sélection Min
- Mise à jour de la matrice
  - Distance maximale



## • Lien moyen (average linkage)



- Sélection Min
- Mise à jour de la matrice
  - Distance moyenne



# Synthèse clustering hiérarchique ascendant

---

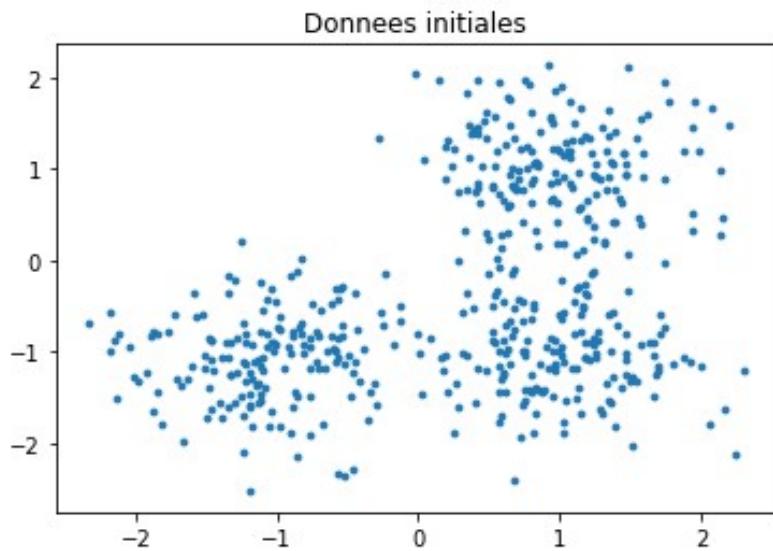
- **Méthode flexible**

- Nombre de clusters non fixé
  - A établir en fonction du dendrogramme
  - Arrêt algorithme : seuil de distance
- Nombre de clusters fixé
  - Coupe le dendrogramme

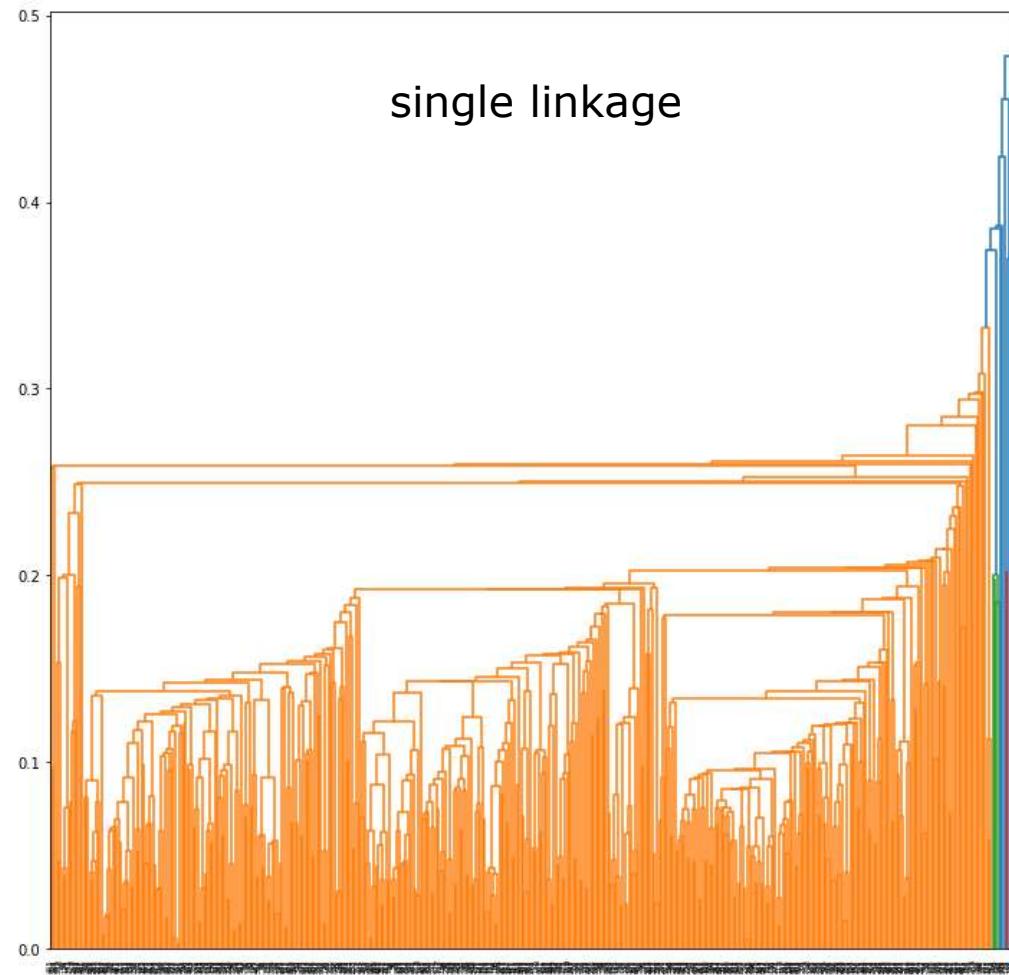
- **Caractéristiques :**

- Complexité : au moins  $n^2$  (calcul de distance)
- Passage à l'échelle difficile
- Pas de remise en cause des clusters fusionnées
- Sensible aux anomalies (outliers)

# Exemple

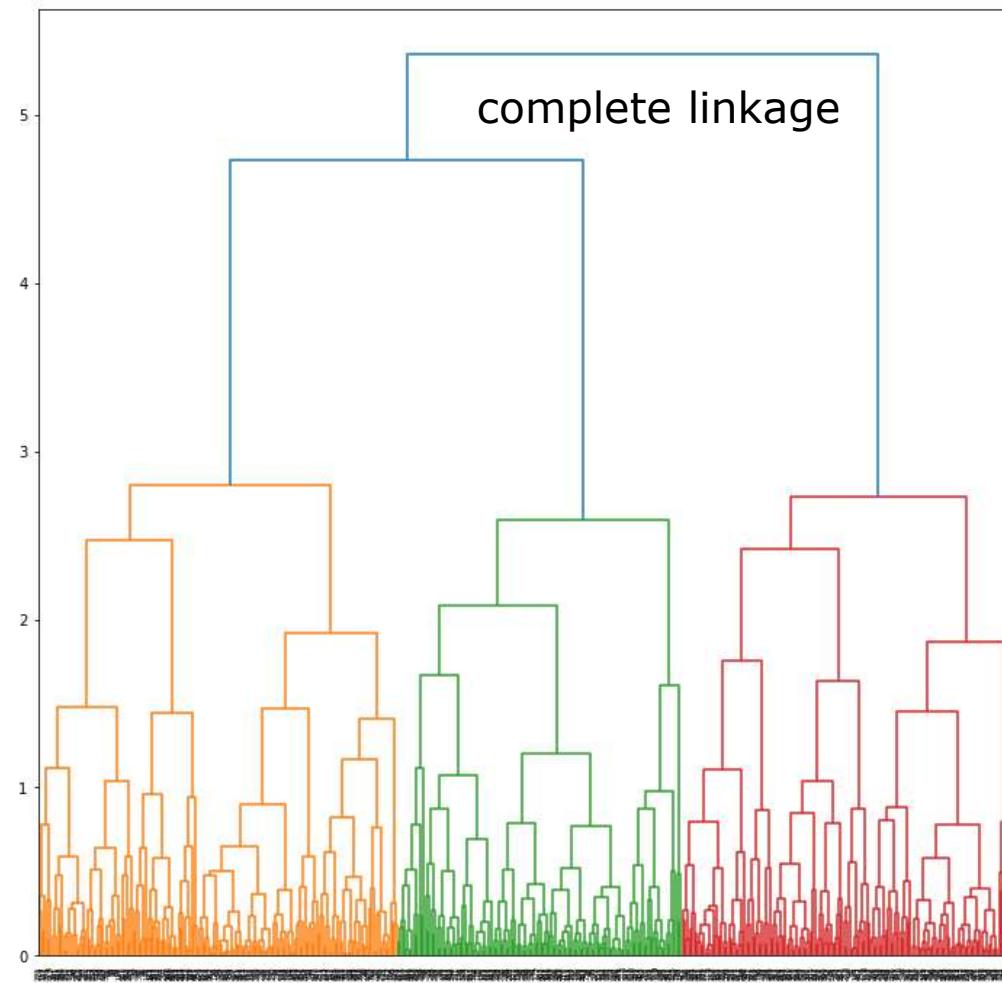


500 exemples

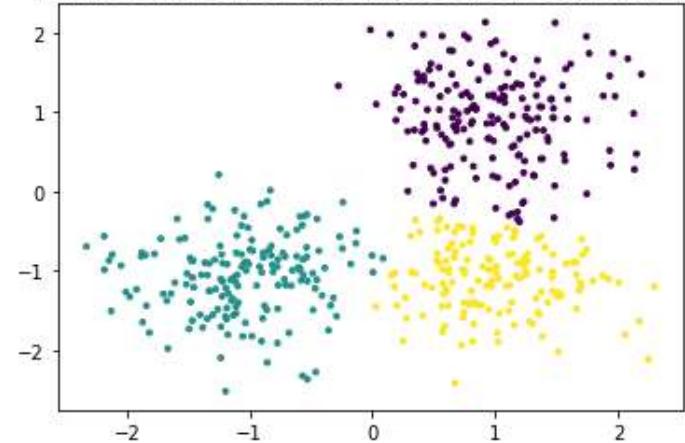


# Exemple

Seuil distance = 3 / 3 clusters

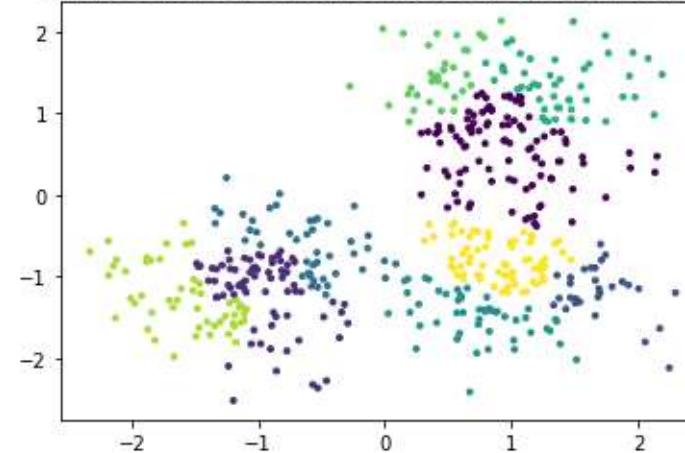


Données après clustering (complete, distance\_threshold=3)

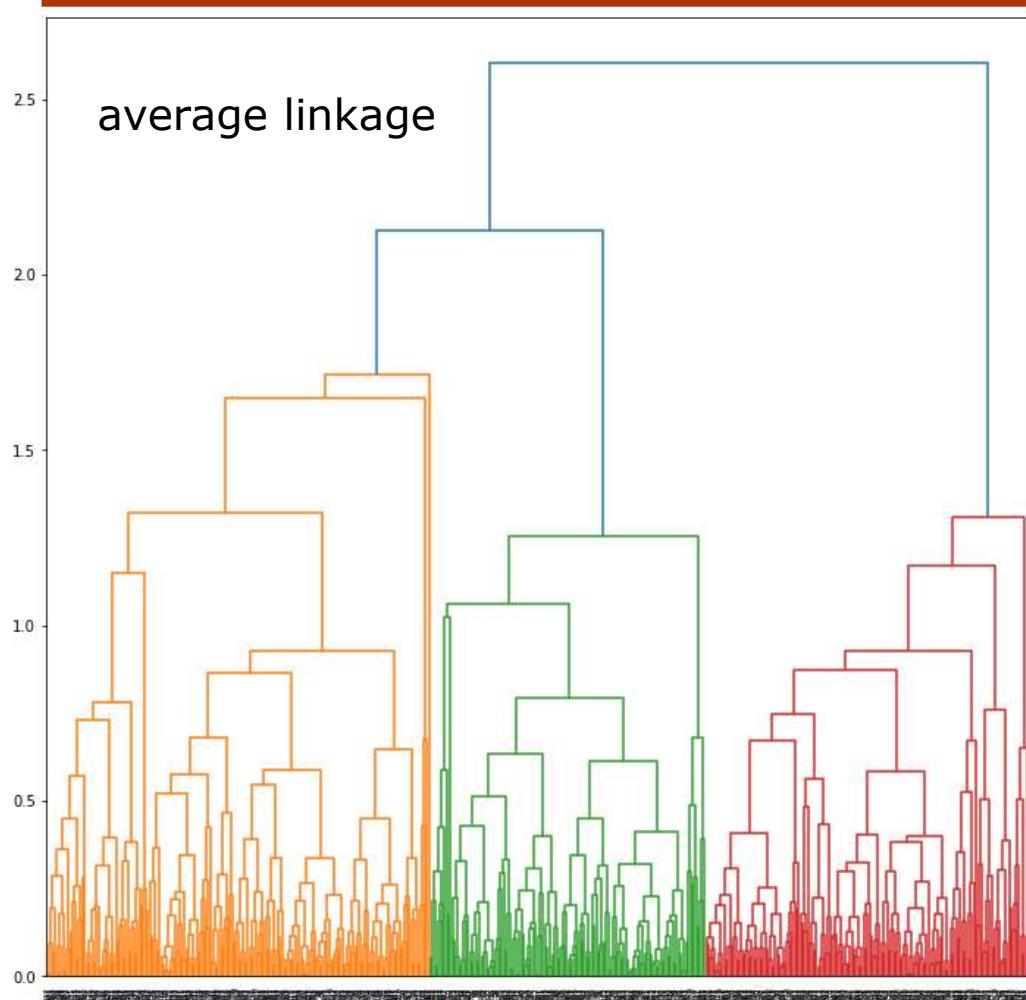


Seuil distance = 2 / 9 clusters

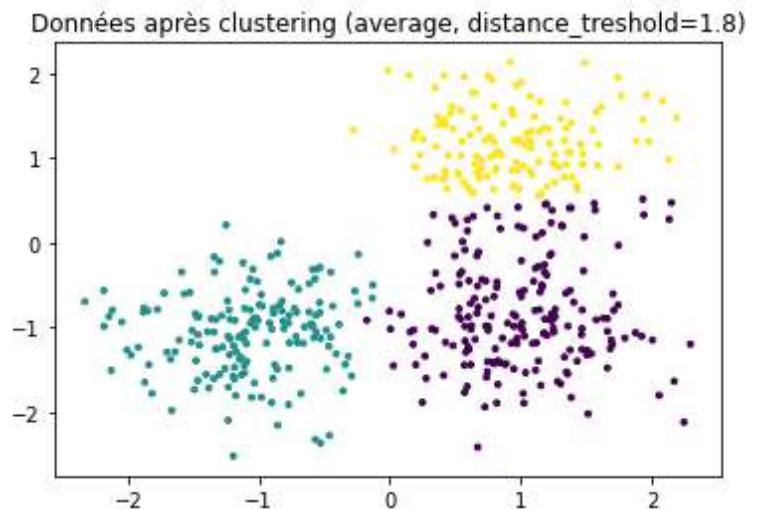
Données après clustering (complete, distance\_threshold=2)



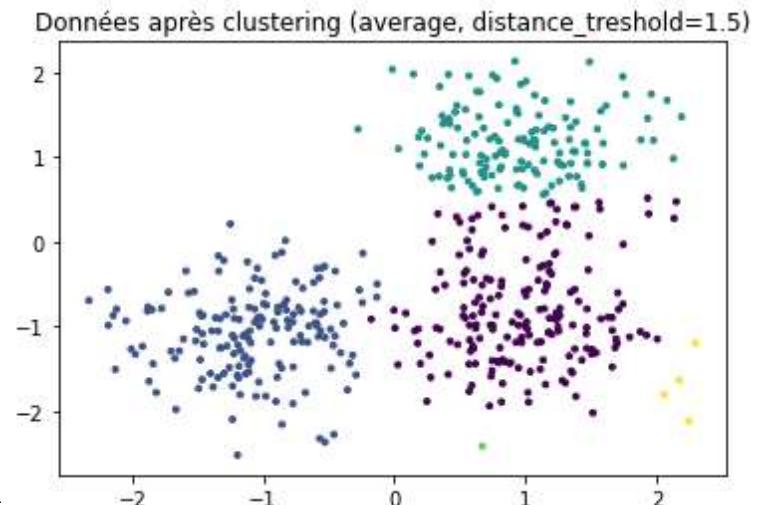
# Exemple



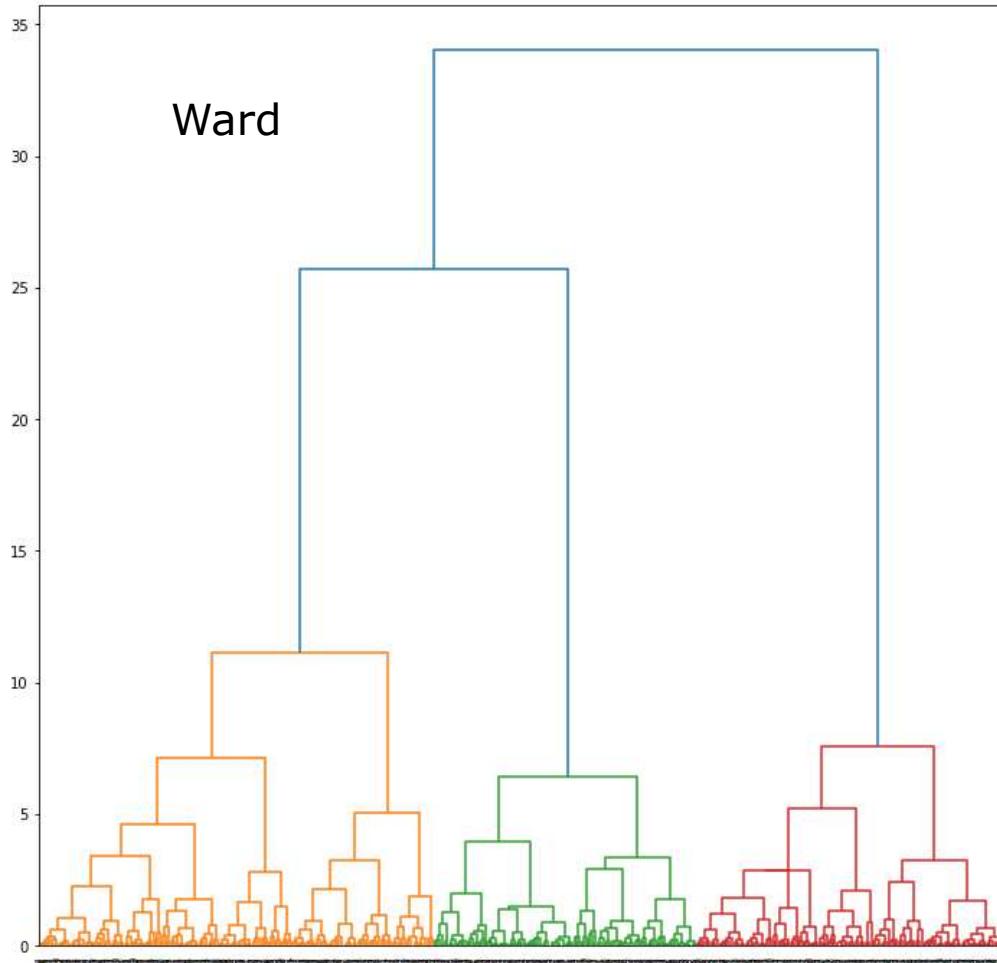
Seuil distance = 1.8 / 3 clusters



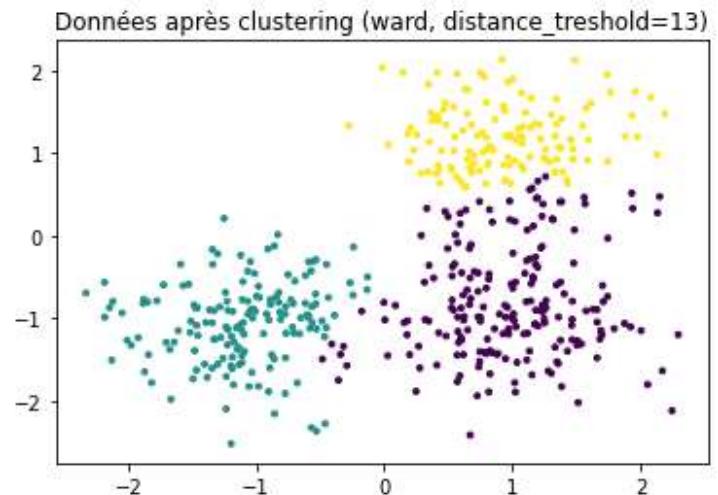
Seuil distance = 1.5 / 5 clusters



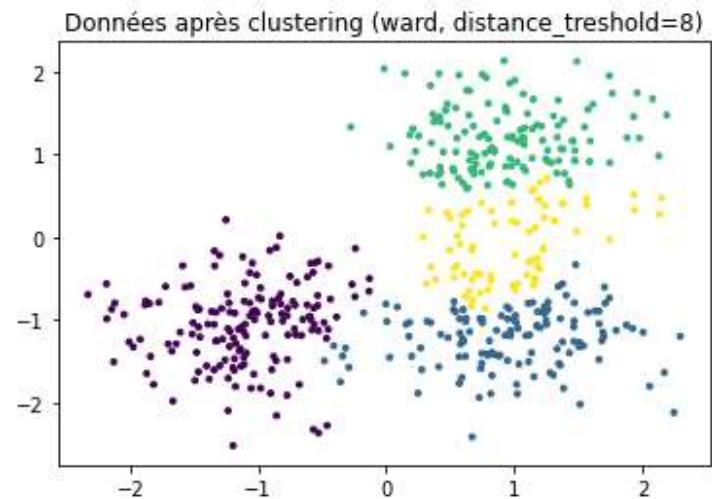
# Exemple



Seuil distance = 13 / 3 clusters

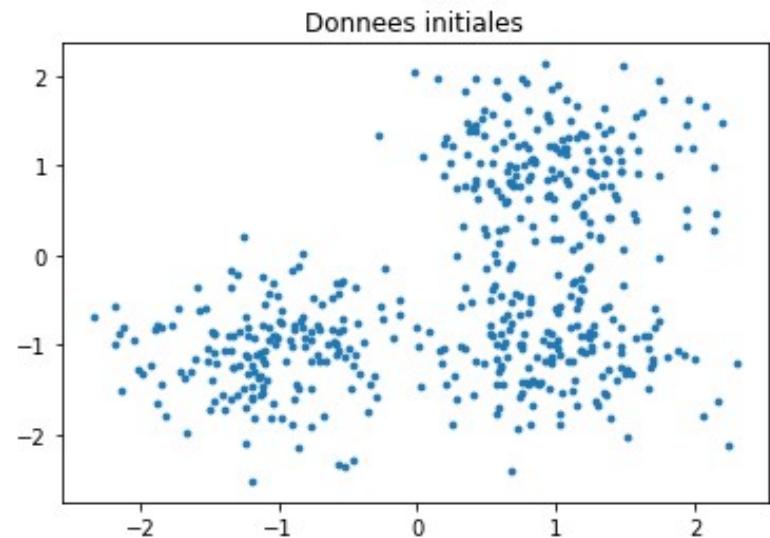


Seuil distance = 8 / 4 clusters

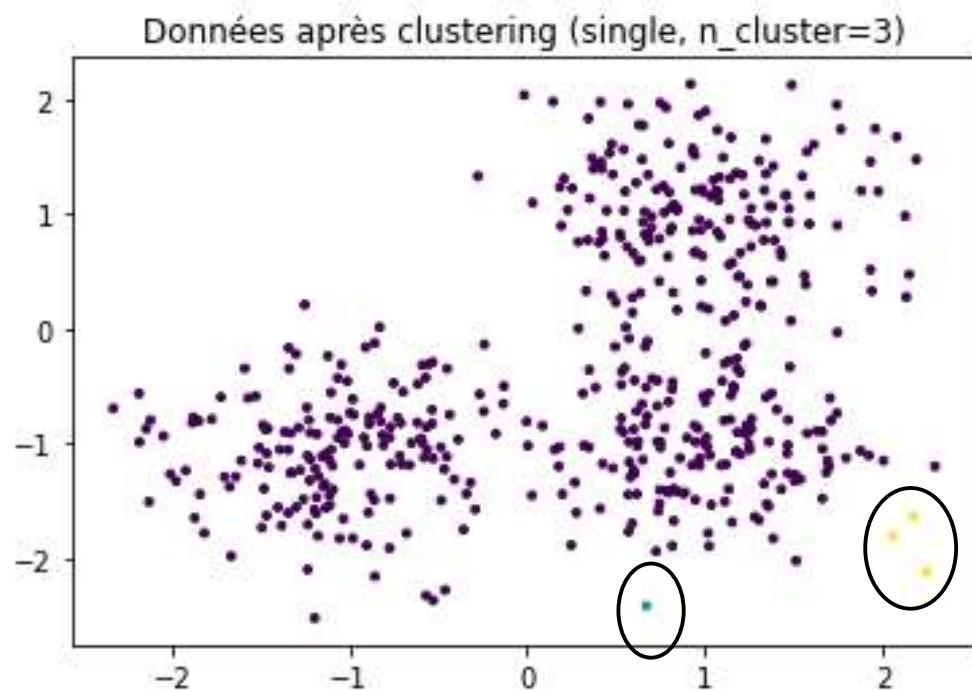


# Exemple

- **Fixer le nombre de clusters**

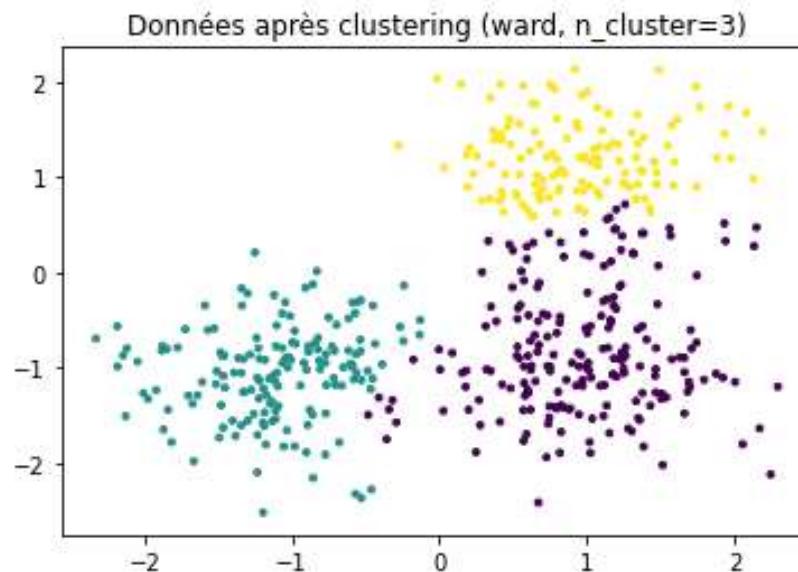
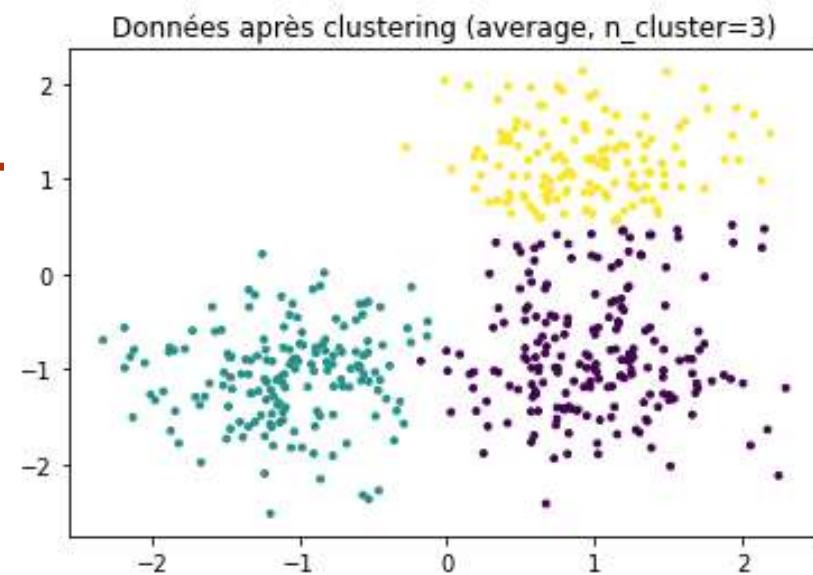
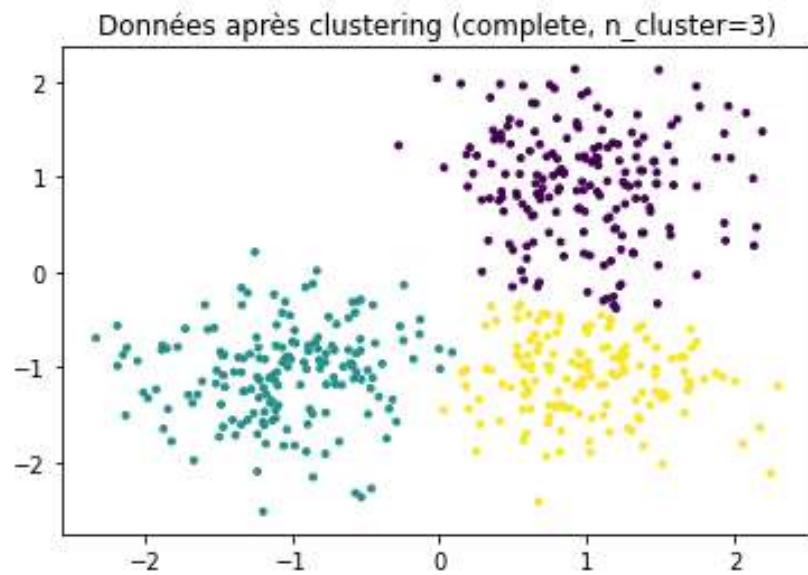


500 exemples;  $K = 3$

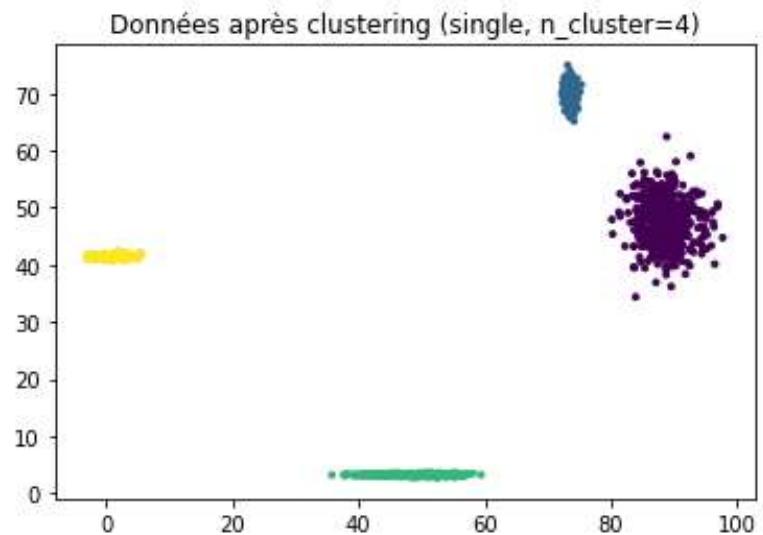
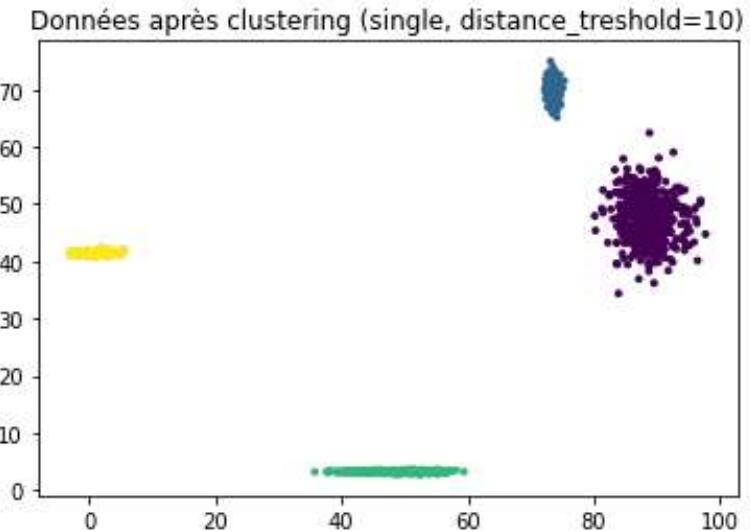
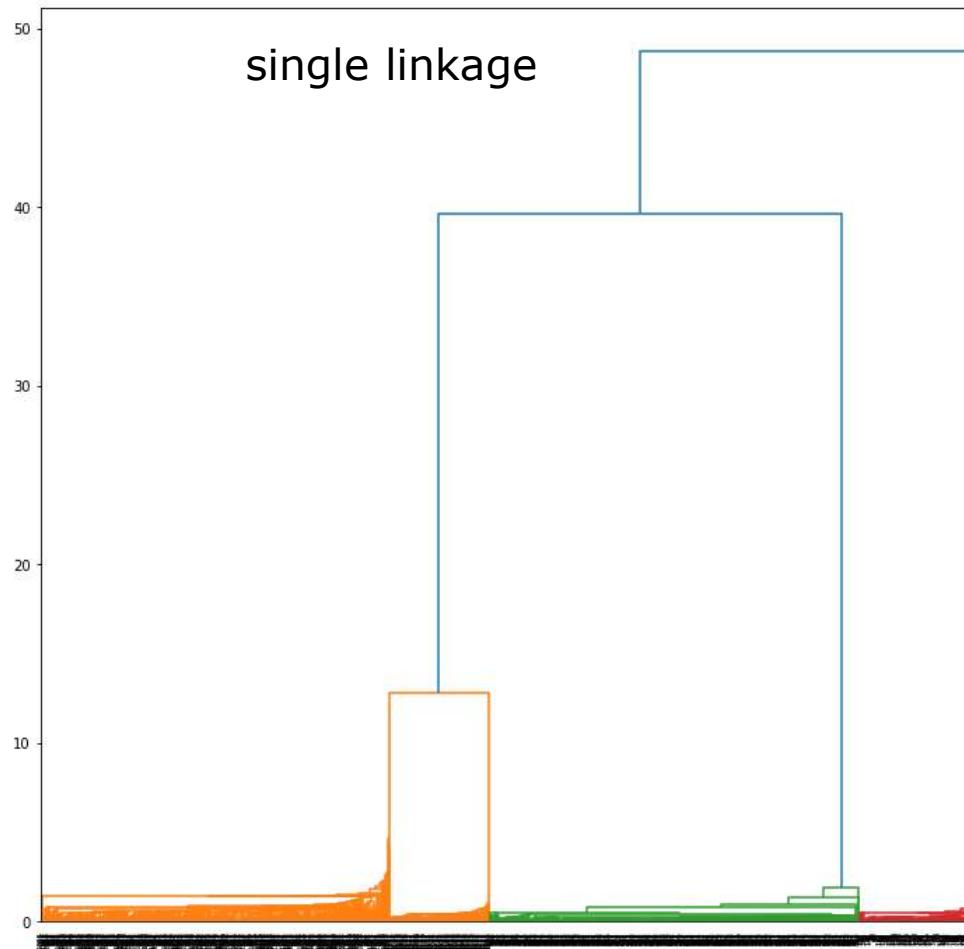
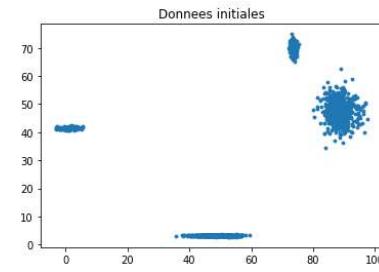


# Exemple

- **Fixer le nombre de clusters**

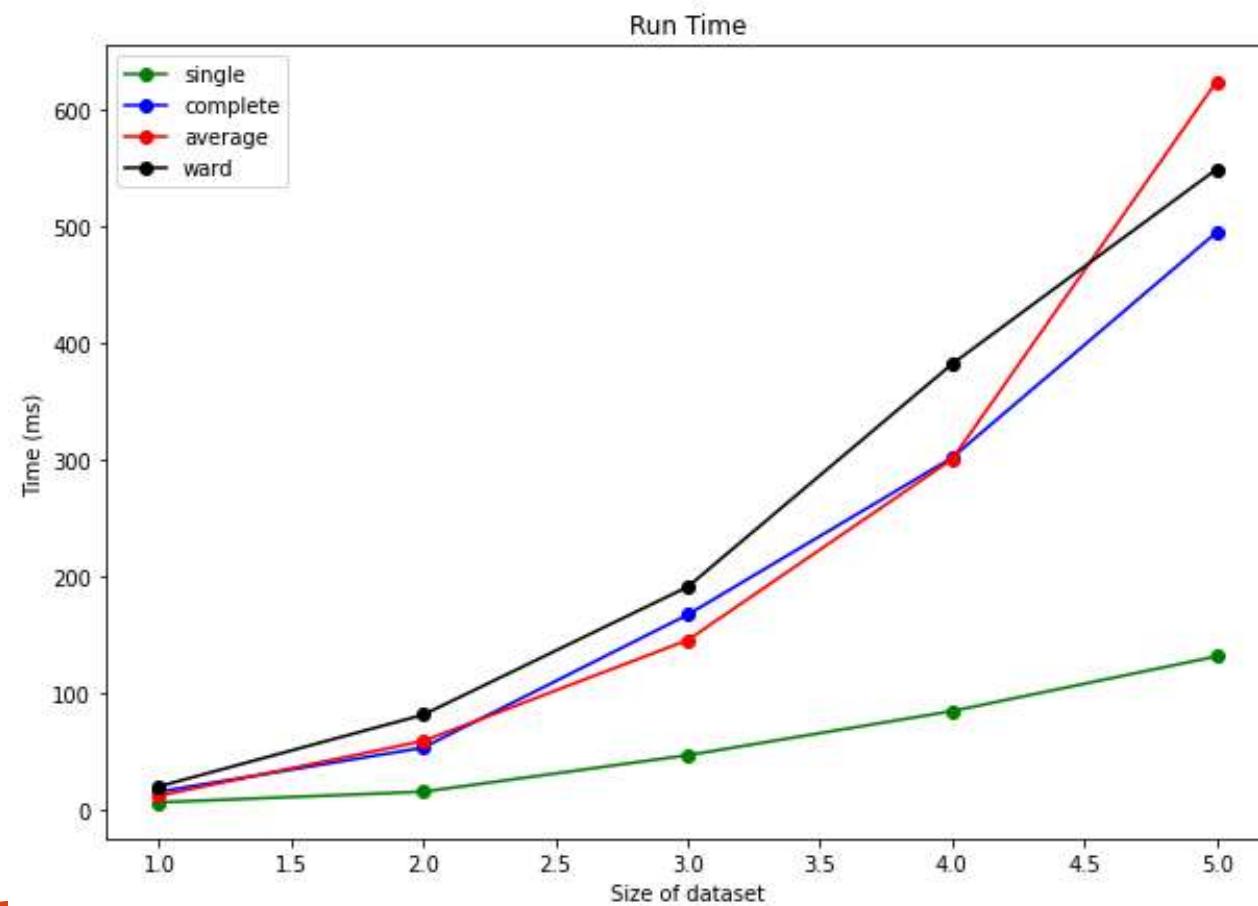


# Autre exemple



# Exemple

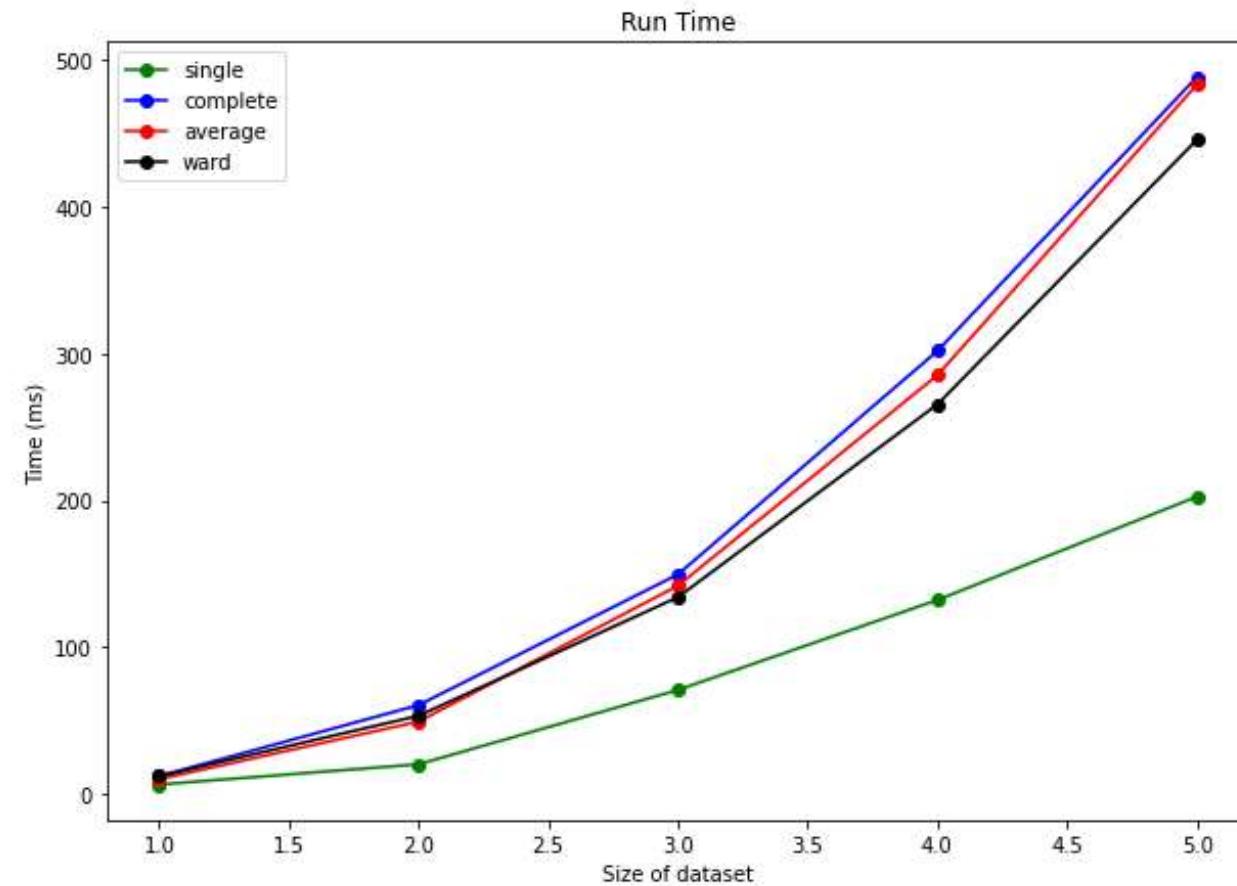
- **Temps de calcul pour déterminer 3 clusters**



Taille des données :  
1000, 2000, 3000,  
4000, 5000

# Exemple

- **Temps de calcul pour calculer tout le dendrogramme**



Taille des données :  
1000, 2000, 3000,  
4000, 5000

# Pour les TP

---

- **Méthode AgglomerativeClustering de scikitlearn**

- Sklearn.cluster. AgglomerativeClustering

<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html#sklearn.cluster.AgglomerativeClustering>

- **Paramètres principaux**

- n\_clusters : defaut = 2 / None
- distance\_threshold : default = None
- affinity : distance entre paire de points (euclidian par défaut)
- linkage : single, complete, average, ward : distance entre deux clusters

- **Résultats**

- cluster\_centers\_ : coordonnées des centres
- labels\_ : labels de chaque exemple
- n\_leaves\_
- n\_connected\_components\_

**Méthodes :**

- fit : pour déterminer le clustering d'un jeu de données
- predict : pour déterminer les clusters de nouveaux exemples

# Clustering hiérarchique descendant

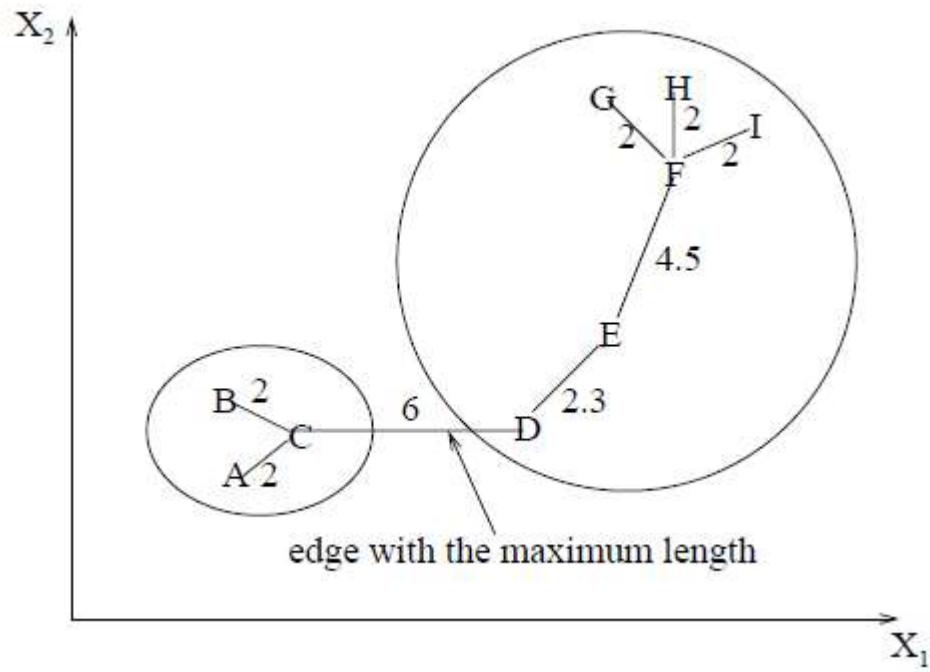
---

- **Principe**

- Clustering descendant (divisif)
    - Initialement tous les exemples sont dans le même cluster
    - Le diviser jusqu'à séparer tous les exemples
      - Sélectionner les exemples les moins similaires
  - Peu utilisé ?
- 
- Approches heuristiques
    - Ascendante : regrouper les exemples les plus proches
    - Descendante : séparer les exemples les plus éloignés
    - basées sur calculs de distance

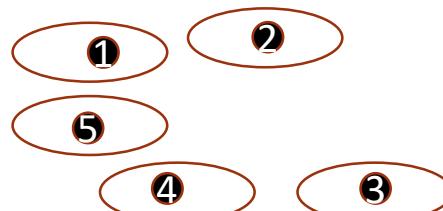
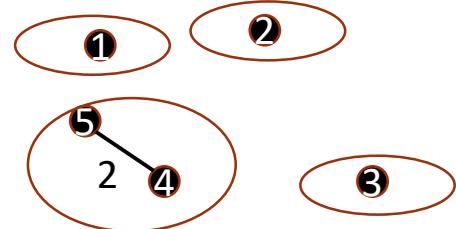
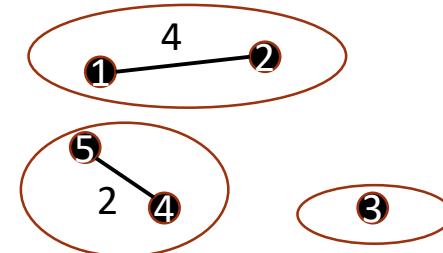
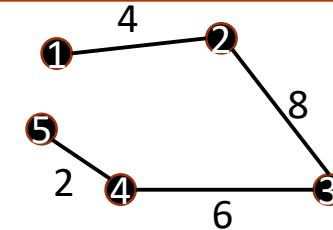
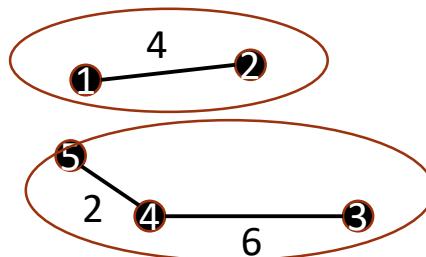
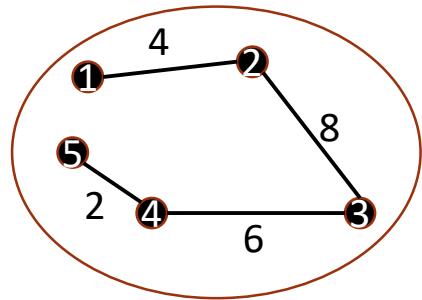
# Méthode basée sur un calcul d'arbre couvrant

- **A partir des distances entre chaque paire de points**
  - Calcul de l'arbre couvrant minimal
    - Minimal Spanning Tree (MST)



# Exemple

- **Exploitation de l'arbre couvrant**
  - Clustering descendant



# Résumé clustering hiérarchique

---

- **Basé sur une mesure de proximité entre exemples et entre clusters**
- **Construction incrémentale des clusters**
  - Pas de remise en cause des clusters fusionnés
  - Il existe des variantes permettant des modifications des fusions
- **Ne nécessite pas de fixer a priori le nombre de clusters**
  - Déterminer ce nombre à partir du dendrogramme
- **Temps de calcul assez important**

# Plan

---

## 1. Caractérisation du problème de clustering

1. Données
2. Distances
3. Problème de partition
4. Synthèse

## 2. Quelques Méthodes

1. Méthodes basées centres de masses
2. Méthodes hiérarchiques
3. Méthodes basées voisinage (densité)
4. Méthodes basées graphes

## 3. Bilan Clustering

1. Evaluation d'un clustering
2. Application

- 
- (en pause – Evaluation clustering)

# Plan

---

## **1. Caractérisation du problème de clustering**

1. Données
2. Distances
3. Problème de partition
4. Synthèse

## **2. Quelques Méthodes**

1. Méthodes basées centres de masses
2. Méthodes hiérarchiques
3. Méthodes basées voisinage (densité)
4. Méthodes basées graphes

## **3. Bilan Clustering**

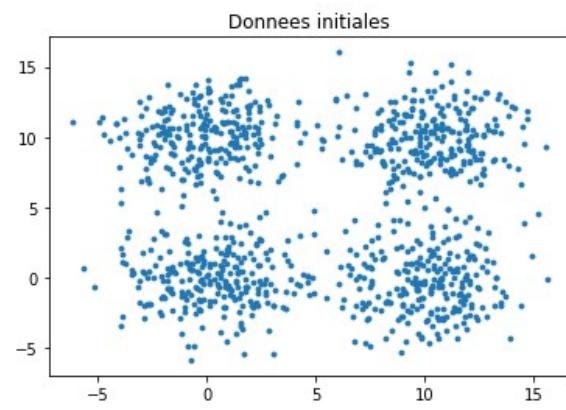
1. Evaluation d'un clustering
2. Application

# Combien de clusters ?

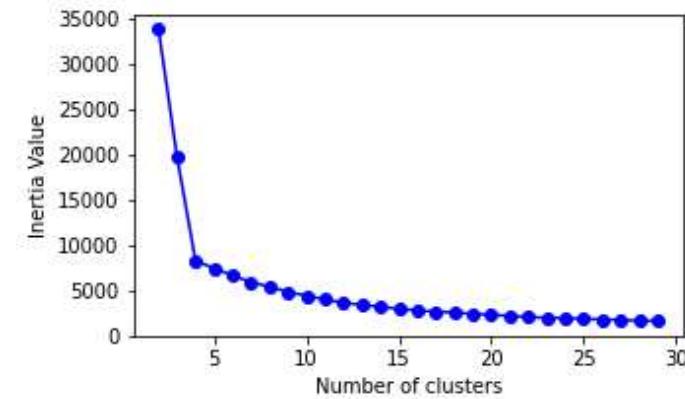
- **Apprentissage non supervisé : il n'y a pas de labels dans le jeu de données**
  - Différence avec l'apprentissage supervisé
  - Métriques « d'évaluation interne »
- **Problème difficile**
  - Fixé : segmenter en  $K$  (contraintes du problème)
    - Itérer sur différentes valeurs de  $K$ 
      - Evaluer la qualité de chaque clustering
      - Exemple avec kmeans : valeur de la fonction objectif (inertie intra-cluster)
    - Non fixé :
      - Itérer sur différentes valeurs de  $K$  et sélectionner la solution ayant la meilleure évaluation
      - Imposer des contraintes sur le volume ou la densité des clusters

# Illustration pour kmeans

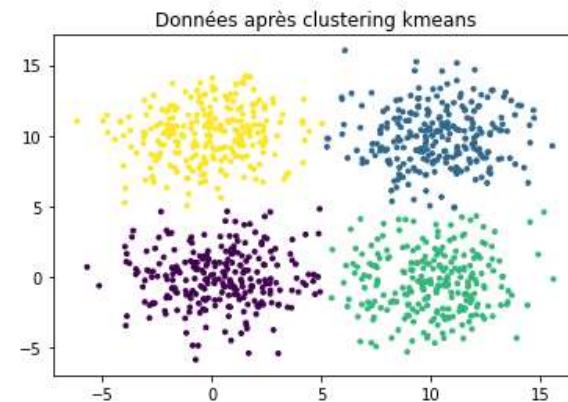
- **Minimisation de l'inertie intra cluster**



Valeur de l'inertie pour  $k = 2$  à 10



- Nombre de clusters minimisant la fonction objectif :  $k = 30$
- Considérer le point d'infexion de la courbe :  $k = 4$ 
  - Méthode du « coude » (Elbow Method)



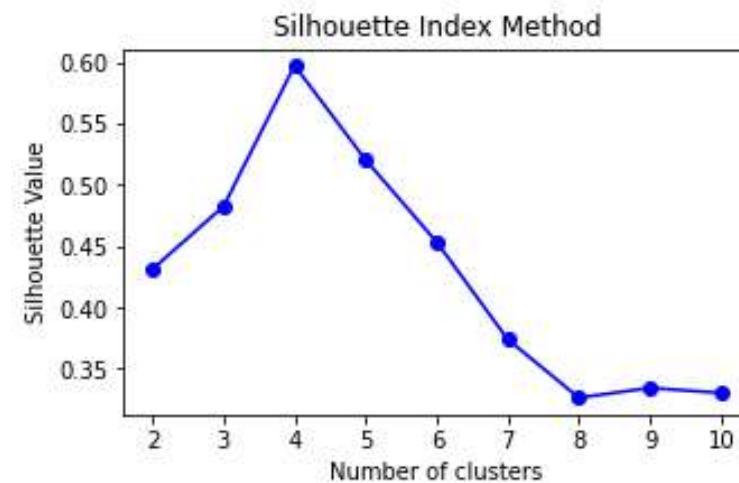
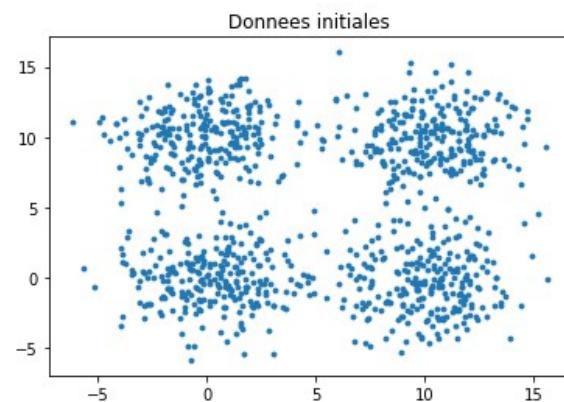
# Evaluation avec le coefficient de silhouette

## • Coefficient de silhouette

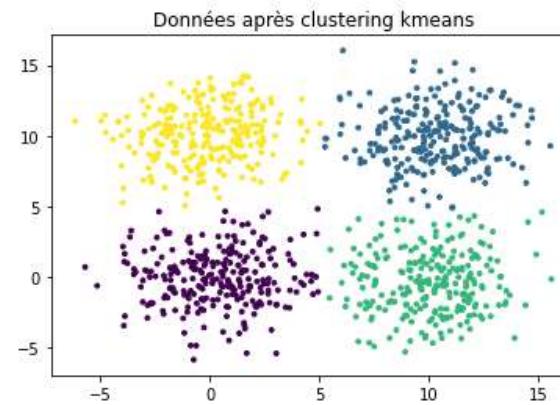
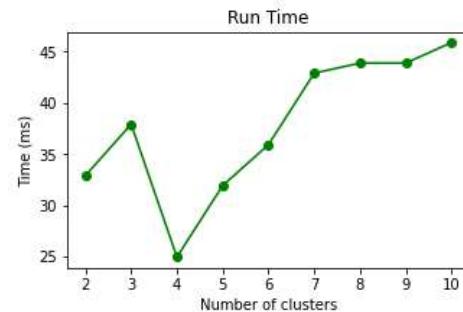
- Indicateur utilisé lorsqu'il n'y a pas de vérité de terrain
- Basé sur deux mesures
  - **Proximité** : l'exemple  $x$  est-il dans le bon cluster ?
    - $a(x)$  : distance moyenne entre  $x$  et les autres exemples du même cluster
      - $$a(x) = \frac{1}{n_k - 1} \sum_{y \in C_k, y \neq x} d(x, y)$$
  - **Séparation** : l'exemple  $x$  est-il éloigné des autres clusters ?
    - $b(x)$  : distance moyenne **minimale** aux exemples des autres clusters (ie. Au cluster le plus proche)
      - $$b(x) = \min_{l \neq k} \frac{1}{n_l} \sum_{y \in C_l} d(x, y), x \in C_k$$
- Pour chaque exemple  $x$ : combiner les deux mesures :
  - $$s(x) = \frac{b(x) - a(x)}{\max(a, b)}; \quad s(x) \in [-1, 1]$$
- Pour le jeu de données : moyenne sur tous les points  $S = \frac{1}{n} \sum s(x)$
- Score silhouette de 1 : clusters très denses et bien séparés

# Illustration

- **Application incrémentale méthode kmeans**

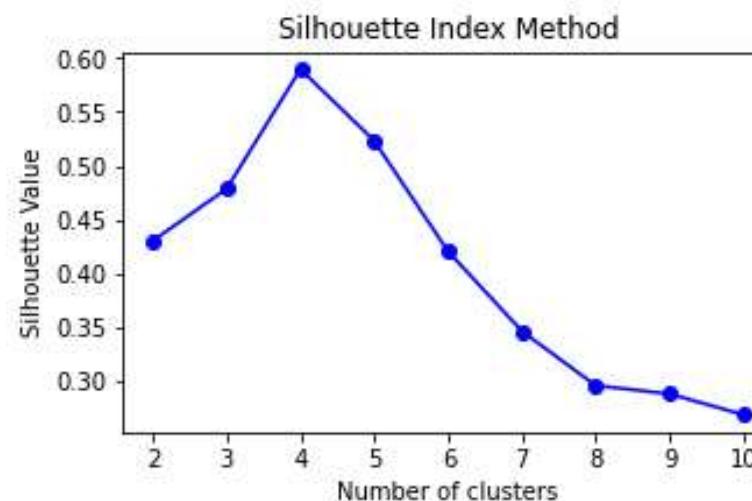
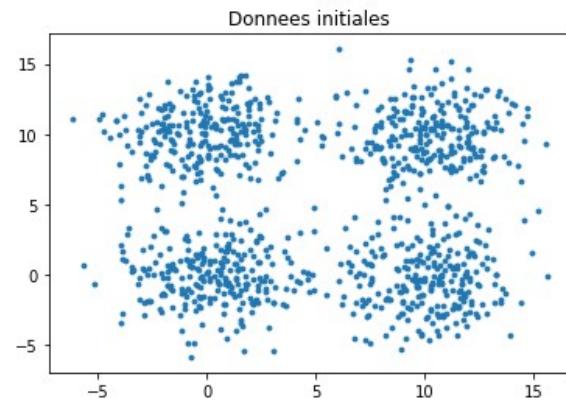


- Nombre de clusters présentant la valeur de silhouette la plus proche de 1 :  $k = 4$

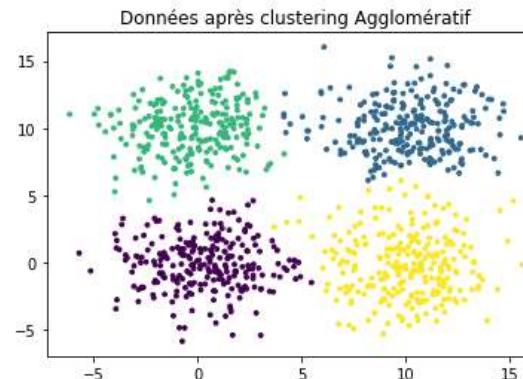
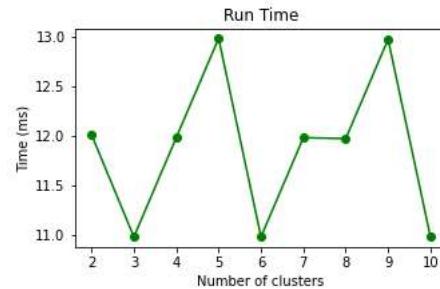


# Illustration

- **Application incrémentale clustering hiérarchique (average)**

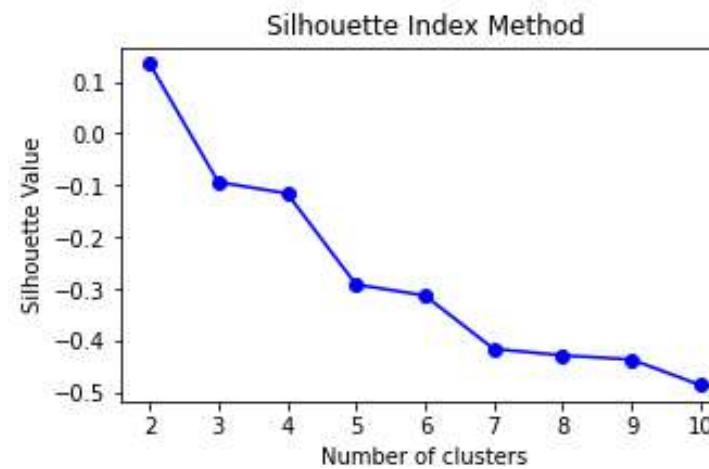
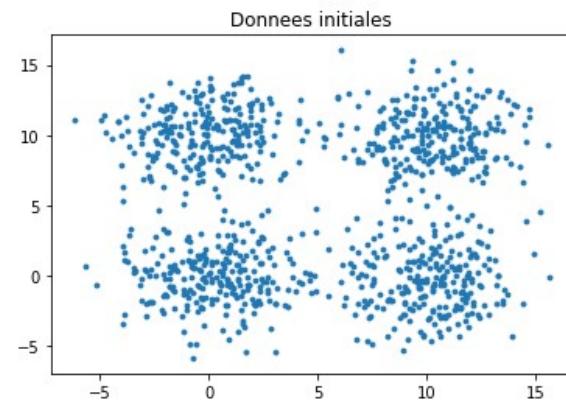


- Nombre de clusters présentant la valeur de silhouette la plus proche de 1 :  $k = 4$

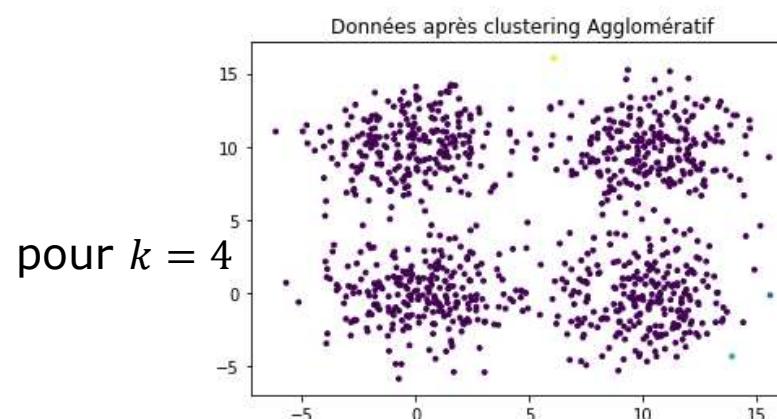
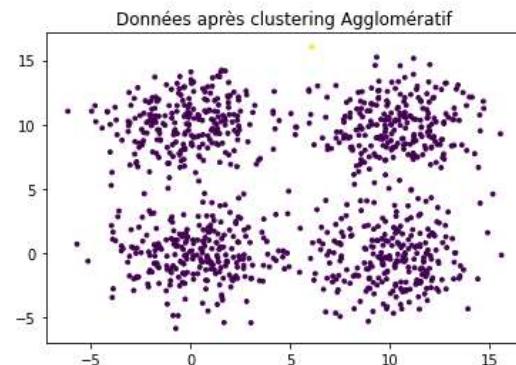


# Illustration

- **Application incrémentale clustering hiérarchique (single)**



- Nombre de clusters présentant la valeur de silhouette la plus proche de 1 :  $k = 2 \dots$



# Evaluation avec l'indice de Davies Bouldin

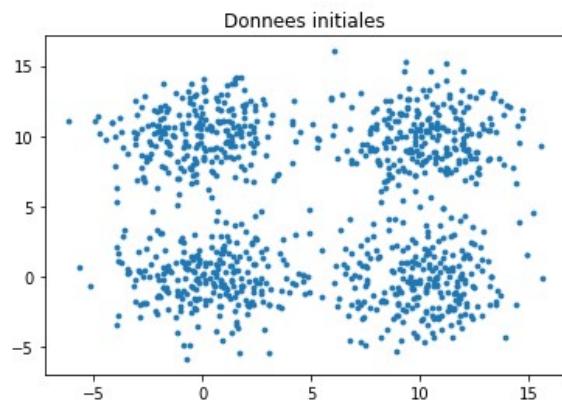
## • Indice de Davies-Bouldin (DB) basé sur

- **Cohésion (homogénéité)** = qualité intra-cluster (**Diamètre moyen**)
  - pour un cluster  $k$  : moyenne des distances entre chaque point et le centre  $\mu_k$
  - $H_k = \frac{1}{n_k} \sum_{x \in C_k} \delta(x, \mu_k)$
  - Un cluster homogène a une valeur  $H_k$  faible
- **Séparation** = qualité inter-cluster
  - Pour 2 clusters  $k$  et  $l$  : distance entre leurs centres
  - $S_{k,l} = \delta(\mu_k, \mu_l)$

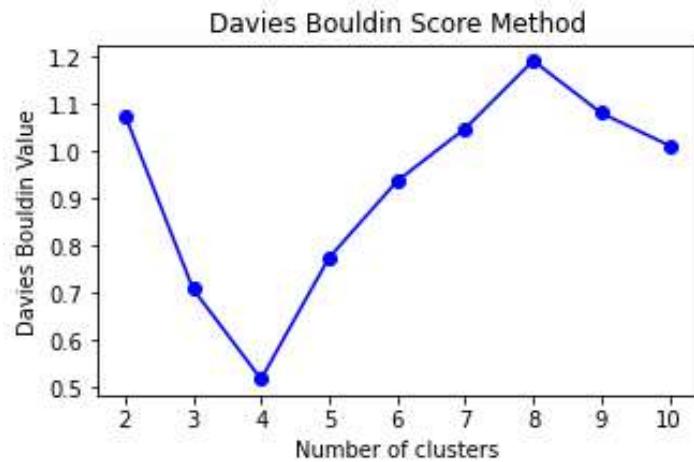
## • Indice DB : combiner les 2 mesures

- $DB = \frac{1}{K} \sum_{k=1}^K DB_k$ , avec  $DB_k = \max_{l \neq k} \left( \frac{H_k + H_l}{S_{k,l}} \right)$
- Valeur faible si les clusters sont homogènes (numérateur petit) et s'ils sont bien séparés (dénominateur grand)
- Minimiser DB → aide pour déterminer le nombre de clusters

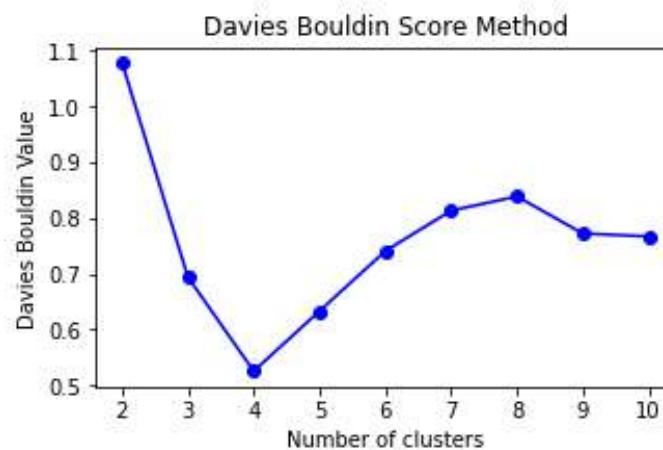
# Illustration : indice de Davies Bouldin



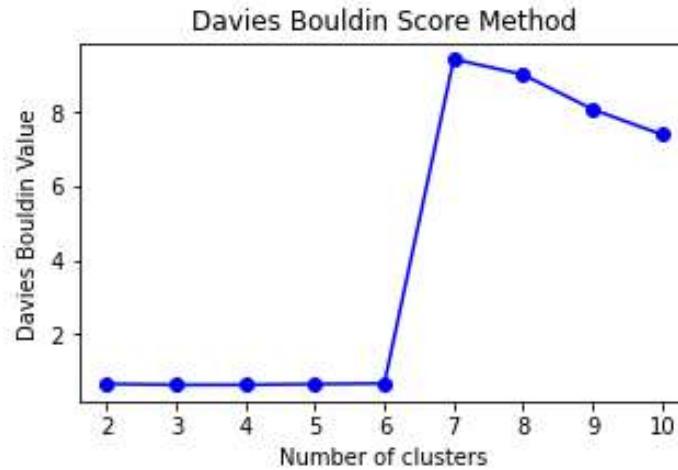
Kmeans incrémental



Agglomératif (average) incrémental



Agglomératif (single) incrémental

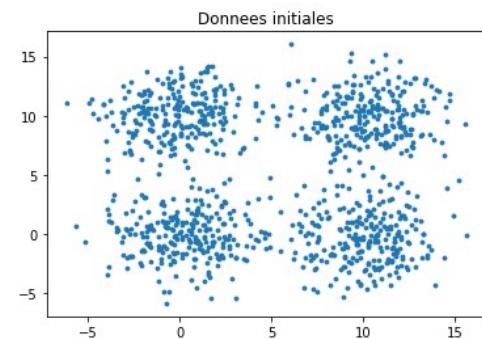
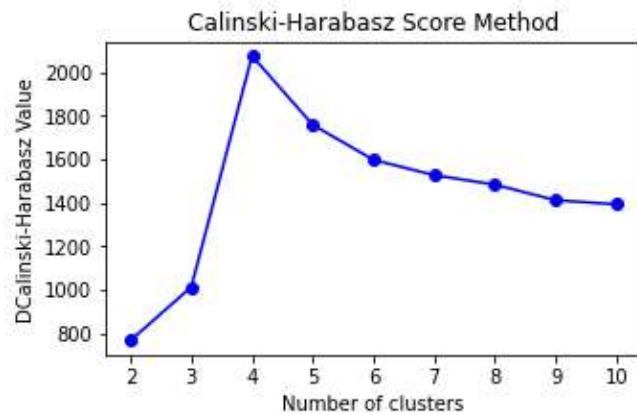


# Evaluation avec indice de Calinski-Harabasz

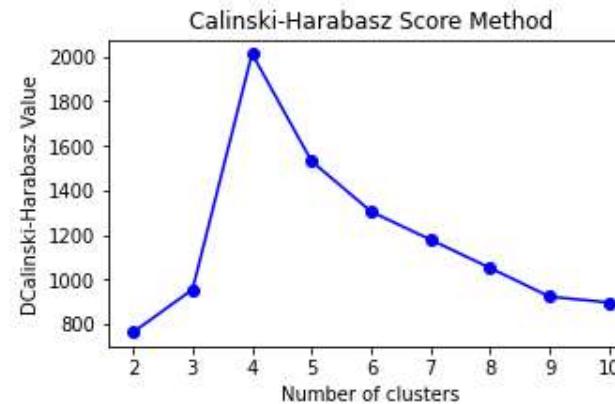
## • Indice de Calinski-Harabasz

- Manière différente de déterminer la qualité d'un clustering et le compromis entre homogénéité et séparation
  - Notion de dispersion interne et entre cluster
  - Indice CAH :
    - Élevé pour clustering de bonne qualité

### Kmeans incrémental



### Agglomératif (average) incrémental



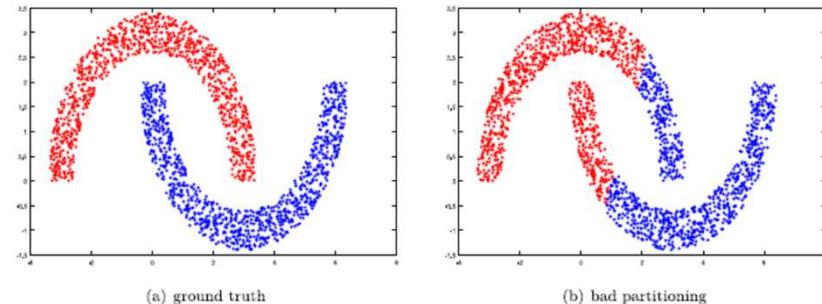
# Comparaison de métriques d'évaluation

- **Présentation de quelques métriques proposées dans scikitlearn**
  - Lorsque pas de label connu
  - Coefficient de silhouette
  - Indice Davies-Bouldin
  - Indice de Calinski Harabasz
- **Comparatif**
  - Article « An extensive comparative study of cluster validity indices ”
    - Arbelaitz et. al Pattern Recognition, 2013
    - Evaluation de 28 indicateurs pour 3 méthodes (Kmeans, Ward, average-linkage)
    - Meilleur clustering : celui le plus proche de la partition correcte
      - Utilisation d'une vérité de terrain
      - 3 mesures de comparaison à cette vérité de terrain
    - Jeux de données générés et réels
    - Indicateurs précédents dans le groupe de tête

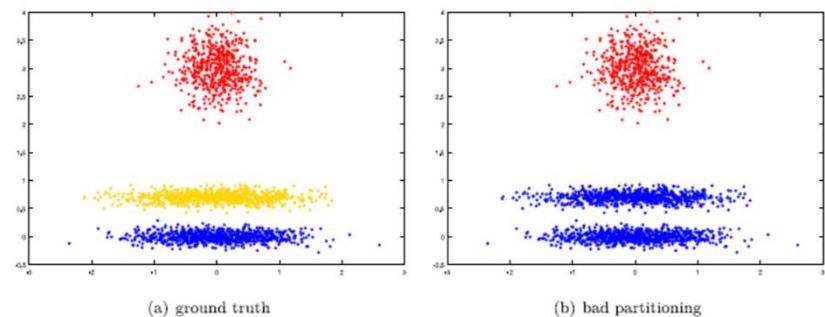
# Autres métriques d'évaluation

- **Limites de ces métriques**

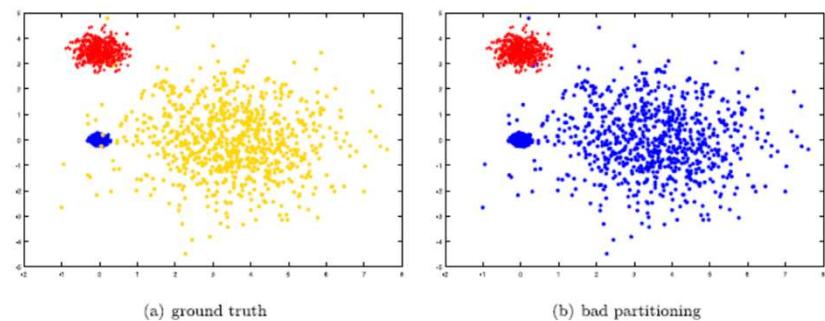
- Clusters sphériques / convexes
  - Distance d'un point au centre de son cluster inférieure à celle du centre d'un autre cluster



- Cluster bien séparés



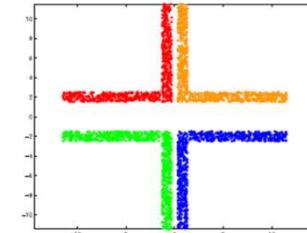
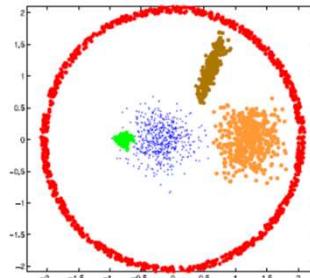
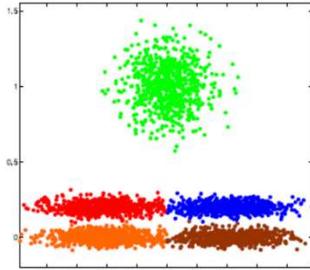
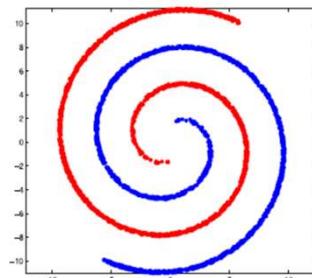
- Variabilité dans la densité



# Autres métriques

- **Voir articles**

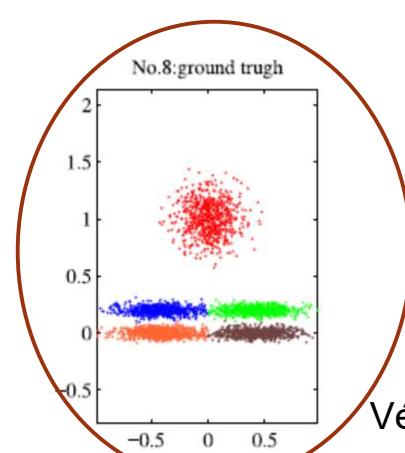
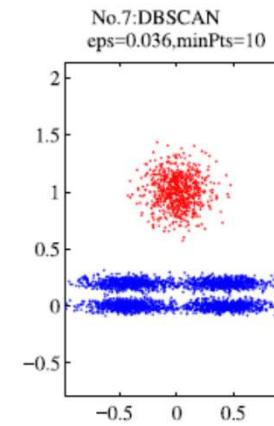
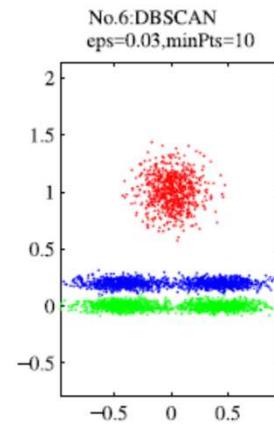
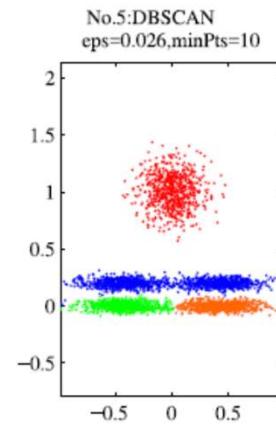
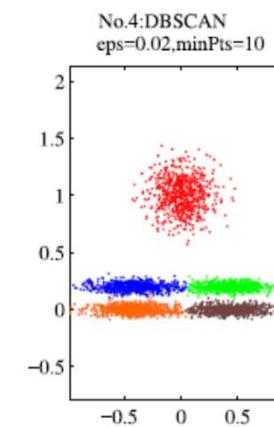
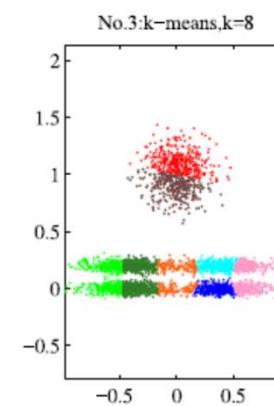
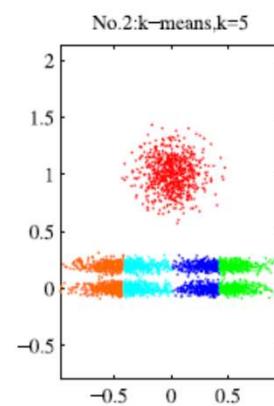
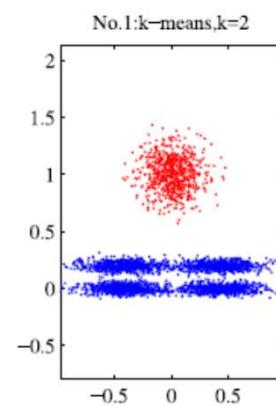
- Cluster Validity Index for irregular clustering results – S. Liang, D. Hen, Y. Yang
  - Applied Soft Computing Journal vol. 95 – 2020
- 12 métriques comparées sur des jeux de données complexes
- Validation de ces métriques par rapport à une vérité de terrain (les jeux de données ont été construits pour cela ...)



- Validation sur des jeux de données « réel »

# Autres métriques

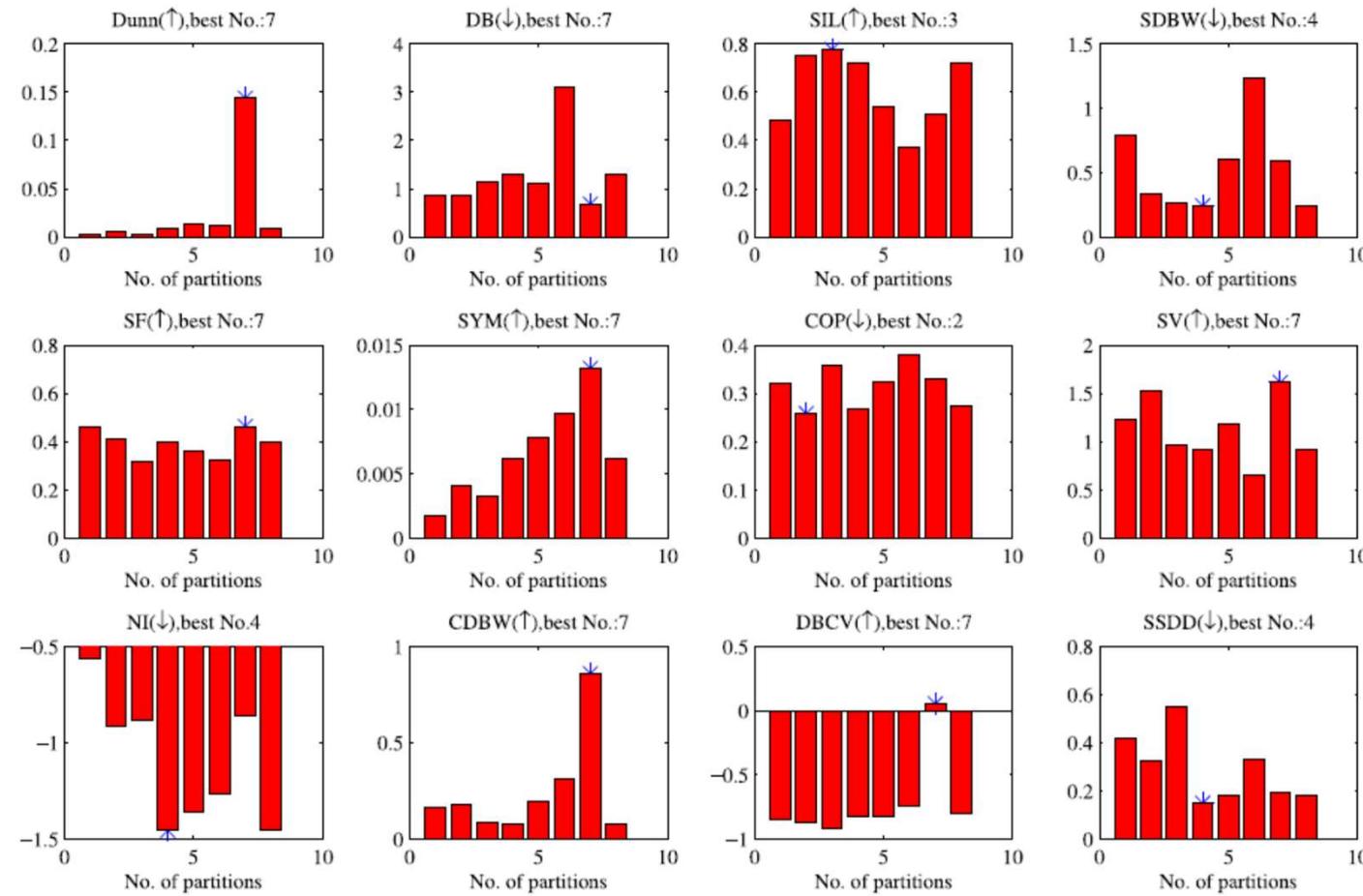
- **Illustration**
  - Application de 7 méthodes de clustering



Vérité

# Autres métriques

- **Illustration**
  - Evaluation de 12 métriques



# En pratique

(Fin séance 2)

---

- **Soyez vigilants**

- Les métriques utilisées sont-elles adaptées au problème de clustering considéré ?
- Vous pouvez implémenter/rechercher des métriques complémentaires !

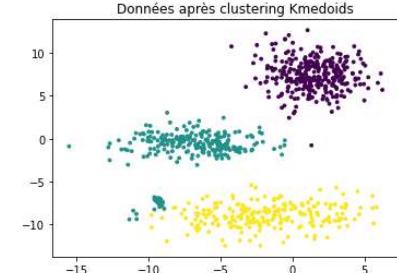
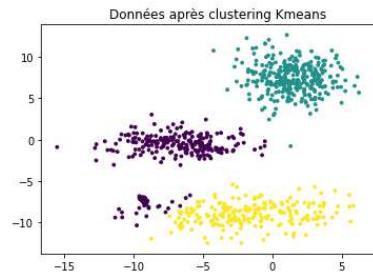
# Evaluation d'un clustering

- **Comparaison à un clustering connu**
  - Métriques « d'évaluation externes »
  - Evaluer un clustering en le comparant au bon clustering
  - Indicateurs de comparaison :
    - Rand, Jaccard, ....
    - Indicateur de similarité entre partitions
  - Suppose l'existence d'un clustering de référence
    - Si on connaît une vérité de terrain, pourquoi effectuer un clustering au lieu d'un apprentissage supervisé ?

# Métriques d'évaluation externes

## • Exemple : indice de Rand

- Pour un ensemble  $X$  de  $n$  éléments et soient deux solutions de clustering :  $A = \{A_1, \dots, A_r\}$  et  $B = \{B_1, \dots, B_s\}$
- Nombre d'accords :  $a + b$ 
  - $a$  = nombre de paires d'exemples qui sont dans un même cluster de  $A$  et dans un même cluster de  $B$
  - $b$  = nombre de paires d'exemples qui sont dans des clusters différents dans  $A$  et dans des clusters différents dans  $B$
- Nombre de désaccords :  $c + d$ 
  - $c$  = nombre de paires d'exemples de  $X$  qui sont dans un même cluster dans  $A$  et dans des clusters différents dans  $B$
  - $d$  = nombre de paires d'exemples qui sont dans des clusters différents dans  $A$  et dans un même cluster dans  $B$
- $R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}}$



# Evaluation d'un clustering

---

- **Stabilité des clusters**

- Certaines méthodes ne sont pas déterministes
- Répéter la méthode pour les mêmes valeurs de  $k$
- Regrouper dans un même cluster final les éléments qui se retrouvent toujours dans les mêmes clusters intermédiaires

- **Cohérence / expertise**

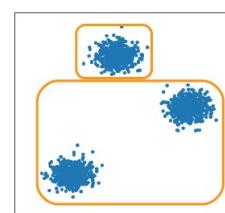
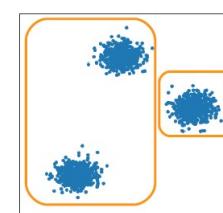
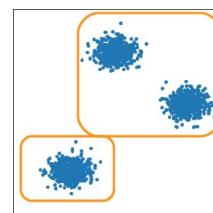
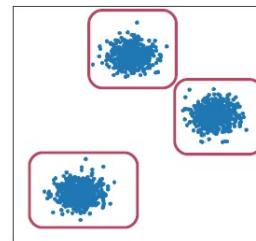
- Évaluation par expert humain ...
- Évaluation « manuelle »
- Vérification sur un sous-ensemble de données

# Stabilité d'un clustering

## • Problématique :

- Méthodes non déterministes : résultats différents si plusieurs exécutions

- lancer la méthode plusieurs fois avec
  - initialisation différente,
  - des sous-ensembles différents,
- Est-ce que les points sont regroupés de manière similaire ?
- Ex : problème de stabilité pour K=2



- Stable pour K=3

- **Indice de Rand** : comparer 2 solution de clustering en ignorant les permutations
  - Nombre de paires dans le même cluster / nb de paires dans des clusters différents

# Plan

---

## **1. Caractérisation du problème de clustering**

1. Données
2. Distances
3. Problème de partition
4. Synthèse

## **2. Quelques Méthodes**

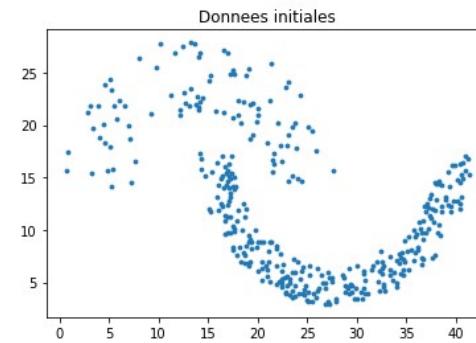
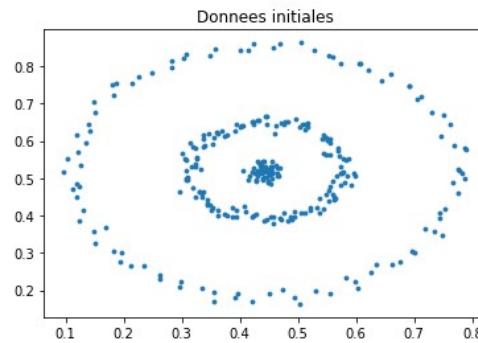
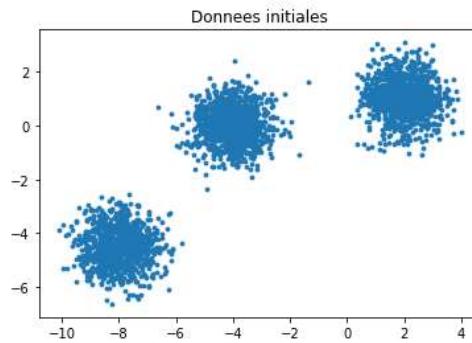
1. Méthodes basées centres de masses
2. Méthodes hiérarchiques
3. **Méthodes basées voisinage (densité)**
4. Méthodes basées graphes

## **3. Bilan Clustering**

1. Evaluation d'un clustering
2. Application

# Objectifs clustering basés densité

- **Déetecter des clusters de formes quelconques**

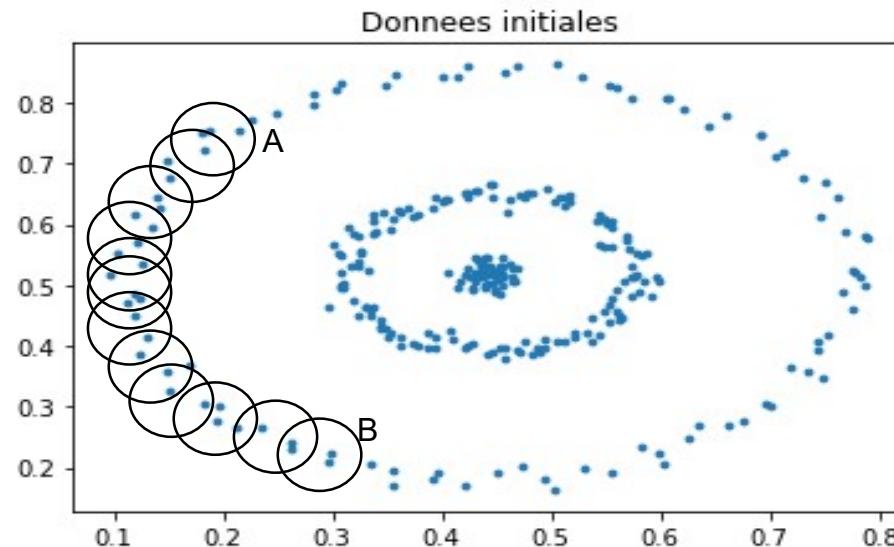


- **Clusters**
  - Zones denses séparées par des zones peu denses
- **Plusieurs méthodes**
  - Focus sur la méthode DBSCAN
    - *1996 : A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*" Ester, M., H. P. Kriegel, J. Sander, and X. Xu, 2nd International Conference on Knowledge Discovery and Data Mining

# Méthode DBSCAN (1)

- **Principe**

- Réaliser des clusters en se basant sur une mesure densité
- Placer deux exemples dans un même cluster :
  - Ils sont reliés par un « chemin » permettant de les relier tout en restant dans un même cluster



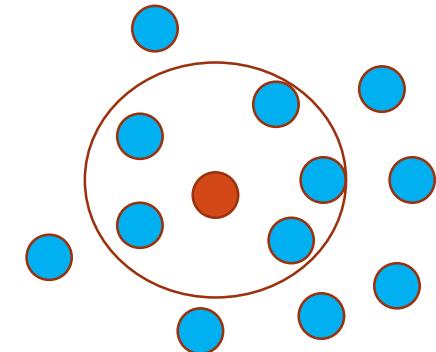
- Notion de voisinage d'un exemple

# Méthode DBSCAN (2)

- **Voisinage**

- **Taille (Fenêtre de visibilité) :  $\epsilon$**

- Epsilon voisinage :  $N_\epsilon(x_i) = \{x_j \in X \mid d(x_i, x_j) < \epsilon\}$

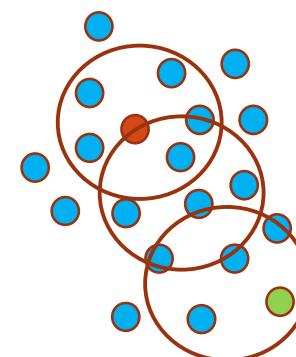
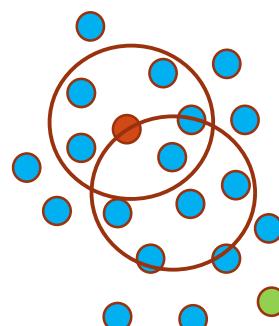
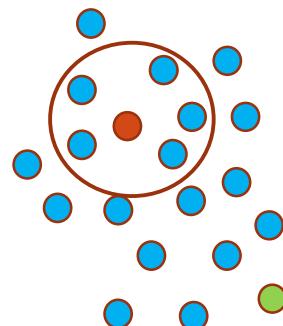


- **Nombre de voisins (densité) :  $min_{pt}$**

- Point intérieur  $x_i$  : si  $|N_\epsilon(x_i)| \geq min_{pt}$

- **Points connectés par densité** :  $x_i$  et  $x_j$  sont connectés si

- Il existe une suite de points intérieurs  $y_1, y_2, \dots, y_m$  tels que
      - $y_1 \in N_\epsilon(x_i), y_2 \in N_\epsilon(y_1), \dots, x_j \in N_\epsilon(y_m)$



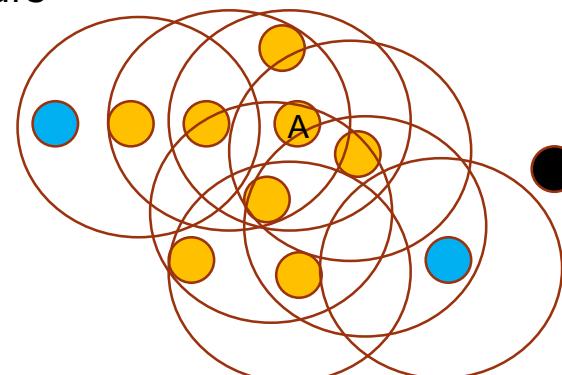
# Méthode DBSCAN (3)

---

- **Différentes étapes**
  - Pour chaque exemple : déterminer le nombre d'exemples dans son voisinage limité par une taille epsilon
  - Si un exemple a un nombre de voisins suffisamment important : il est considéré comme un exemple « cœur/intérieur »
    - Détection d'un exemple dans une zone de forte densité
  - Tous les exemples dans le voisinage d'un exemple cœur sont placés dans un même cluster
    - Séquence d'exemples cœur connectés de proche en proche
- **Anomalie (bruit)**
  - Un exemple qui n'est pas déterminé comme cœur et dont le voisinage ne comporte pas d'exemples cœur est considéré comme une anomalie

# Exemple

- Seuil  $\min_{pt}=4$ 
  - **Exemples cœurs / intérieurs** : tous les points en orange
    - Même cluster que A
  - **Exemples atteignables** : tous les points en bleu
    - Ne sont pas des points intérieurs
    - Taille de voisinage trop faible
    - Mais dans voisinage de points intérieurs
    - Même cluster que A
  - **Exemples atypiques** : point en noir
    - Ne sont pas atteignables par les points intérieurs existant
    - Ne sont pas eux-mêmes des points intérieurs

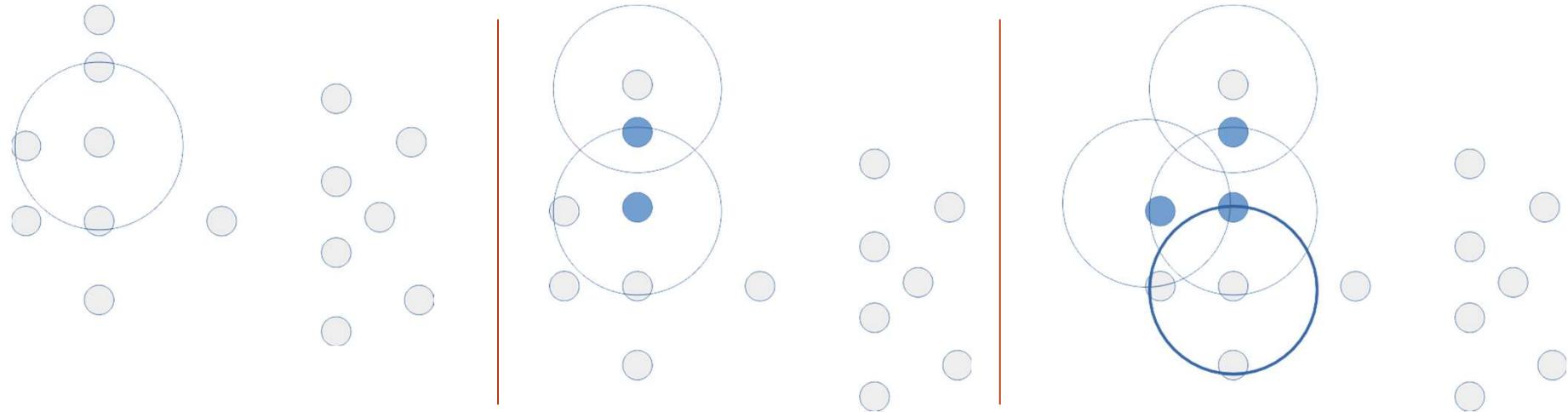


# Algorithme DBSCAN

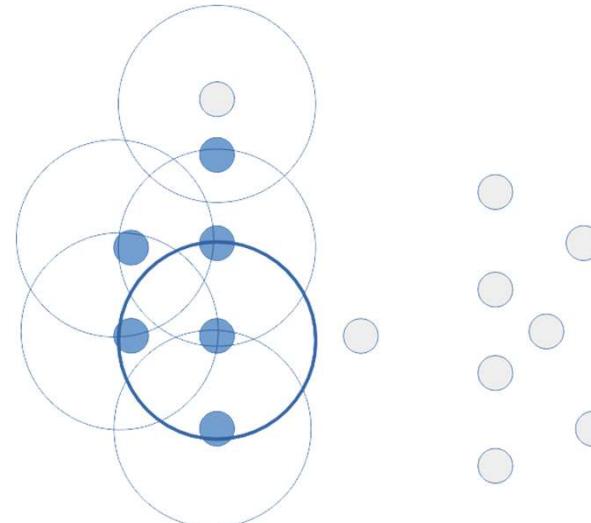
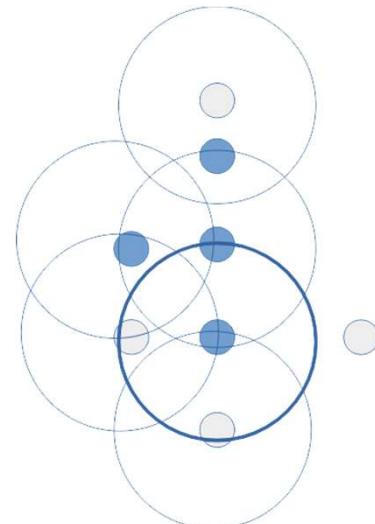
```
DBSCAN(Data, eps, MinPts)
C=0
Pour chaque exemple p de Data / p non visité
    p.etat ← Visité
    Voisins ← EpsilonVoisinage(p, Data, eps)
    si taille(Voisins) < MinPts alors
        p.etat ← Bruit
    sinon
        C ← C+1
        EtendreCluster(p, C, data, Voisins, eps, MinPts)
Fin pour

EtendreCluster(p, C, data, Voisins, eps, MinPts)
    p.cluster ← C
    tant que Voisins ≠ Ø faire
        sélectionner p' dans Voisins
        si p' non visité alors
            p'.etat ← visité
            VoisinsSuivant ← EpsilonVoisinage(p', Data, eps)
            si taille(VoisinsSuivant) ≥ MinPts alors
                Ajouter(VoisinsSuivant, Voisins)
            si p'.cluster=null alors
                p'.cluster ← C
        fin tantque
```

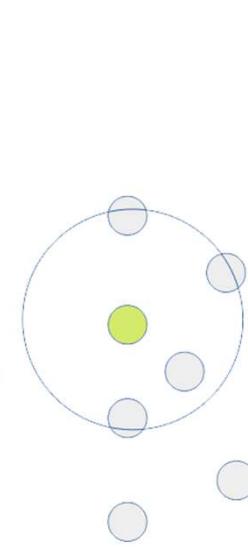
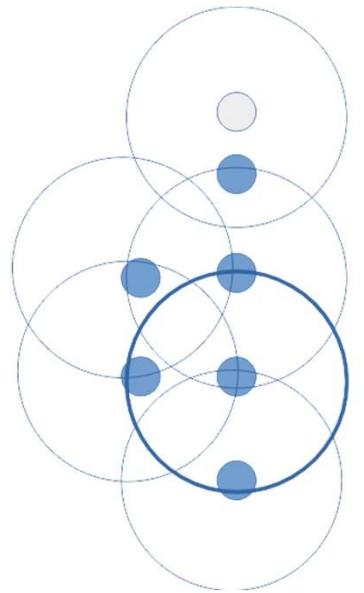
# Exemple



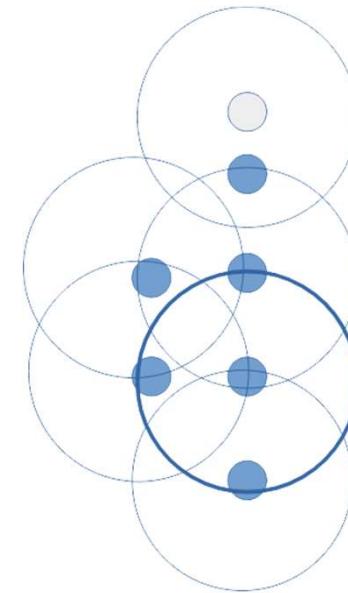
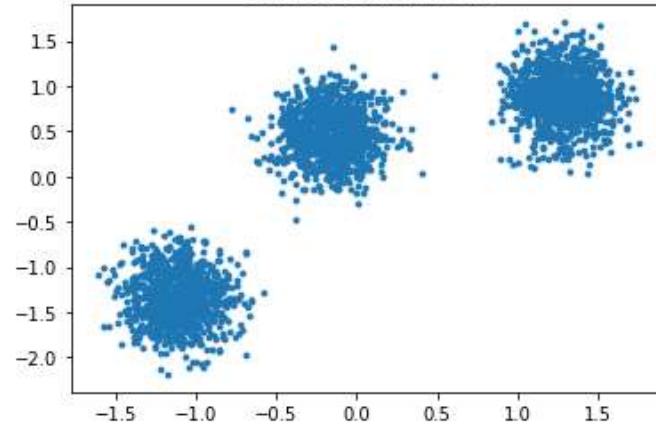
MinPts=4



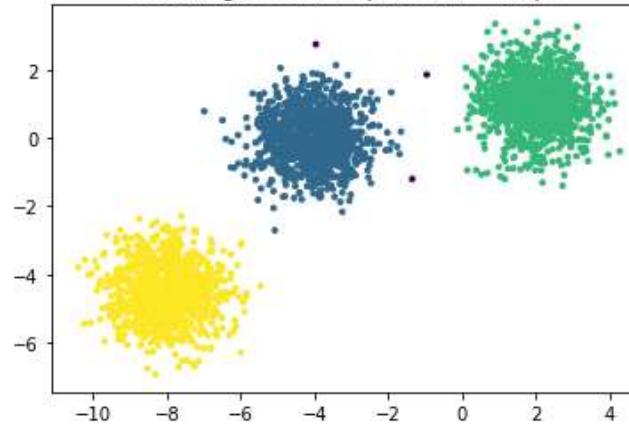
# Exemple



Données standardisées



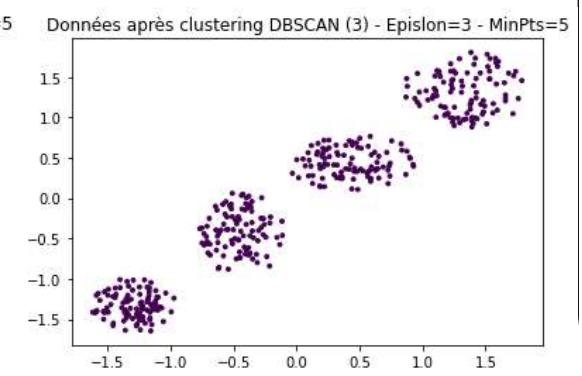
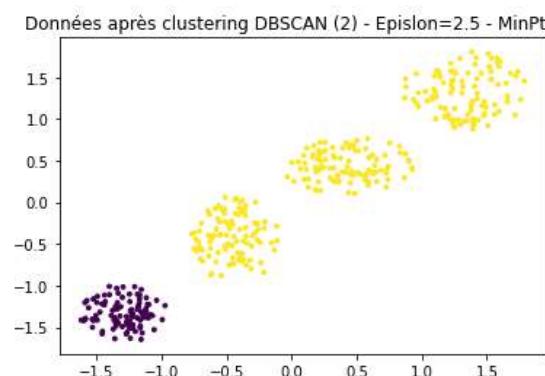
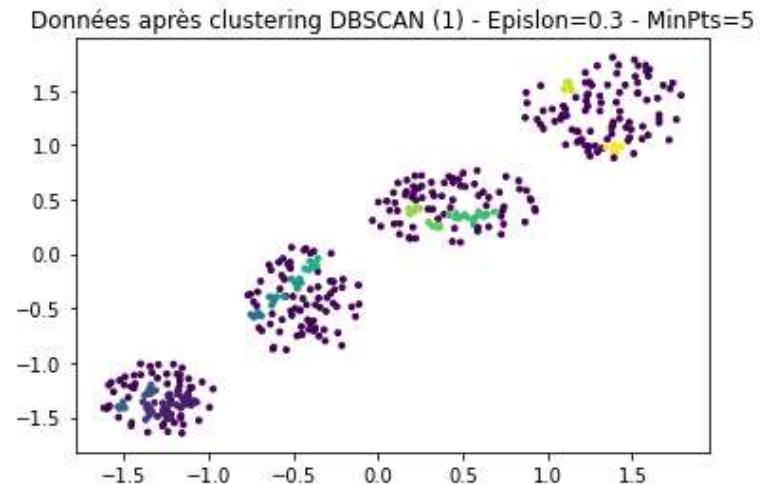
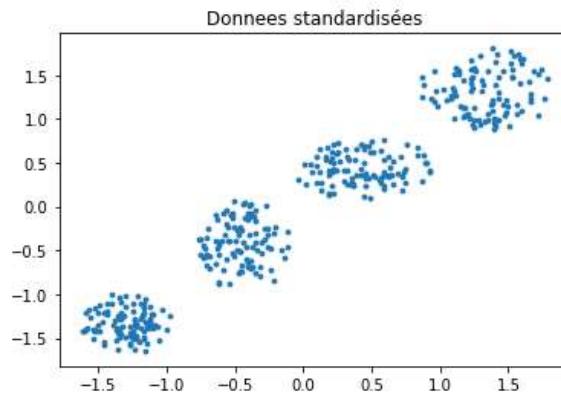
Clustering DBSCAN - Epsilon=.... - Minpt=



# Déterminer les paramètres de DBSCAN

## • Difficultés

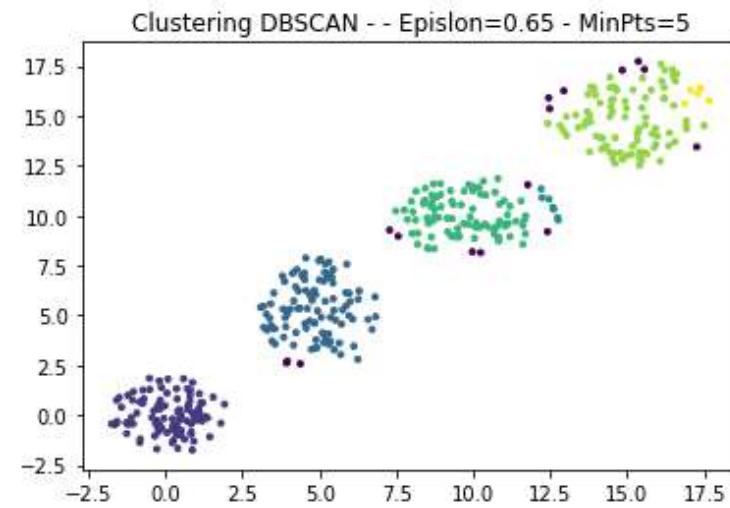
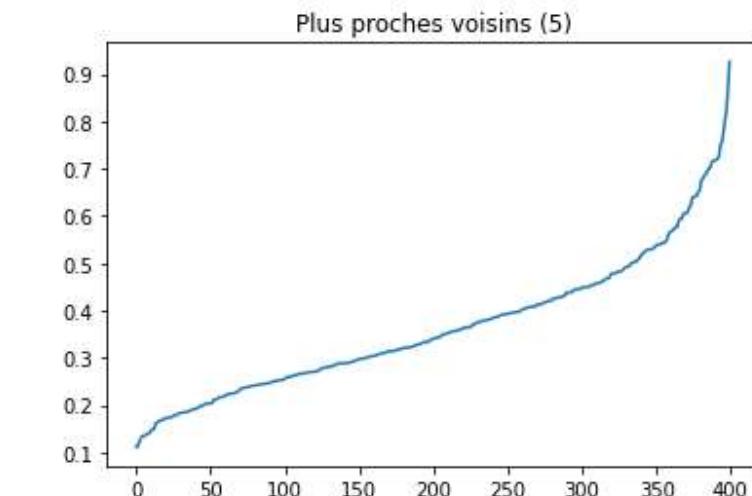
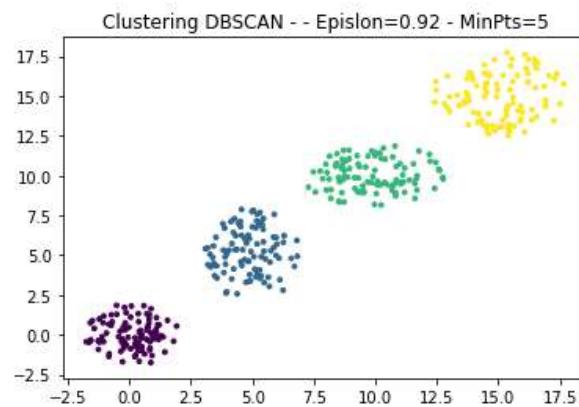
- Valeur epsilon trop faible : beaucoup d'anomalies
- Valeur epsilon trop grande : tous les exemples vont être dans le même cluster



# Déterminer les paramètres de DBSCAN

## • Aide

- Fixer Taille du voisinage
- Rechercher epsilon
- Pour chaque exemple :
  - Quelle est la distance pour ses k voisins ?
  - Fixer epsilon pour que « presque » chaque exemple ait k voisins à distance epsilon



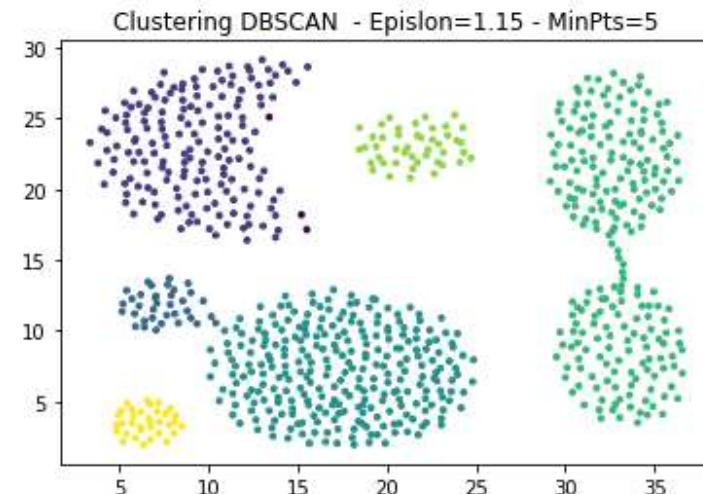
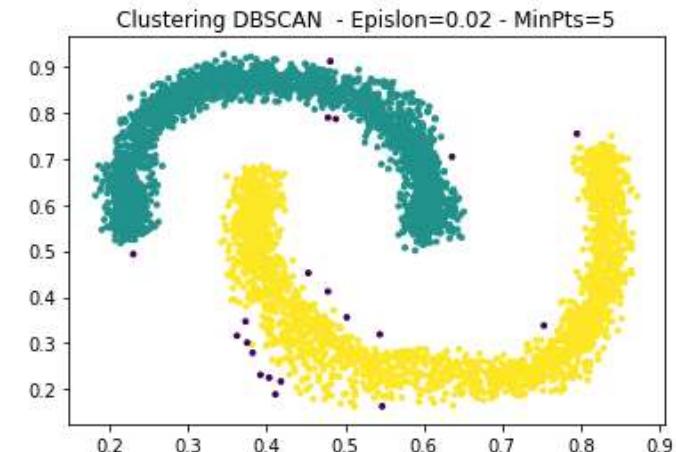
# Caractéristiques DBSCAN (1)

- **Intérêts**

- Pas besoin de fixer le nombre de cluster
- Peut déterminer des clusters non convexes
- Est robuste au bruit et anomalies

- **Difficulté**

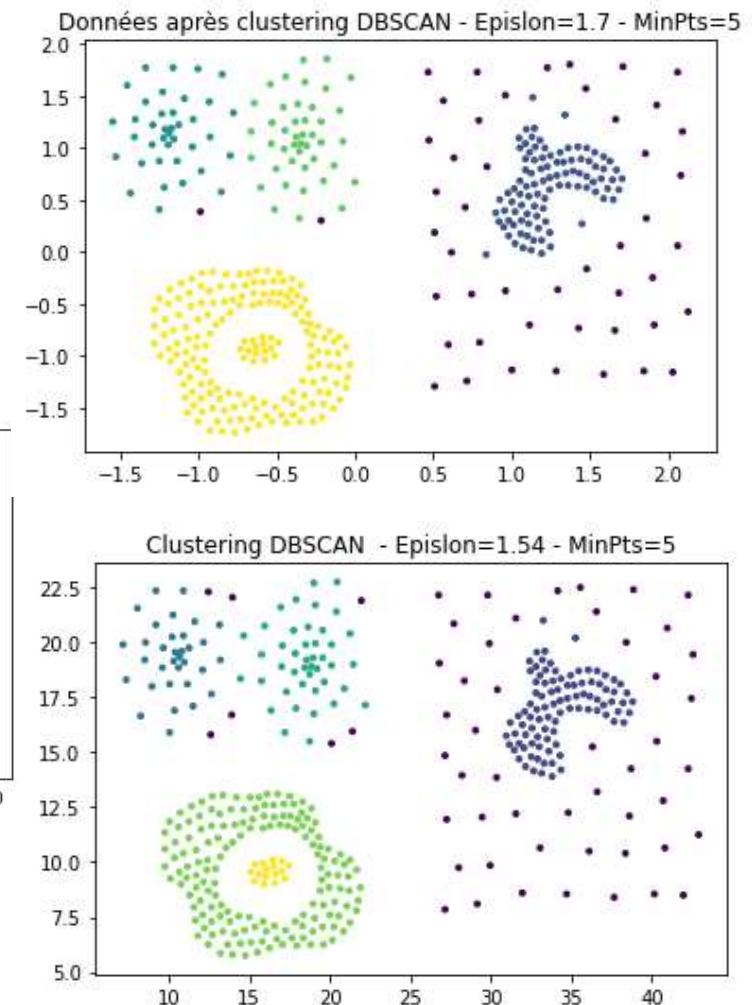
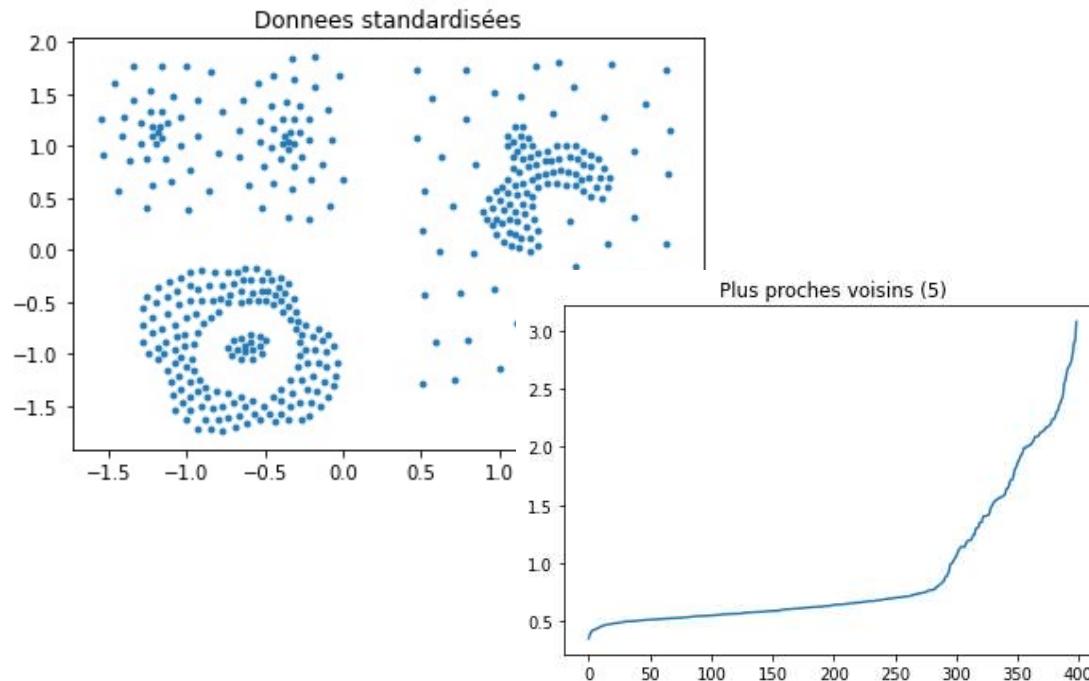
- Paramètres à déterminer
  - Fixer la taille du voisinage et le nombre de voisins à considérer



6 clusters ...

# Caractéristiques DBSCAN (2)

- **Limites**
  - Densité variable dans les données



- Grandes dimensions
  - Temps de calcul

# Pour les TP

## • Méthode AgglomerativeClustering de scikitlearn

- Sklearn.cluster.DBSCAN

### • Paramètres principaux

- eps : default = 0,5
- min\_samples : default = 5
- metric : (euclidian par défaut)

<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html#sklearn.cluster.DBSCAN>

### • Résultats

- labels\_ : labels de chaque exemple

### Méthodes :

- fit : pour déterminer le clustering d'un jeu de données
- predict : pour déterminer les clusters de nouveaux exemples

## • Plus proches voisins (non supervisé)

- sklearn.neighbors import NearestNeighbors

# Méthode HDBSCAN

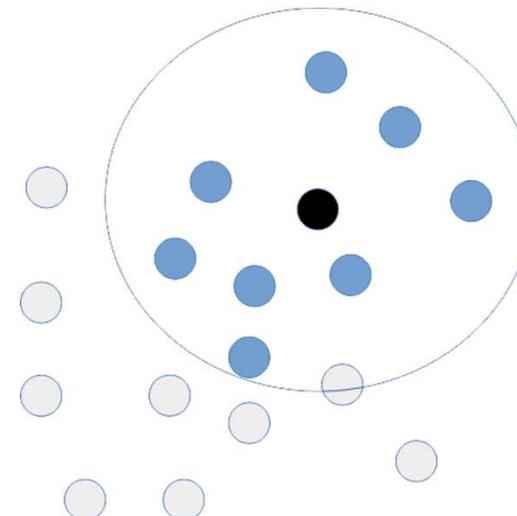
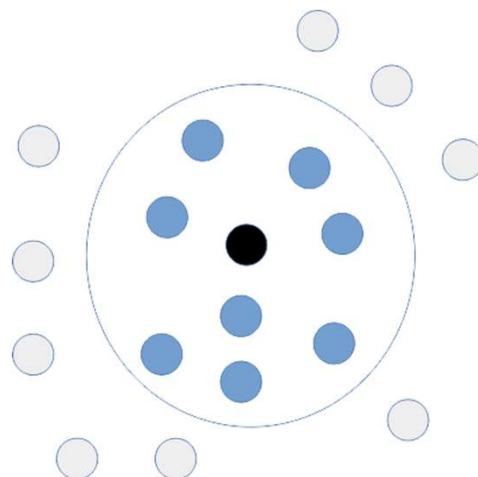
---

- **Objectif**
  - Prendre en compte des densités variables
  - HDBSCAN : Hierarchical DBSCAN
    - Combinaison de DBCSAN et de clustering hiérarchique
    - Simplifier le réglage de paramètres de DBSCAN
    - Passer à l'échelle
- Articles :
  - 2013 : R. J. G. B. Campello, D. Moulavi, et J. Sander, « *Density-Based Clustering Based on Hierarchical Density Estimates* », in *Advances in Knowledge Discovery and Data Mining*
  - 2017 : L. McInnes et J. Healy, « *Accelerated Hierarchical Density Clustering* », *IEEE International Conference on Data Mining Workshops*

# Principes HDBSCAN (1)

## • Voisinage et densité

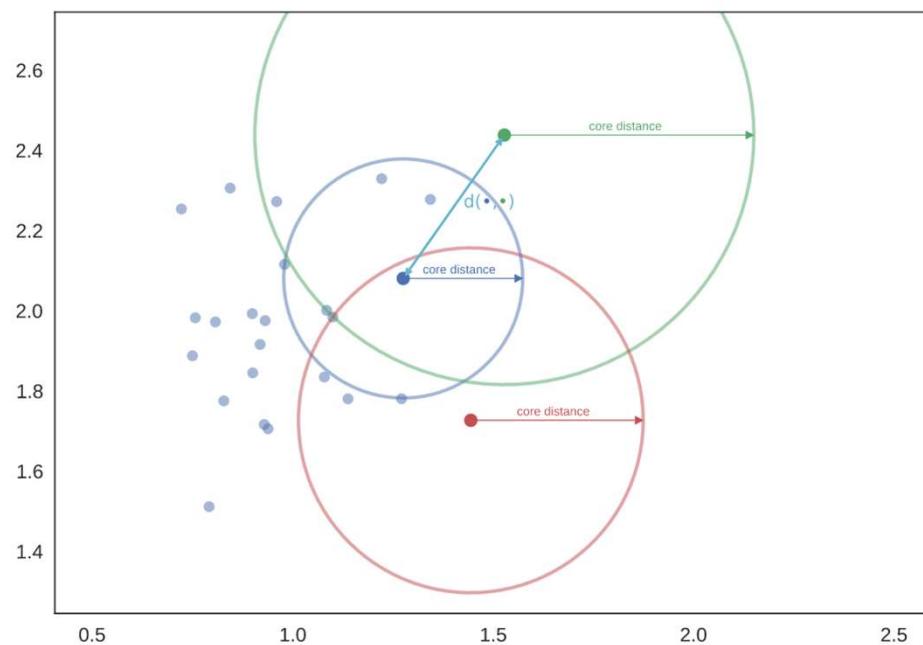
- Se donner un nombre d'exemples : MinPts
- Pour chaque exemple : déterminer la distance permettant de couvrir MinPts exemples : core distance



- Quand la distance est grande : la densité d'exemples est faible
- Quand la distance est faible : la densité d'exemples est élevée

## Principes HDBSCAN (2)

- **Construire un graphe basé sur distance d'accessibilité mutuelle entre exemples**
  - Distance d'accessibilité mutuelle entre deux exemples X et Y
    - $\text{Max}(\text{core-distance}(X), \text{core-distance}(Y), \text{distance}(X,Y))$



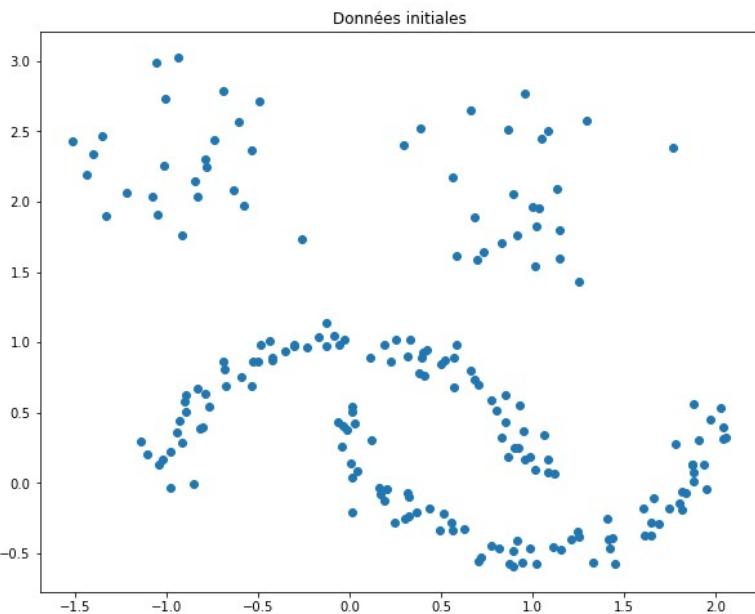
Est-ce que les exemples sont dans des zones denses ?  
Est-ce qu'ils sont proches l'un de l'autre ?

# Principes HDBSCAN (3)

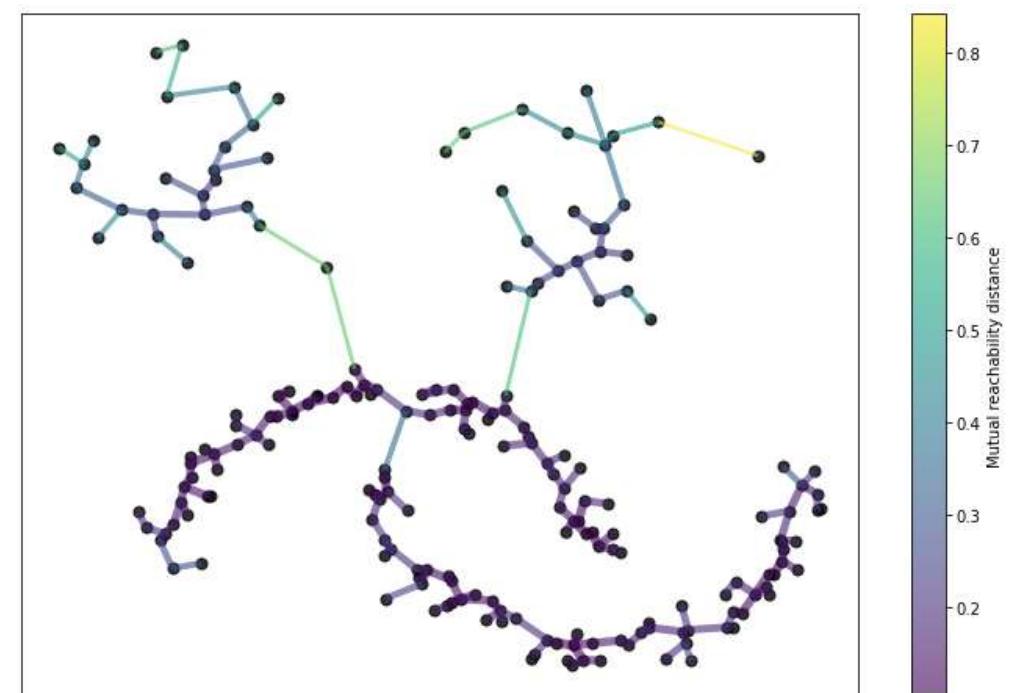
- **Graphe d'accessibilité**

- Graphe complet
- Construire un arbre couvrant de poids minimal

- **Exemple**

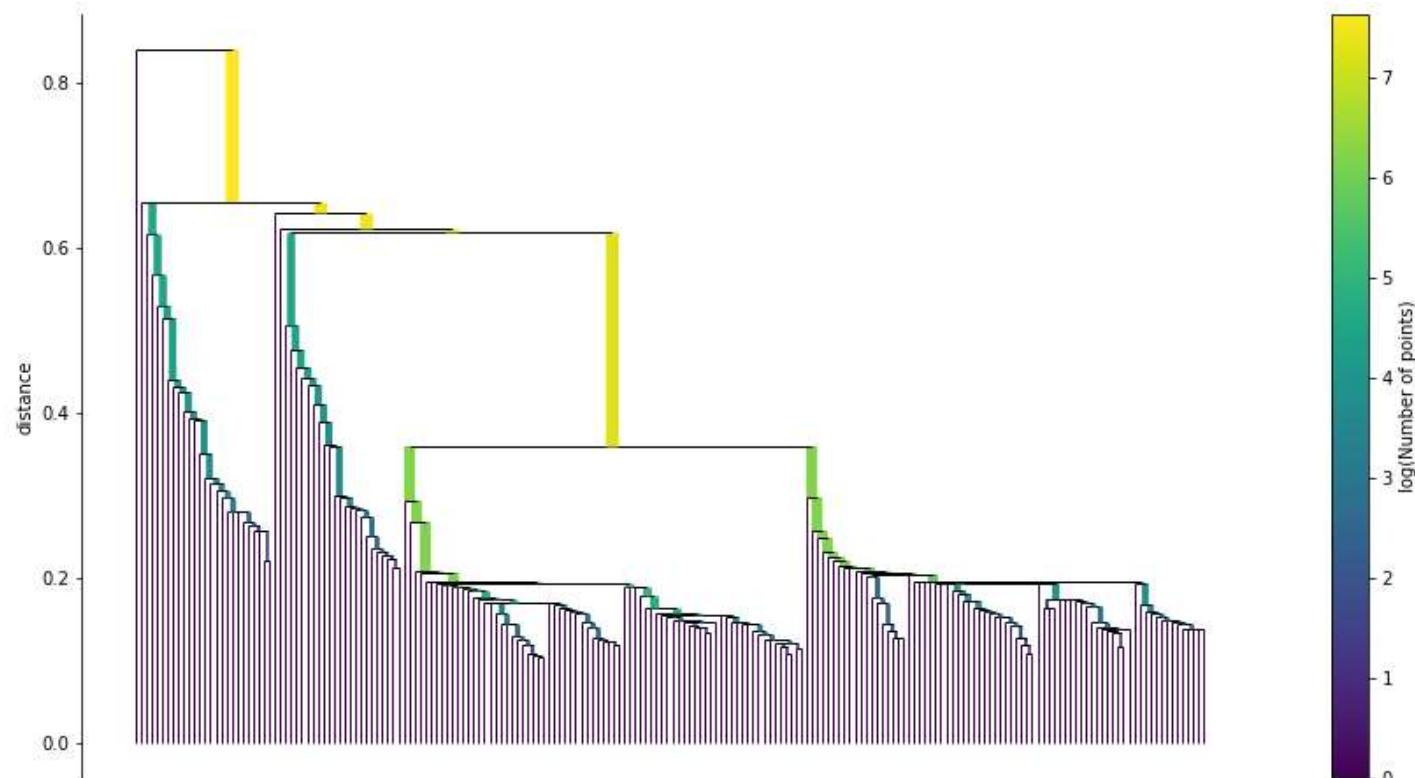


Graphe d'accessibilité (arbre couvrant) – MinPts=5



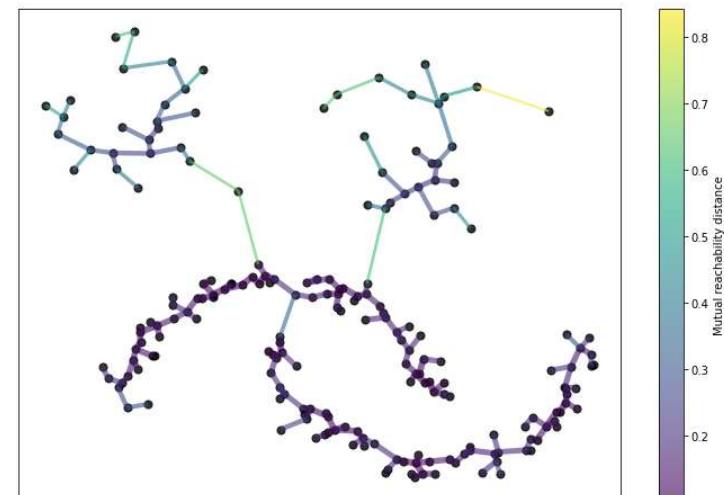
# Principes HDBSCAN (4)

- **Clustering hiérarchique sur l'arbre couvrant**
  - Obtention d'un dendrogramme (par distance croissante)



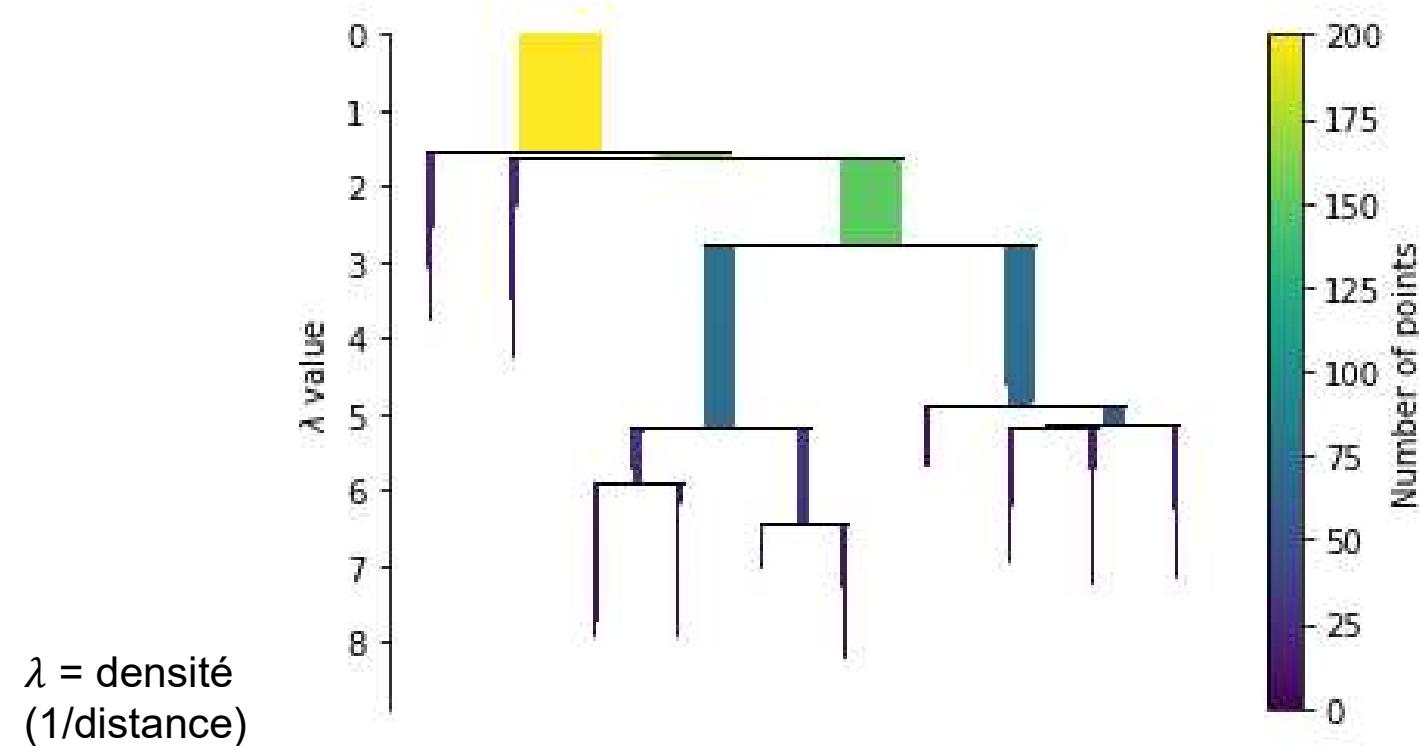
# HDBSCAN et DBSCAN

- **Graphe (Arbre) d'accessibilité**
  - Fait apparaître des zones plus denses
  - Lien avec DBSCAN
    - Regrouper des exemples ayant au moins MinPts dans son epsilon-voisinage
    - Dans un cluster 2 exemples X et Y sont reliés si :
      - X (Y) est dans le epsilon-voisinage de Y (X)
      - Il y a une suite d'exemple reliés par epsilon-voisinage suffisamment dense
    - Distance d'accessibilité mutuelle inférieure à epsilon



# Autres extensions dans HDBSCAN (1)

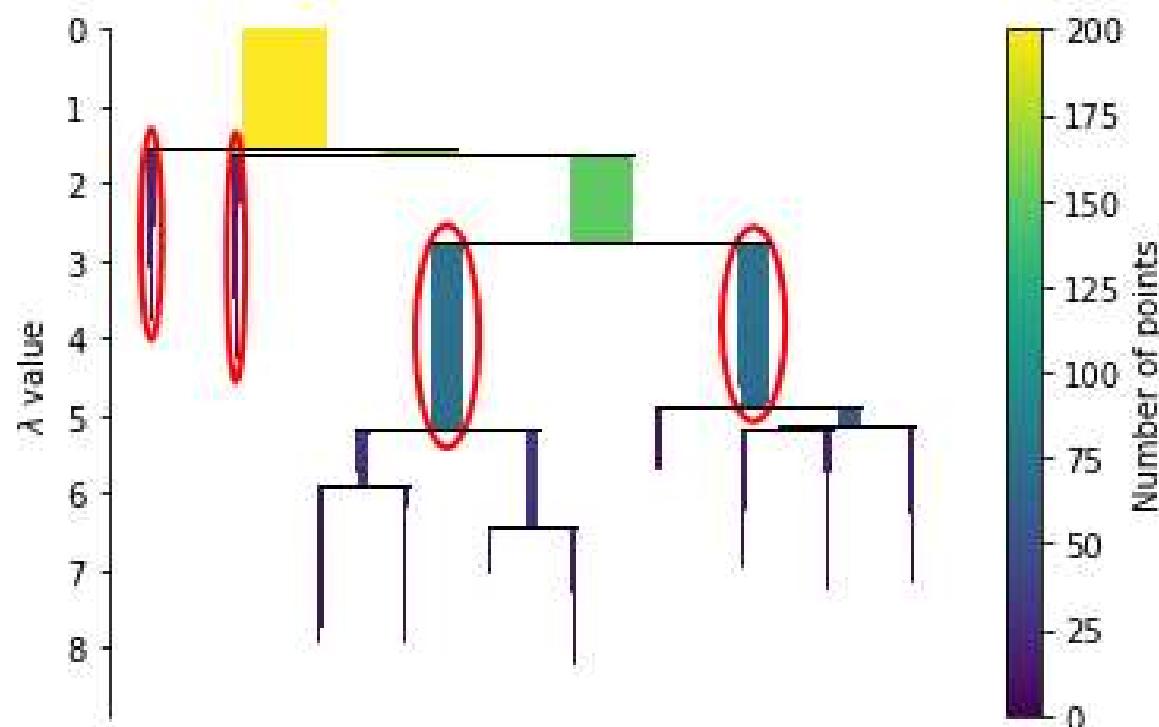
- **Retirer les clusters de petite taille**
  - Paramètre sur la taille des clusters pour retirer ceux qui sont trop petits en parcourant la hiérarchie obtenue dans le dendrogramme
  - Représentation plus condensée



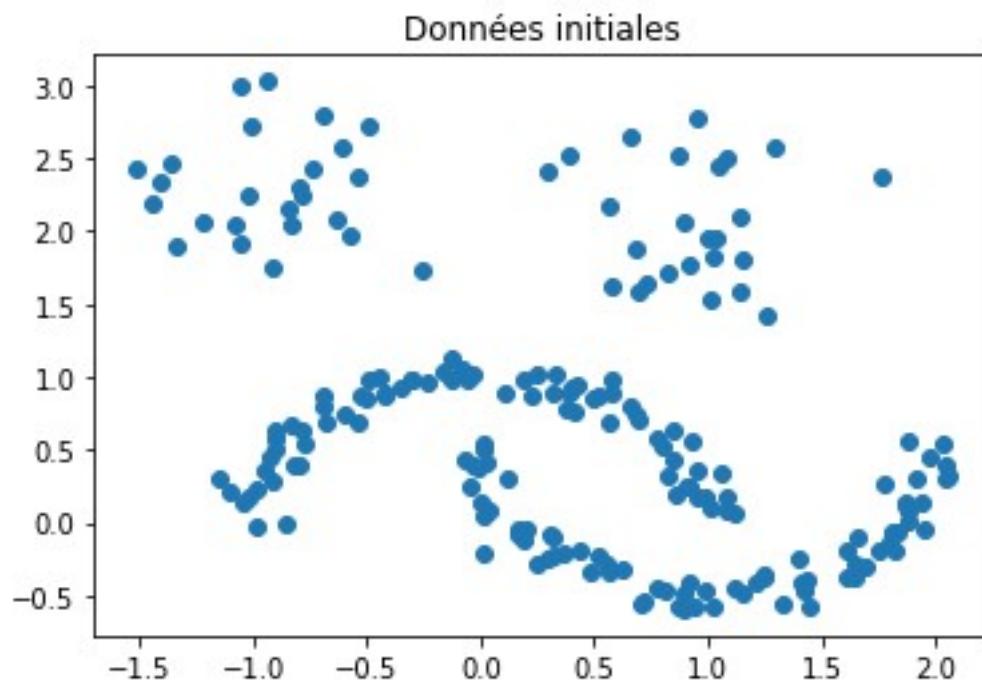
# Autres extensions dans HDBSCAN (2)

- **Extraction des clusters du dendrogramme**
  - Utiliser une métrique de stabilité d'un cluster pour déterminer comment séparer la hiérarchie
    - Densité d'apparition et densité de fusion des clusters

Cluster stable :  
cluster contenant  
des exemples  
présents sur  
plusieurs niveaux  
de la hiérarchie



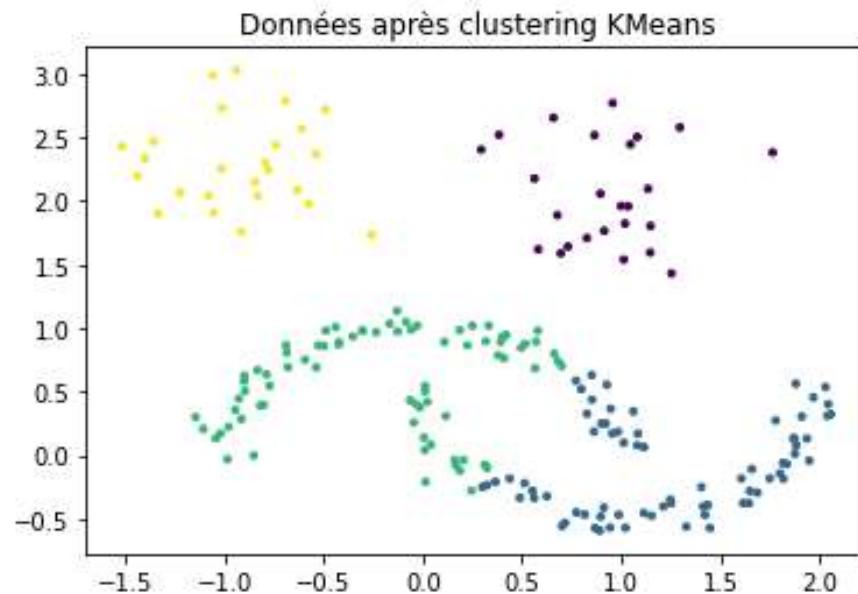
# Exemple : Evaluation clustering



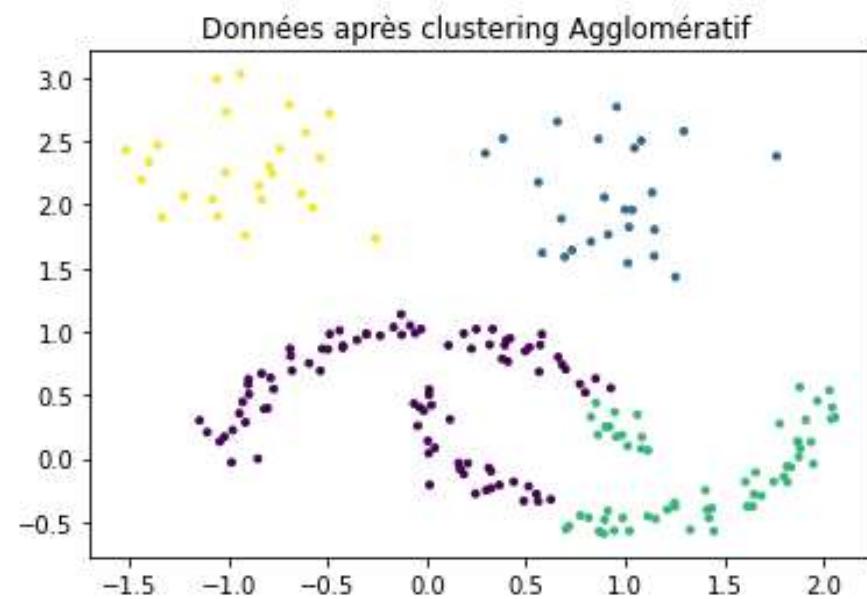
```
np.random.seed(20)
moons, _ = data.make_moons(n_samples=150, noise=0.08)
blobs, _ = data.make_blobs(n_samples=50, centers=[(-0.75,2.25), (1.0, 2.0)],
cluster_std=0.40)
test_data = np.vstack([moons, blobs])
```

# Exemple

---

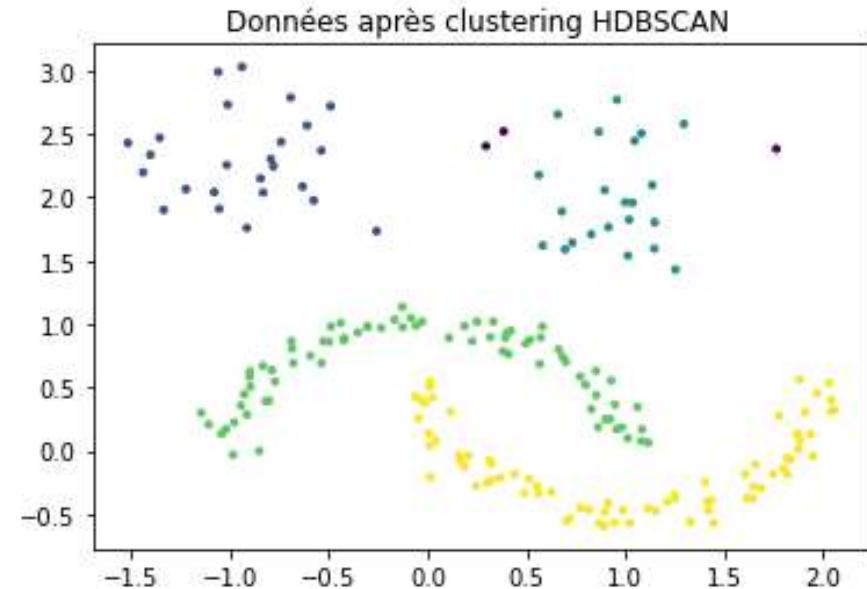
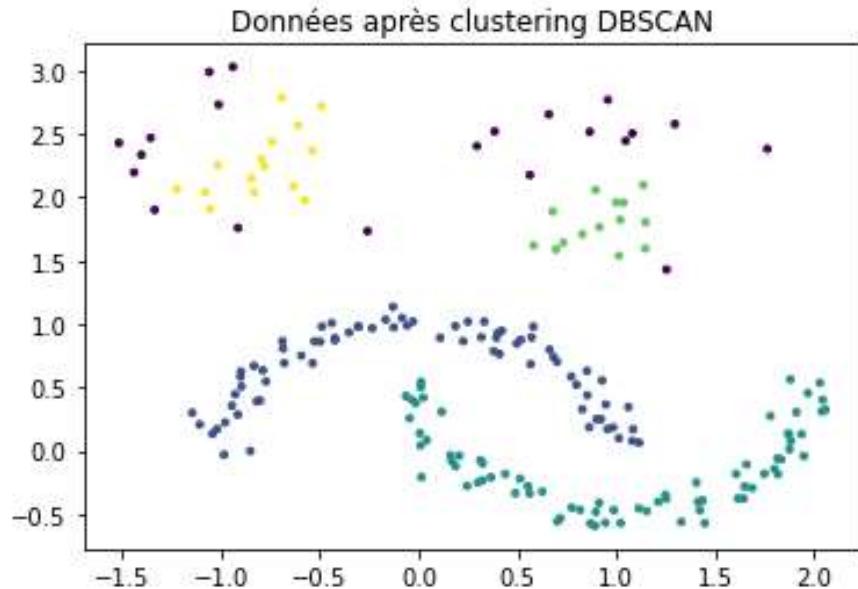


Kmeans (k=4)  
Inertie : 71.930  
Silhouette : 0.507



Agglomératif Ward (k=4)  
Silhouette : 0.486

# Exemple



# Plan

---

## **1. Caractérisation du problème de clustering**

1. Données
2. Distances
3. Problème de partition
4. Synthèse

## **2. Quelques Méthodes**

1. Méthodes basées centres de masses
2. Méthodes hiérarchiques
3. Méthodes basées voisinage (densité)
4. **Méthodes basées graphes**

## **3. Bilan Clustering**

1. Evaluation d'un clustering
2. Application

# Clustering basé sur des graphes (1)

- **Graphe :**

- Représentation de relations entre des entités

- **Graphe de similarité :**

- Sommets : exemples du jeu de données
- Arêtes (ou arcs) : valuées par une mesure de similarité entre 2 sommets
- Graphe complet ou non

- **Différentes métriques :**

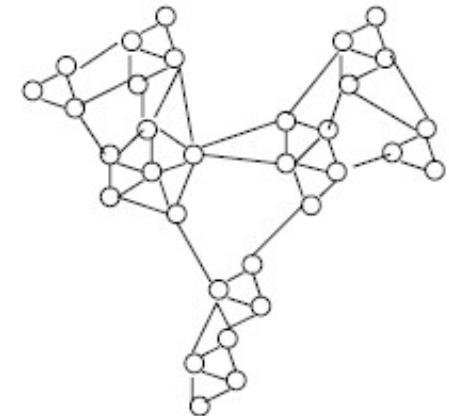
- En lien avec les applications considérées
  - Analyse de réseaux sociaux
    - Mesure de centralité d'intermédiairité (est-ce qu'un sommet participe à la diffusion d'information)

- Basé sur distance

- etc

- $s(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{1+d(\mathbf{x}_i, \mathbf{x}_j)^2}$

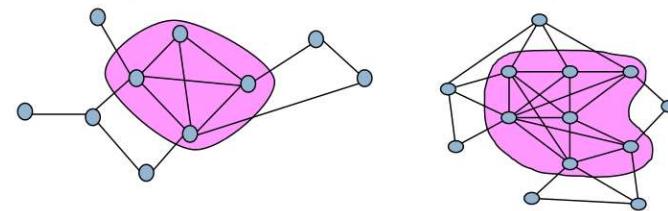
- $s(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{d(\mathbf{x}_i, \mathbf{x}_j)^2}{2\sigma^2}\right)$



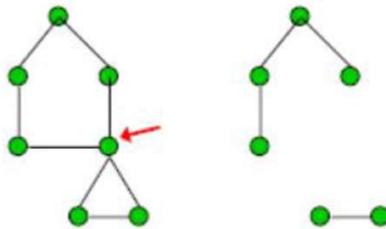
# Clustering basé sur des graphes (2)

## • Structures spécifiques dans des graphes

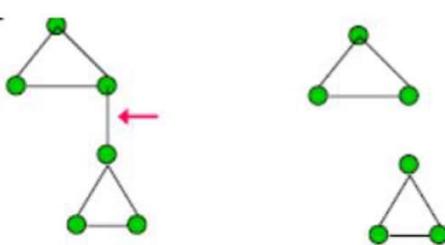
- **Composantes connexes** : ensemble maximal de sommets tels qu'il existe un chemin entre chaque paire
- **Composantes k-connexes** : ensemble maximal de sommets tels qu'il existe k chemins disjoints entre chaque paire
- **Cliques** : sous-graphe complet



- **Point de coupure** :

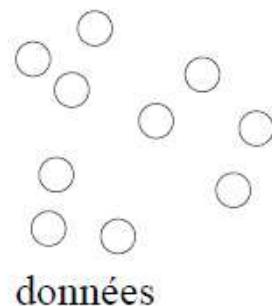


- **Pont (arête)** :

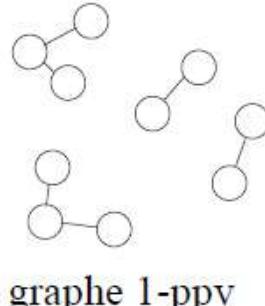


# Graphe de similarité (1)

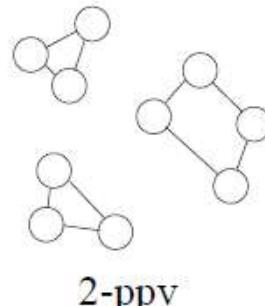
- **Graphe  $\epsilon$  Voisinage**
  - Sommet : exemple du jeu de données
  - Arête entre un sommet sommet  $p_i$  et un sommet  $p_j$  si  $p_j$  appartient au  $\epsilon$  voisinage de  $p_i$
- **Graphe des k plus proches voisins**
  - Sommet du graphe : exemple du jeu de données
  - Arête entre un sommet  $p_i$  et un sommet  $p_j$  si  $p_j$  appartient au k-voisins de  $p_i$



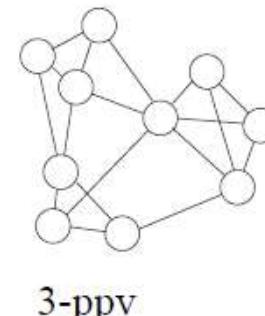
données



graphe 1-ppv



2-ppv

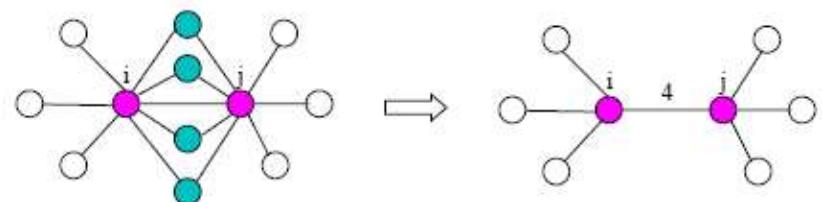


3-ppv

# Graphe de similarité (2)

- **Graphe des k plus proches voisins avec valuation**
  - Sommet : exemple du jeu de données
  - Arête entre un sommet  $p_i$  et un sommet  $p_j$  si  $p_j$  appartient au k-voisins de  $p_i$ 
    - L'arête est valuée par le nombre de voisins en commun

- **Graphe Shared NN valué**



- **Graphe d'accessibilité mutuelle**
  - Cf. méthode HDBSCAN

# Partitionnement de graphes (1)

## • Plusieurs problèmes

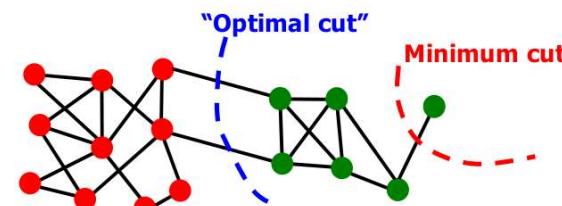
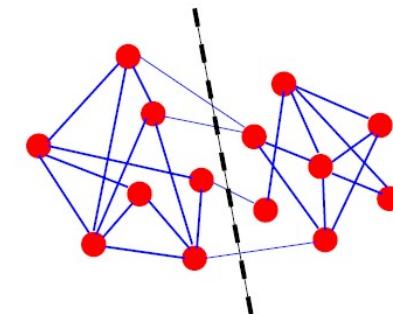
- Diviser un graphe en  $k$  groupes tel que le nombre (la valeur) de liens entre les groupes soit minimal
- Nombre de liens entre groupes : taille de la coupe
  - Minimiser la coupe (Min Cut)
    - Coupe : somme des similarités des arêtes entre chaque cluster

$$cut(P) = \frac{1}{2} \sum_{c=1}^k cut(C_c)$$

- Problème polynomial pour 2 clusters (Max Flow)

- Sinon NP-difficile

- Risque de partition déséquilibrée
    - sommet dans un cluster



# Partitionnement de graphes (2)

## • Ajout de contraintes

- Nombre de groupes
  - Sinon solution triviale pour minimiser la coupe : tous les sommets dans le même groupe
- Taille des groupes
  - Sinon solution triviale : séparer le sommet de plus petit degré

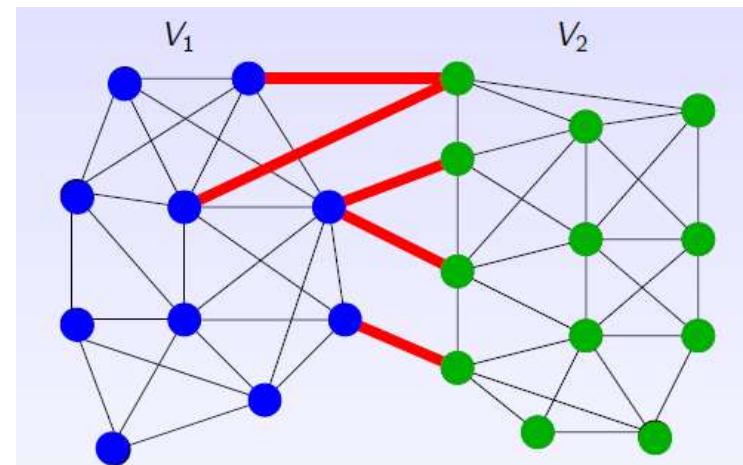
## • Equilibrer la taille des partitions $P_k = \{S_1, \dots, S_k\}$

- Poids moyen :  $P_{moy} = \frac{\sum_{S_i} poids(S_i)}{k}$
- Equilibrage :  $bal(P_k) = \frac{\max_i poids(S_i)}{poids_{moy}}$

## • Trouver une partition telle que

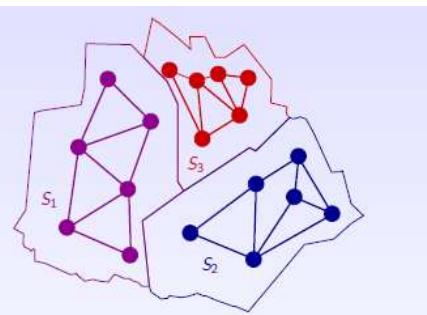
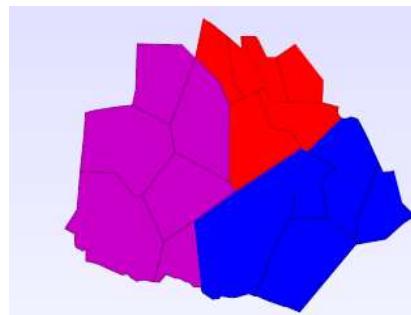
- Equilibrage  $\leq 1,05$  (taille similaire)
- Cout de coupe minimisée
  - $cut(P_k) = \sum_{i < j} cut(S_i, S_j)$
  - Avec

- $cut(S_i, S_j) = \sum_{x_i \in S_i, x_j \in S_j} poids(x_i, x_j)$



# Partitionnement de graphes (3)

- **Sans contrainte sur la taille**
  - Intégrer taille ou volume dans la fonction objectif
  - Exemple : Minimiser une fonction objectif intégrant la différence de taille entre les groupes
  - Coupe normalisée
    - $ncut(P_k) = \sum_{i < i} ncut(S_i, S_j)$
    - $ncut(S_i, S_j) = \frac{cut(S_i, S_j)}{nb\_ext(S_i)} + \frac{cut(S_i, S_j)}{nb\_ext(S_j)}$ 
      - où  $nb_{ext}(X)$  le nombre de liens ayant une extrémité dans  $X$
  - Application : partitionnement espace aérien (thèse Bichot 2012)

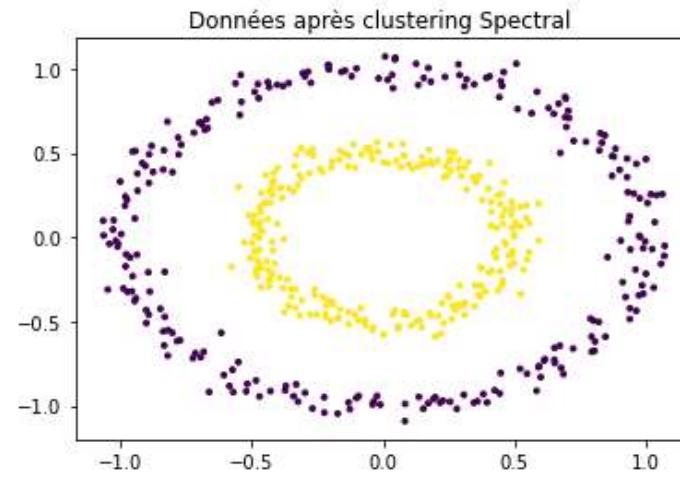
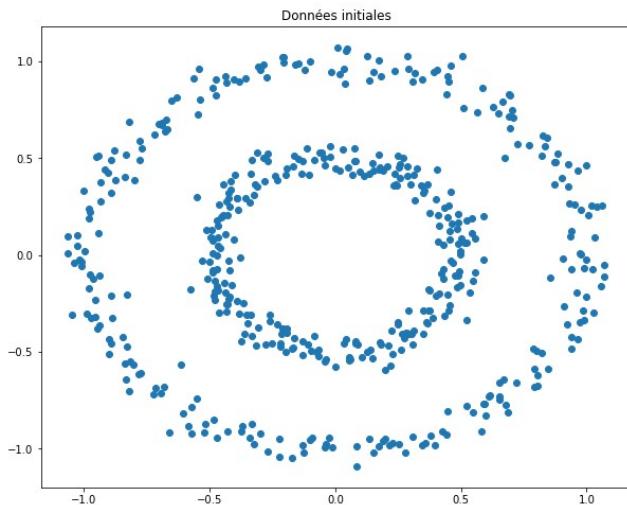


# Méthodes de partitionnement de graphes

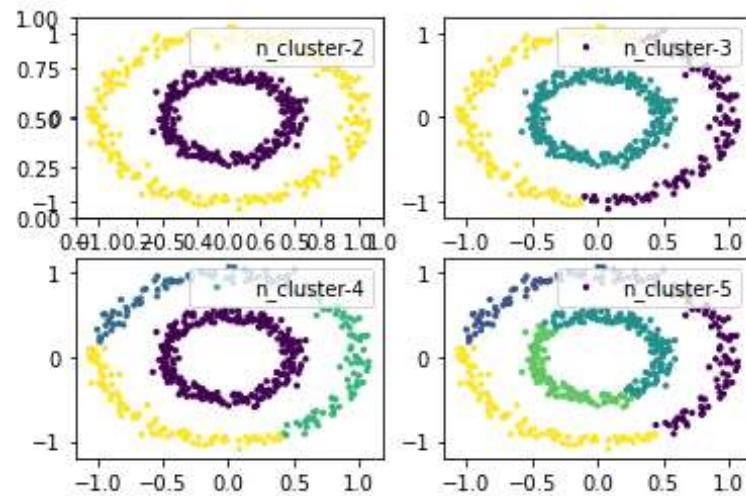
---

- **Problèmes NP-difficile**
  - Sauf cas particulier (Flot Max – Coupe Min)
- **Differentes familles de méthodes**
  - Heuristique Kernighan-Lin (bi-partition)
  - Méthodes de clustering type k-medoids, agglomératif, ...
  - Méthode spécifique : clustering spectral
    - Proposée dans scikit-learn
    - Résolution approchée du problème de coupe dans un graphe basée sur des résultats d'algèbre linéaire (transformation de la matrice d'adjacence du graphe, calcul de valeurs propres, projection du graphe, recherche de clusters dans le graphe projeté)
  - Prise en compte de formes quelconques
  - Nécessite de fixer le nombre de clusters

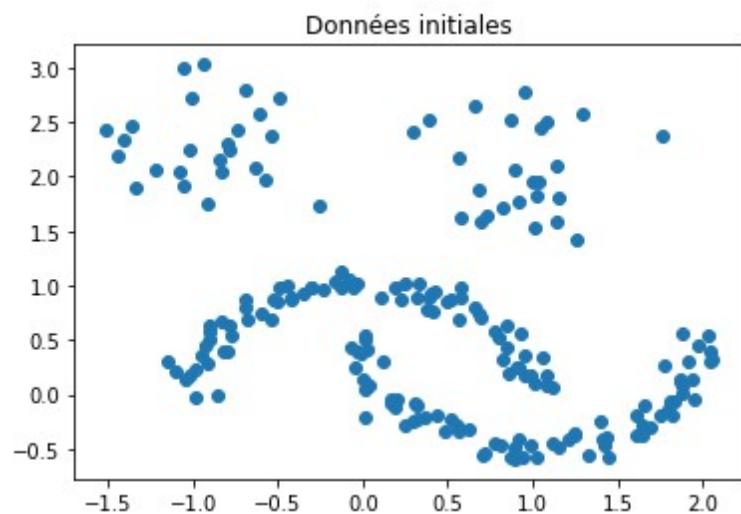
# Exemple



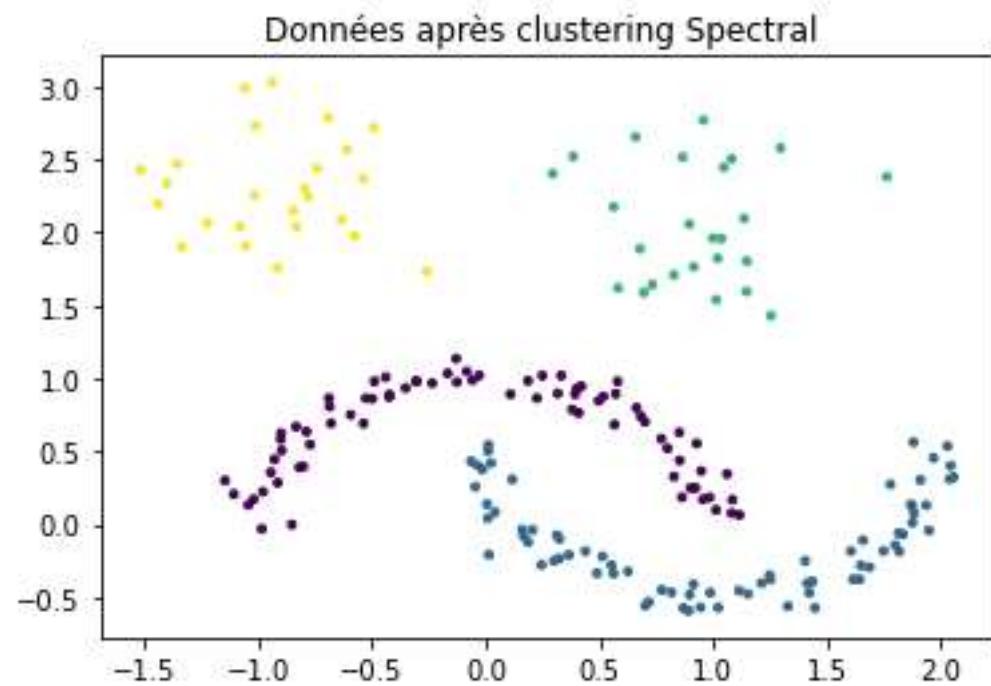
```
SpectralClustering(n_clusters=2, affinity='nearest_neighbors')
```



# Exemple



Ne détecte pas les anomalies



Spectral Clustering  
Number of clusters: 4  
Silhouette : 0.380

# Plan

---

## **1. Caractérisation du problème de clustering**

1. Données
2. Distances
3. Problème de partition
4. Synthèse

## **2. Quelques Méthodes**

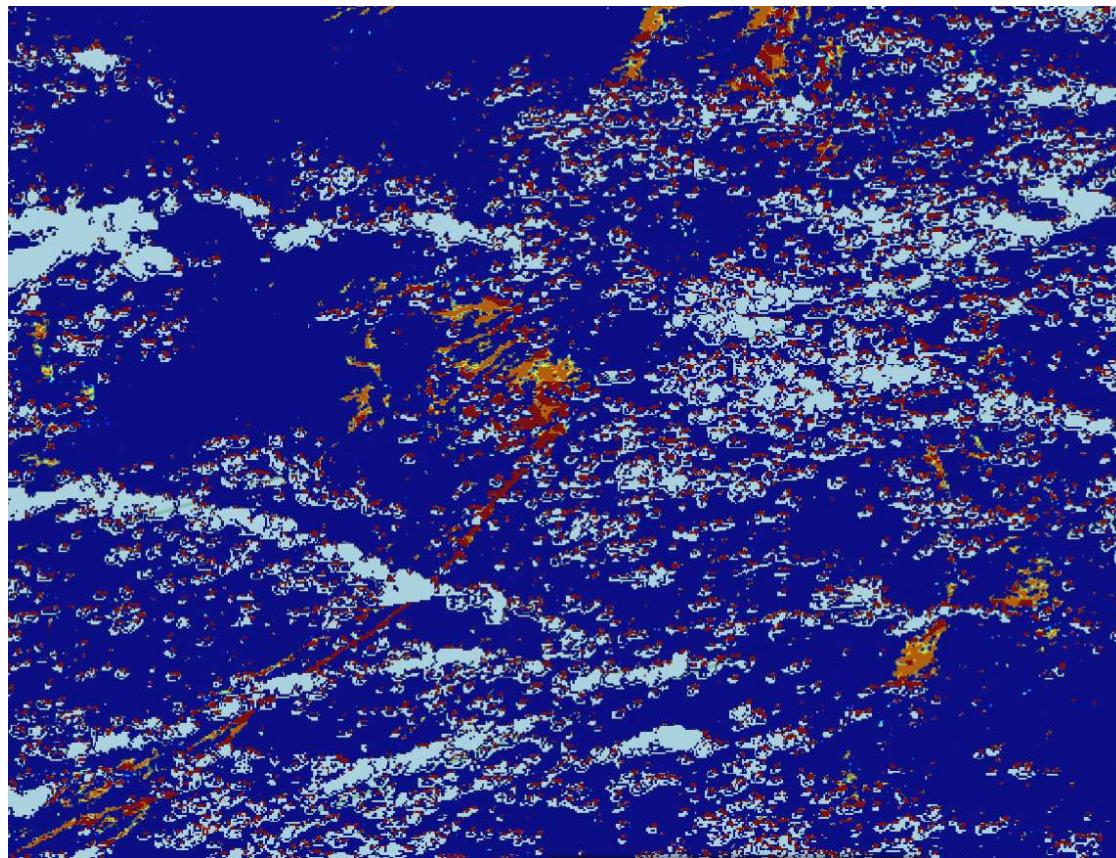
1. Méthodes basées centres de masses
2. Méthodes hiérarchiques
3. Méthodes basées voisinage (densité)
4. Méthodes basées graphes

## **3. Bilan Clustering**

1. Evaluation d'un clustering
2. Application

# Application (1)

- **Identification dans des images**
  - Algues sargasses (travail avec CLS/CNES)



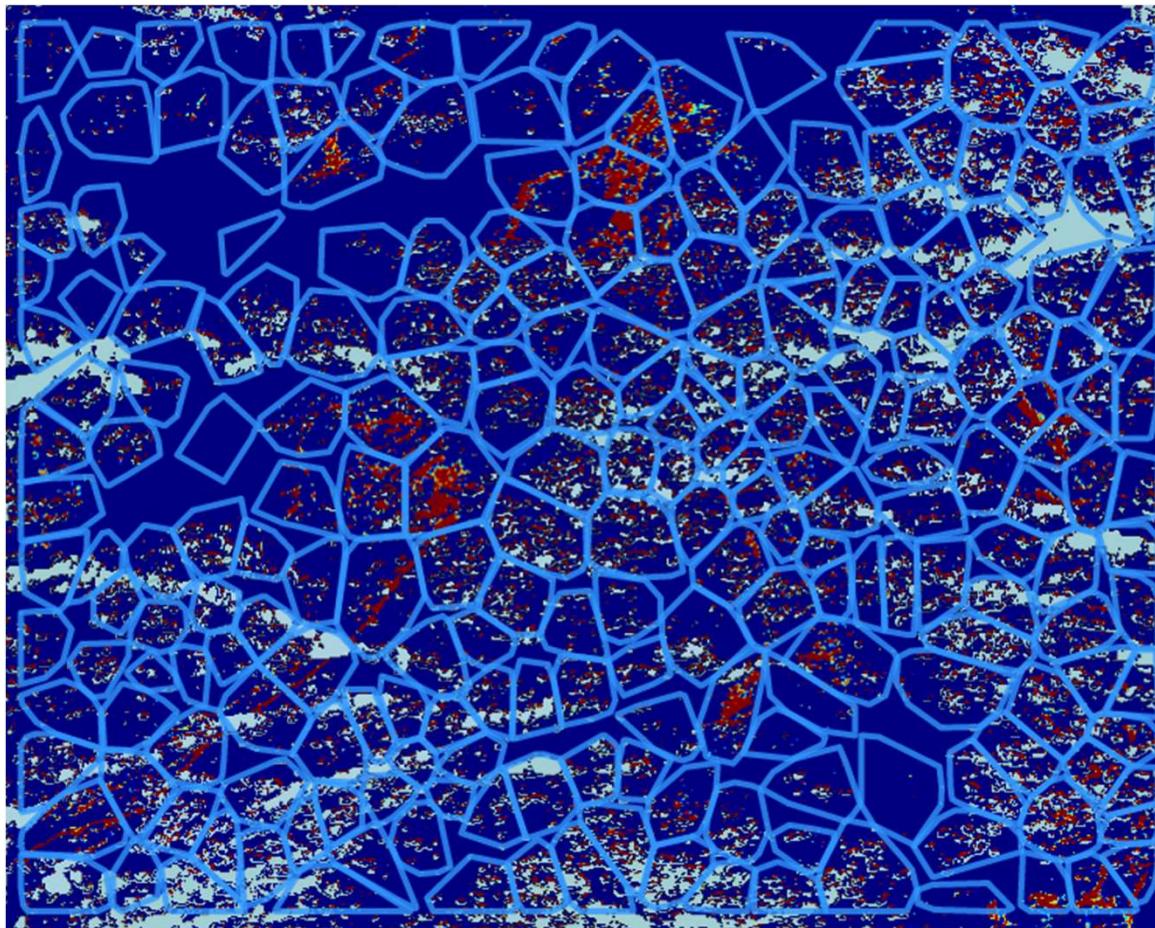
Carte 1

Différentes valeurs sur  
chaque pixel

- jaune = NFAI,
- rouge = Raw NFAI,
- bleu = mer et
- blanc = nuage

# Application (2)

- **Application Mean-Shift (→)**



Carte 1

Pas de détection bruit  
Nombre de clusters très important

# Méthode Mean-Shift (1)

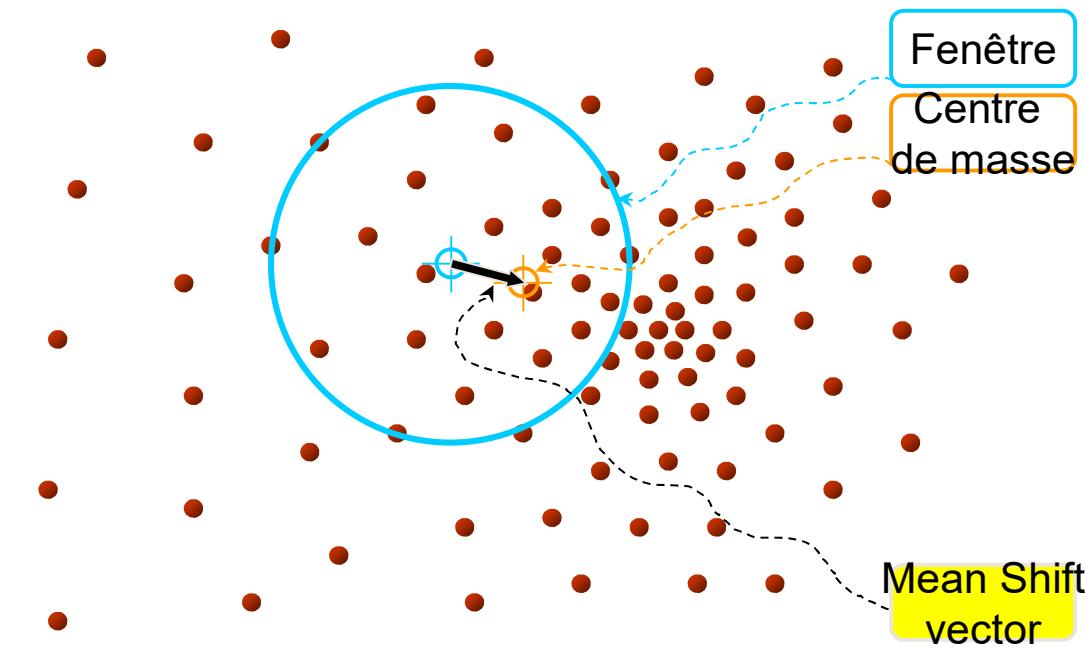
---

- **Principe**
  - Le nombre de clusters n'est pas fixé
  - Localiser des centres en fonction des densités locales
    - Utilisation de centres aléatoires initiaux
    - Exploration d'une « zone de visibilité » autour de chaque centre
    - Décalage du centre vers des régions dont la densité moyenne est plus élevée
  - Arrêt : le nombre d'exemples dans une fenêtre est stabilisé
    - Centre de densité moyenne locale la plus élevée
  - Processus sur plusieurs centres initiaux
    - Plusieurs centres initiaux peuvent converger au même endroit

# Méthode Mean-Shift (2)

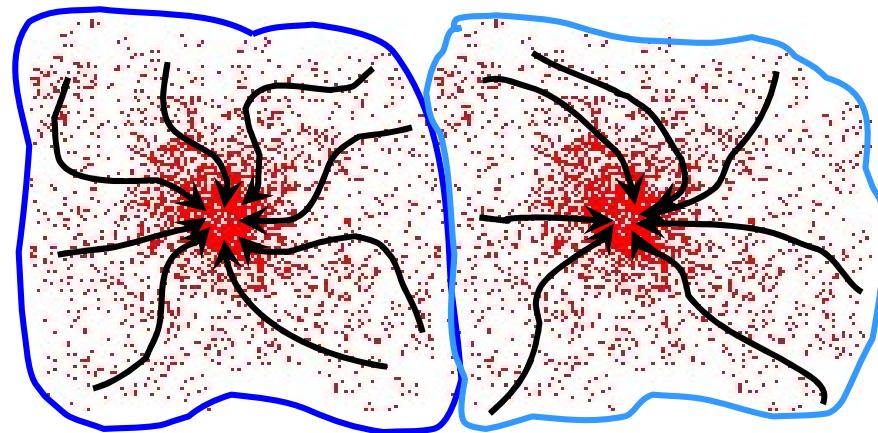
## • Algorithme

1. Initialisation : centres aléatoires et fenêtre W
2. Calculer les centre de gravité (« mean ») de W
3. Décaler la fenêtre vers la moyenne
4. Retour en 2 jusqu'à convergence



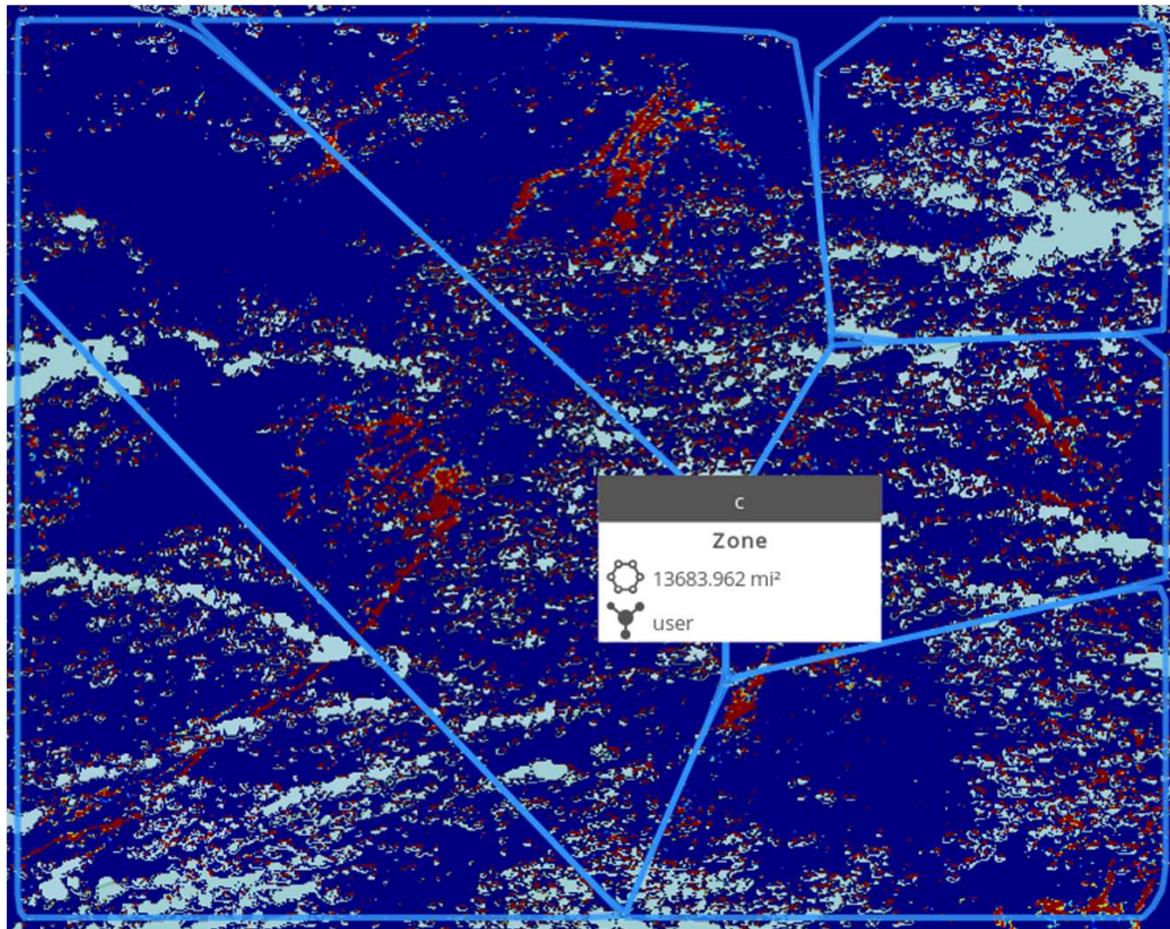
# Méthode Mean-Shift (3)

- **Ensemble de centres initiaux**
  - Obtention de bassins d'attraction



# Application (3)

- **Application Mean-Shift**

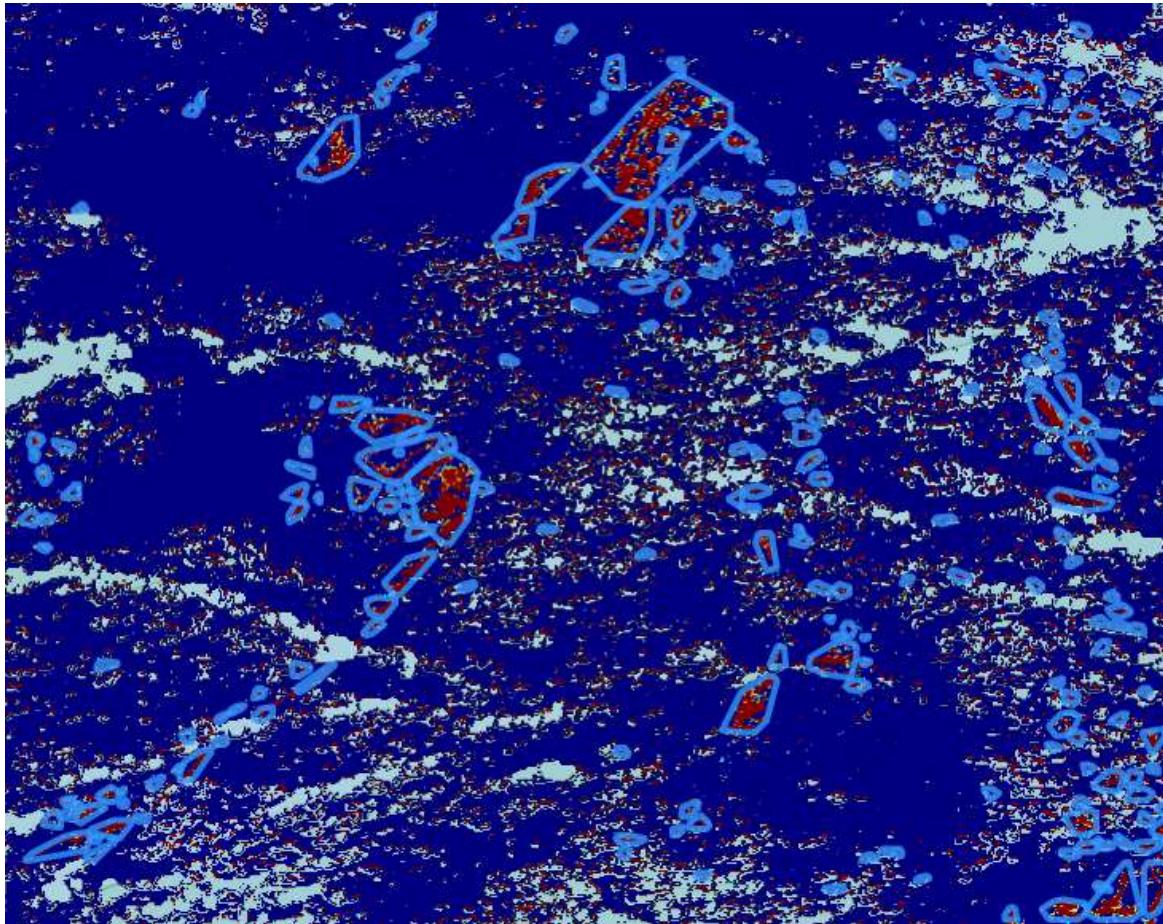


Carte 1

Pas de détection bruit  
Réduction du nombre de clusters  
Peu pertinent / bancs d'algues

# Application (4)

- **Application DBSCAN**

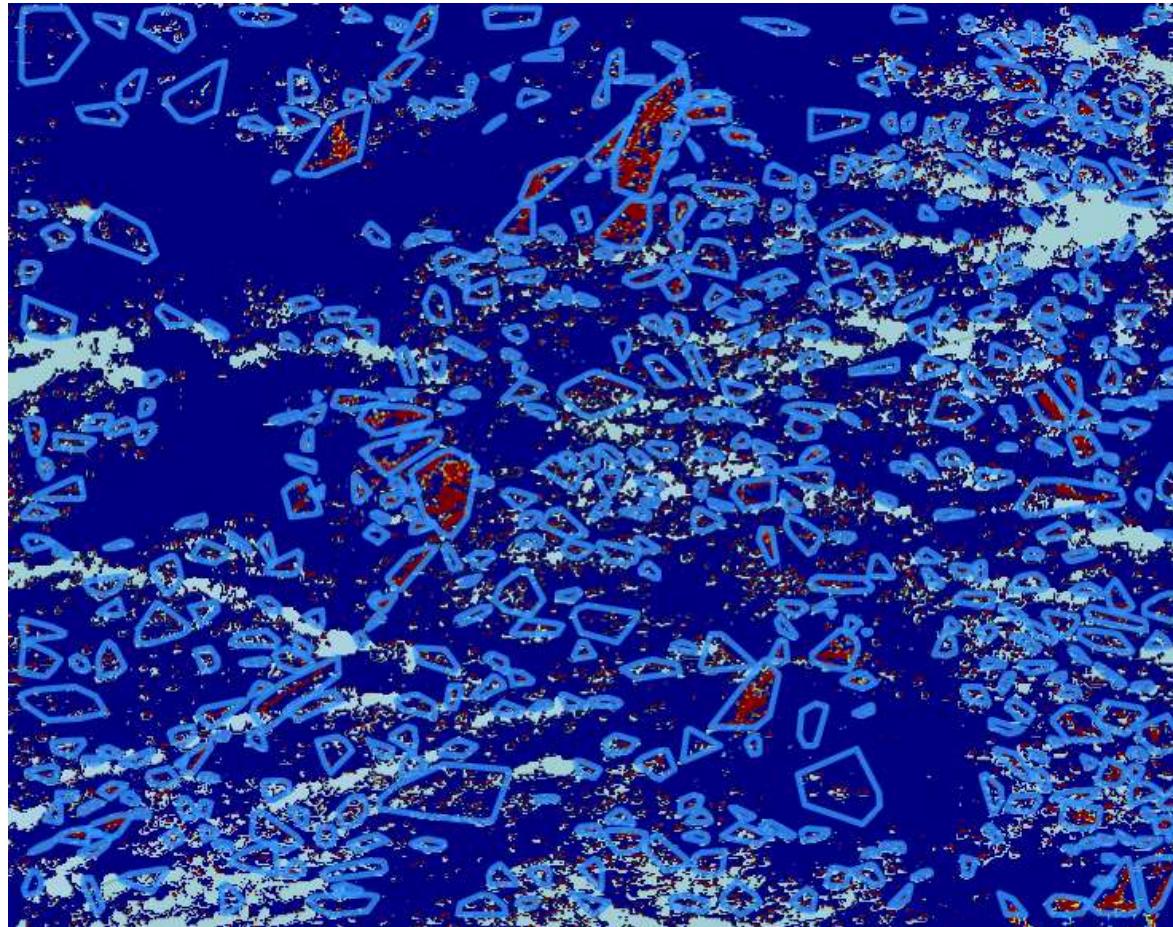


Carte 1

Détection bruit  
Nombre de clusters limité  
Sensibilité à la densité  
Plus pertinent sur les  
bancs d'algues

# Application (5)

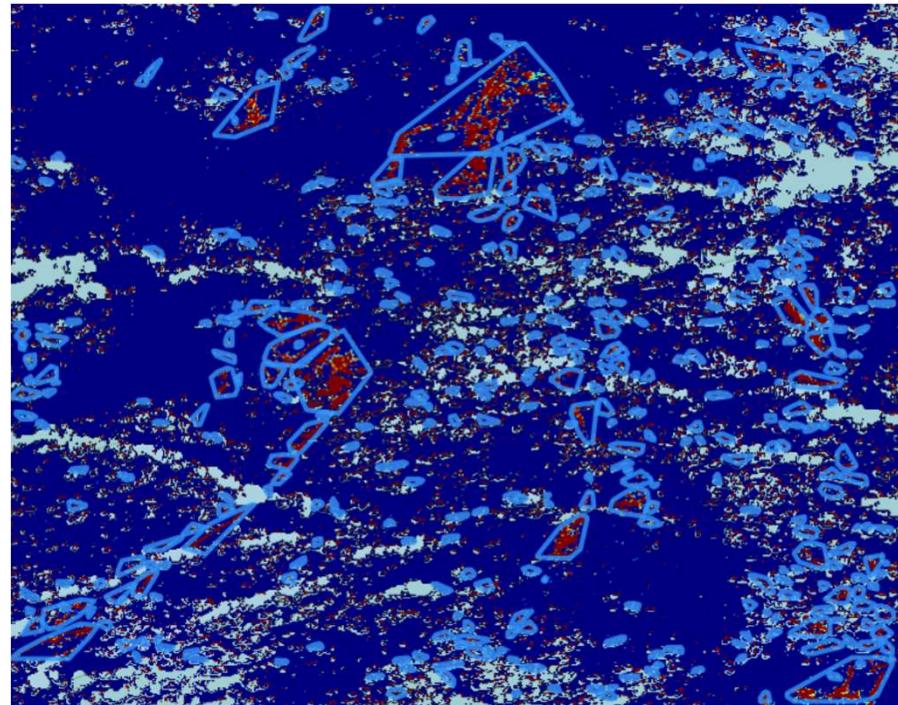
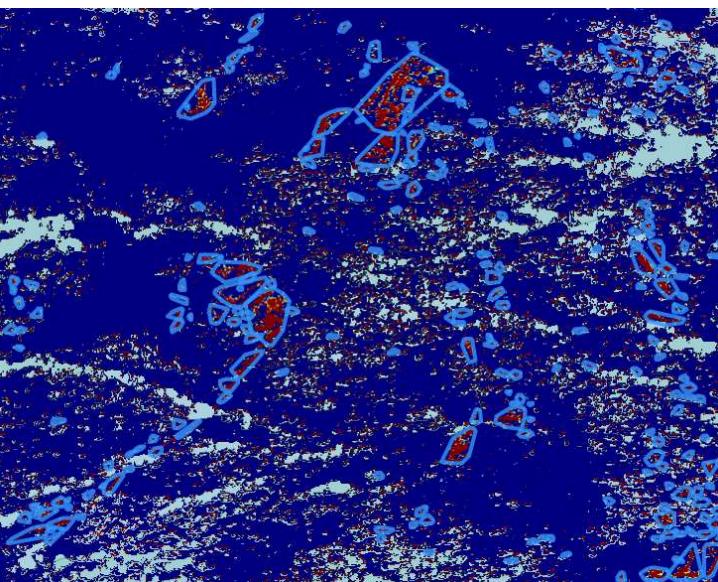
- **Application HDBSCAN**



Carte 1

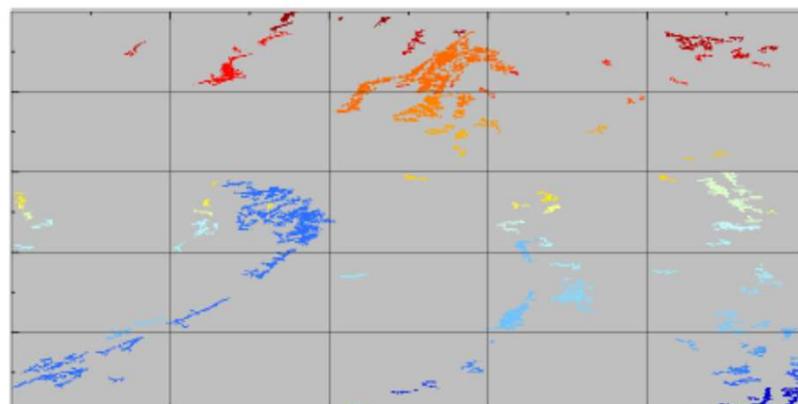
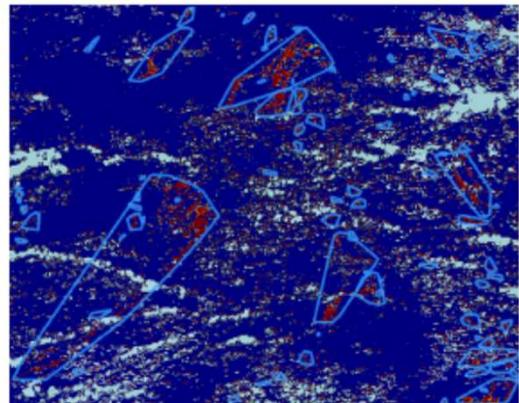
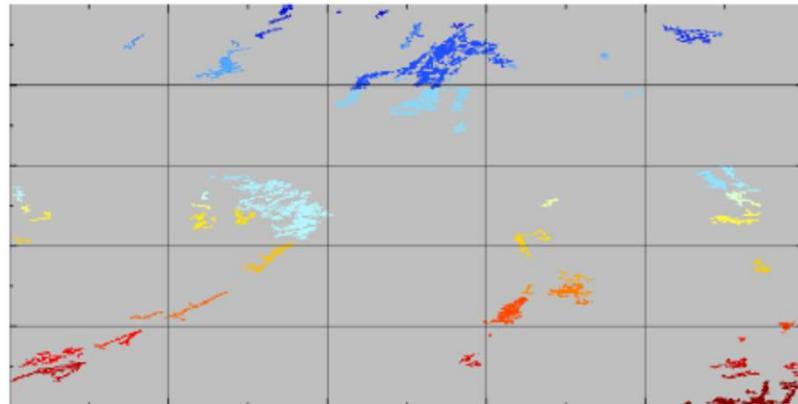
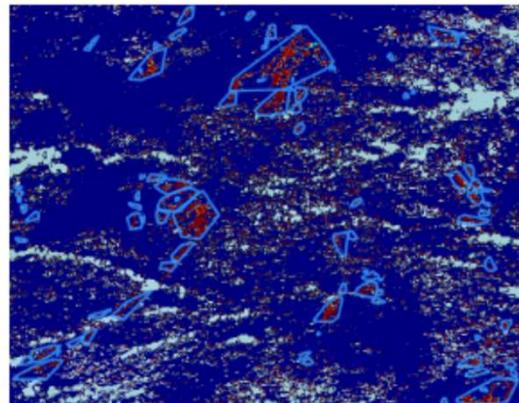
Peu sensible à la densité  
Hausse importante du  
nombre de clusters  
Moins pertinent sur les  
bancs d'algues

Au final ..



# Au final ..

- 
- **Autre visualisation**
    - Plusieurs postprocessing



# Au final ..

---

- **Qualification d'un clustering**
  - Forme
  - Comparaison relative de méthodes
  - Avis expert

# Bilan - Clustering

---

- **Objectif du Clustering**
  - Regrouper (séparer) des données en fonction de similitudes (différences)
  - Donner du sens aux données
  - Identifier des anomalies
- **Difficultés**
  - Formalisation du problème
    - Caractériser similitudes
    - Déterminer une fonction « objectif »
      - Maximiser la similarité intra-cluster
      - Minimiser la similarité inter-cluster
  - Nombre très élevé de partitions possibles d'un jeu de données
  - Qualité d'une solution de clustering

# Bilan - Clustering

---

- **Très nombreuses méthodes**
  - Méthodes génériques ou issues d'un domaine d'application
  - Familles de méthodes
    - centres
    - connectivité (hiérarchique)
    - densité
    - graphes
    - ...
- Focus sur quelques méthodes approchées
  - Heuristique, exploration de voisnages, approximation d'un problème d'optimisation, ...
- peu de méthodes exactes
  - Graphes, Programmation Linéaire en nombre entiers, Programmation par Contraintes

# Extensions

---

- **Soft Clustering**

- Un exemple peut appartenir à plusieurs clusters
  - Pour chaque exemple : déterminer un score ou degré d'appartenance aux différents clusters
  - Adaptation k-means : fuzzy c-means (année 70-80)
    - Minimiser l'inertie pondérée
$$\sum_{i=1}^n \sum_{j=1}^c w_{ij}^m \| \mathbf{x}_i - \mathbf{c}_j \|^2$$
  - Applications : bioinformatique, marketing, traitement d'images, ...

# Extensions

## • Clustering avec recouvrement

- Exemple

- Find a set of (at most  $K$ ) overlapping clusters:

$$\{S_1, \dots, S_t\}$$

each cluster with volume  $\leq B$ ,

covering all nodes,

and minimize:

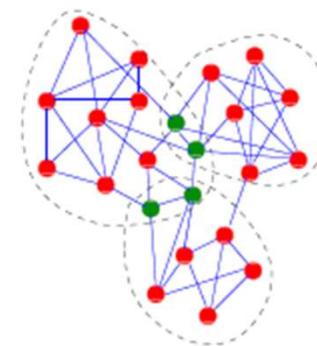
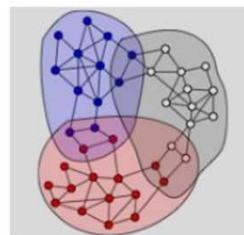
- Maximum conductance of clusters (Min-Max)
- Sum of the conductance of clusters (Min-Sum)

$$\text{Conductance of a cluster } S = \frac{\#\text{cut edges}}{\min(\text{vol}(S), \text{vol}(\bar{S}))}$$

- Inside: Well-connected,  
– Toward outside: Not so well-connected.

- Applications : réseaux sociaux, recommandation, fouille de textes - Années 2000

Rohit Khandekar, Guy Kortsarz,  
and Vahab Mirrokni



## 20 years of network community detection

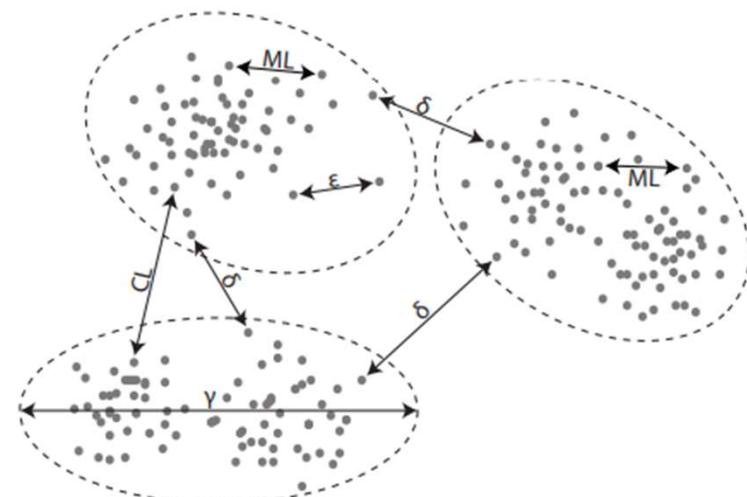
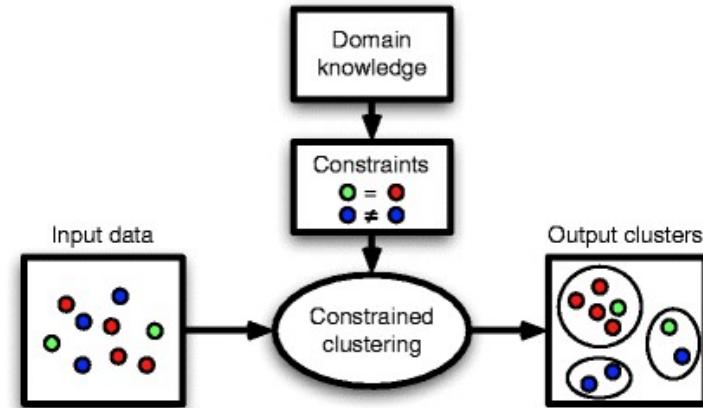
Santo Fortunato & Mark E. J. Newman

[Nature Physics](#) 18, 848–850 (2022) | [Cite this article](#)

# Extensions

## • Clustering sous contraintes

- Intégrer des connaissances pour réaliser un clustering
  - Années 2000
- Différents types de contraintes
  - Sur les clusters :
    - nombre,
    - taille,
    - Diamètre  $\gamma$ ,
    - Densité  $\epsilon$
    - Distance de séparation  $\delta$
  - Sur les exemples : must-link (ML) et cannot-link (CL)



**Constrained  
Clustering**

Advances in Algorithms,  
Theory, and Applications