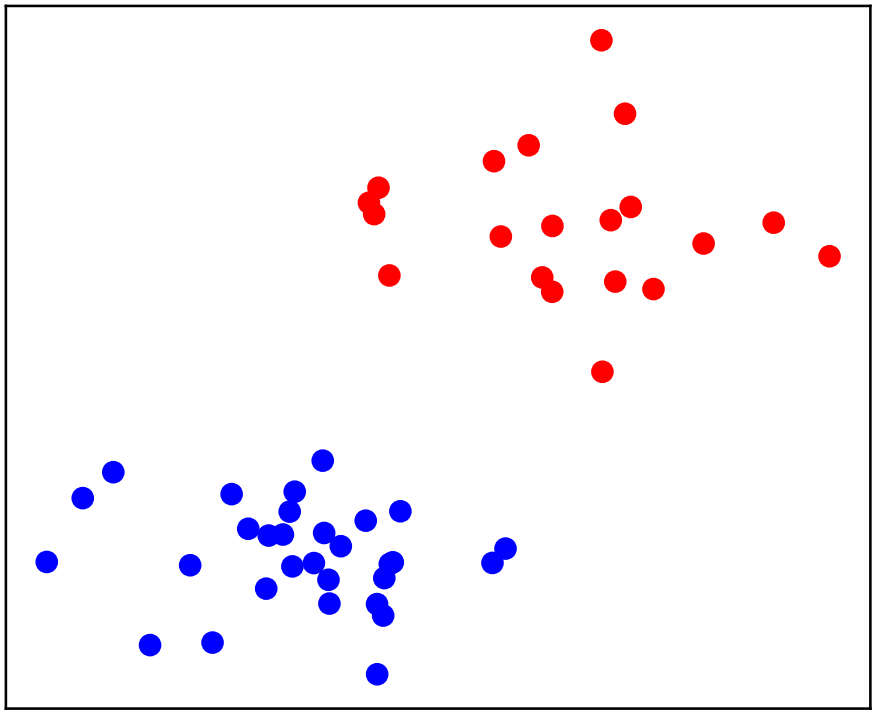
Algorithme des k-Moyennes

Charlotte Pelletier

05 février2020

Plan

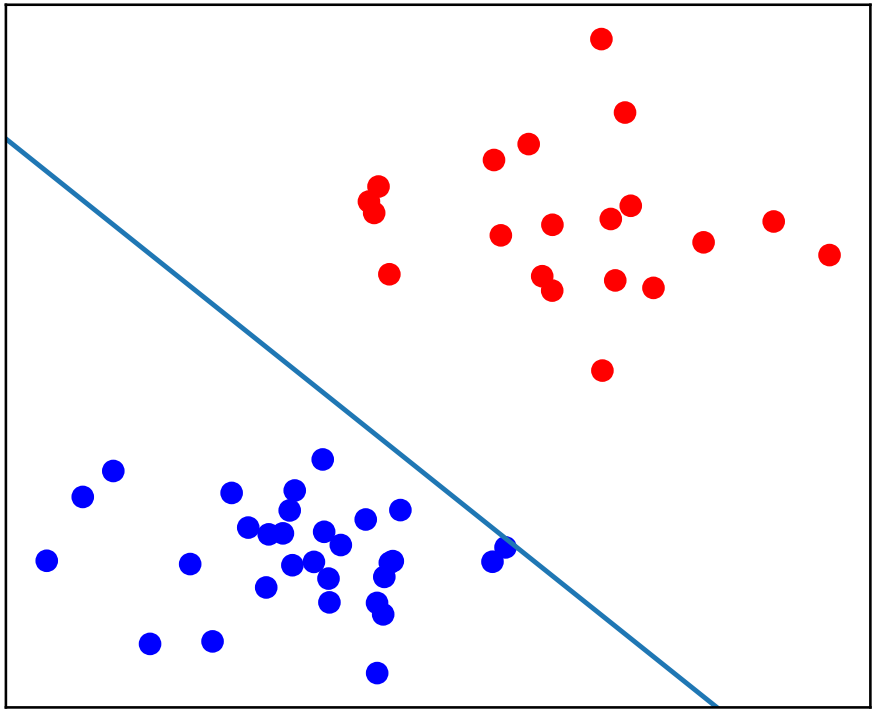
Apprentissage supervisé



Données d'apprentissage $\{\mathbf{x}_i, y_i\}_{i=1}^m$

- observations : $\mathbf{x}_i \in \mathbb{R}^d$ de dimension d
- valeurs à prédire (classe) : $y_i \in \mathcal{Y}$

Apprentissage supervisé

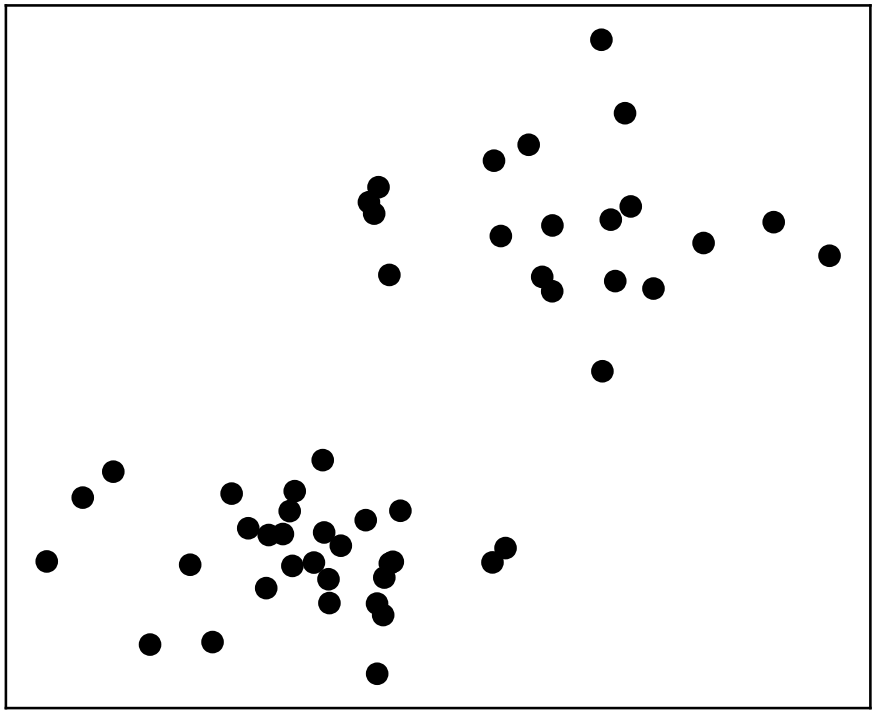


Données d'apprentissage $\{\mathbf{x}_i, y_i\}_{i=1}^m$

- observations : $\mathbf{x}_i \in \mathbb{R}^d$ de dimension d
- valeurs à prédire (classe) : $y_i \in \mathcal{Y}$

Objectif : trouver une fonction pour prédire la classe (●/●) de nouvelles observations.

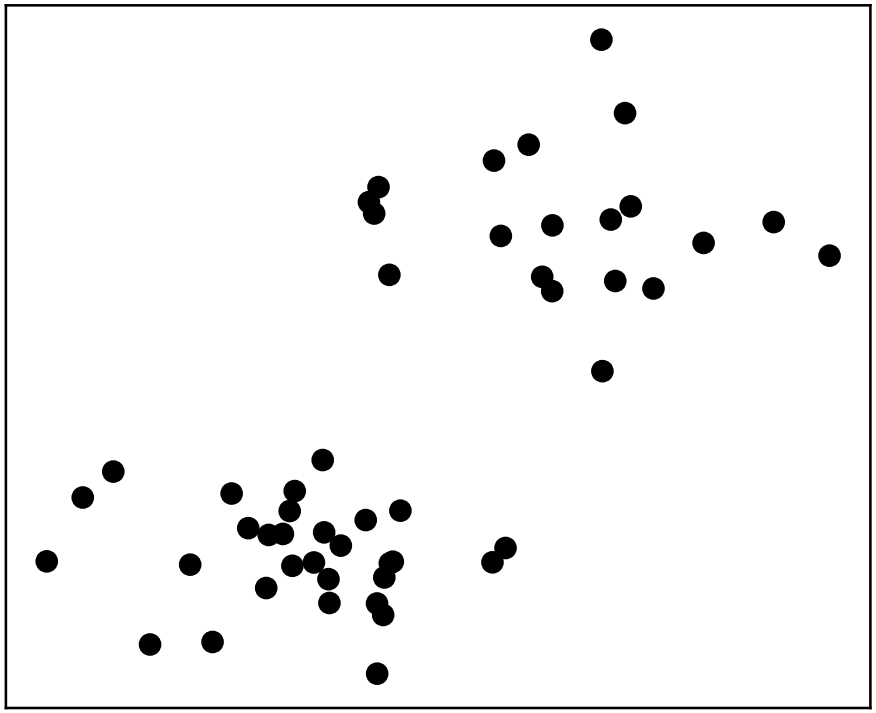
Apprentissage non-supervisé



Données d'apprentissage $\{\mathbf{x}_i\}_{i=1}^m$

- observations : $\mathbf{x}_i \in \mathbb{R}^d$ de dimension d

Apprentissage non-supervisé



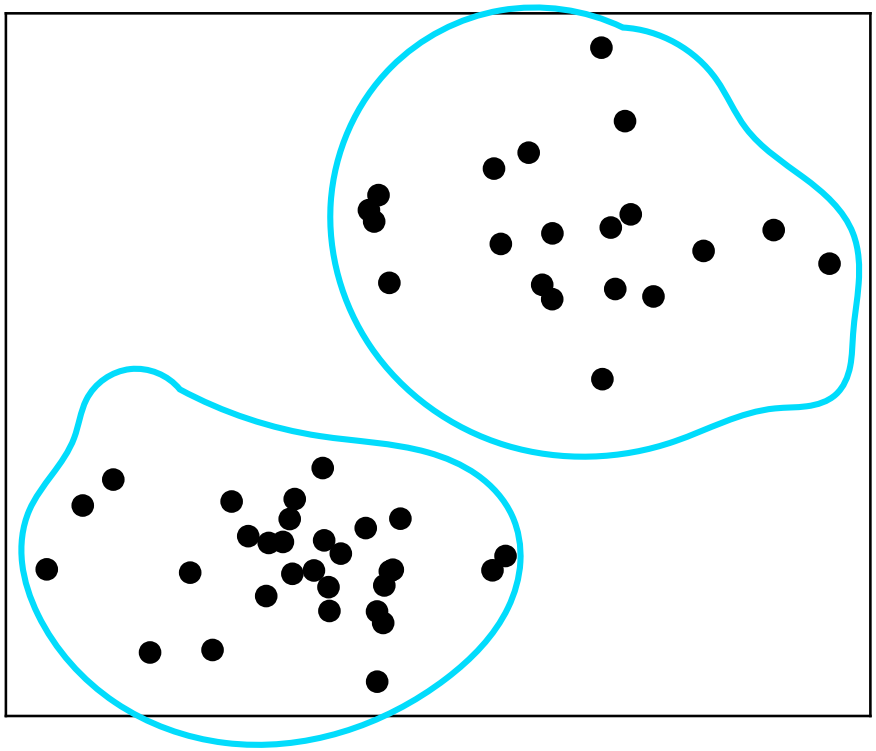
Données d'apprentissage $\{\mathbf{x}_i\}_{i=1}^m$

- observations : $\mathbf{x}_i \in \mathbb{R}^d$ de dimension d

Applications

- Estimation de densité de probabilité
- Réduction de dimension
- Clustering

Apprentissage non-supervisé



Données d'apprentissage $\{\mathbf{x}_i\}_{i=1}^m$

- observations : $\mathbf{x}_i \in \mathbb{R}^d$ de dimension d

Applications

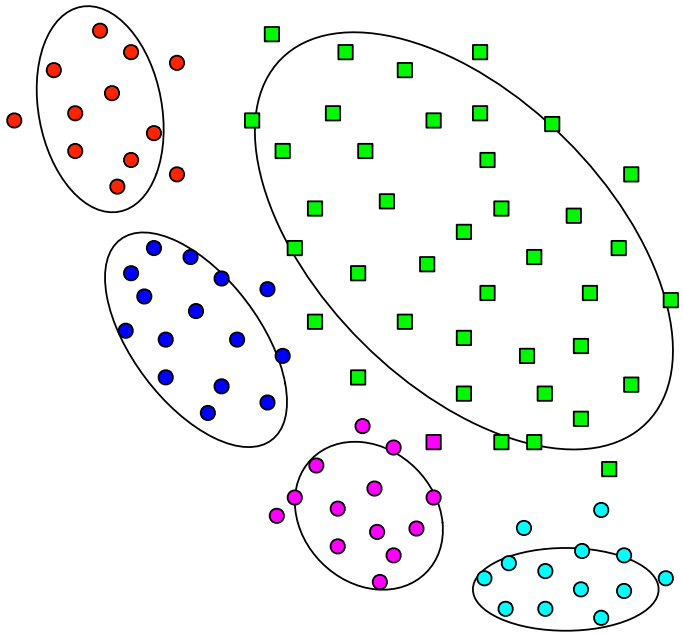
- **Clustering**

Objectif : grouper les observations qui se ressemblent.

Clustering [Rappel - Introduction]

Objectif

- Organiser les exemples d'apprentissage par groupes.
- $\{\mathbf{x}_i\}_{i=1}^m \Rightarrow \{\hat{y}_i\}_{i=1}^m$ où $\hat{y} \in \mathcal{Y}$ représente un groupe (un cluster) $\{1, \ldots, k\}$
- Paramètres :
  - k nombre de groupes
  - mesure de similarité (caractériser les similarités entre les observations)



Méthodes

- k-Means (k-Moyennes).
- Mélange de gaussiennes
- Clustering hiérarchique

Exemples

- Taxonomie d'animaux
- Regroupement de gènes
- Réseaux sociaux

Hypothèses

- il existe une structure sur les données
- chaque observation $\mathbf{x}_i$ est utilisée pour définir la structure

Objectifs

- Rechercher une typologie, une segmentation, un clustering des données
- Constituer des groupes homogènes et différenciés, *i.e.*,
  - dans un même groupe les individus doivent se ressembler le plus (critère de compacité)
  - dans deux groupes différents les individus doivent être aussi dissemblables que possible (critère de séparabilité)

Applications

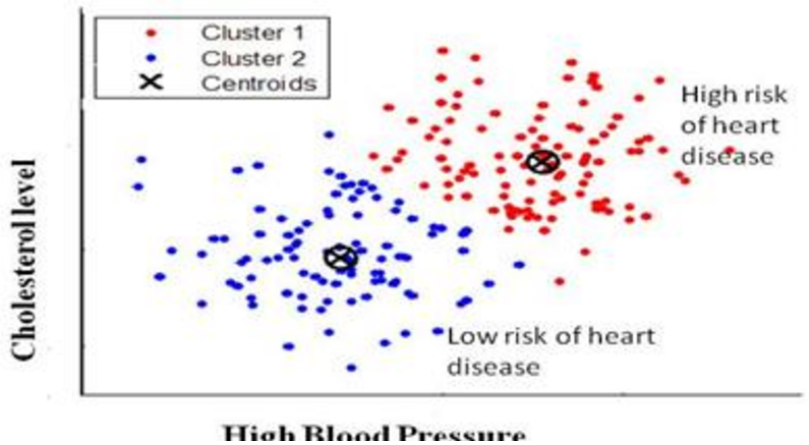
Réseaux sociaux

- recommandation
- compréhension de contenu

Marketting

- analyse du comportement de clients
- regroupement des clients similaires

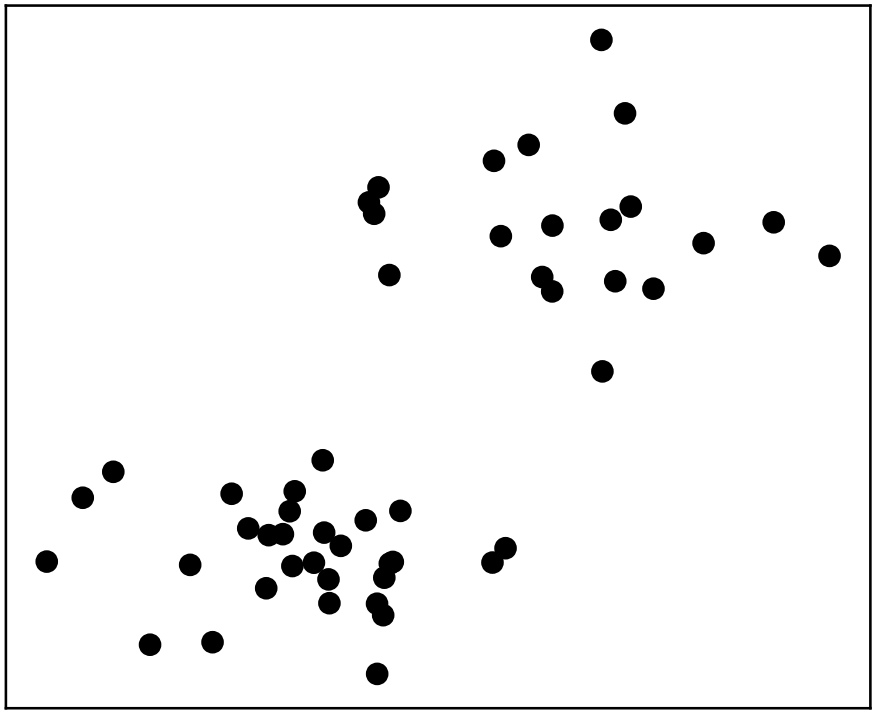
Bio-médical



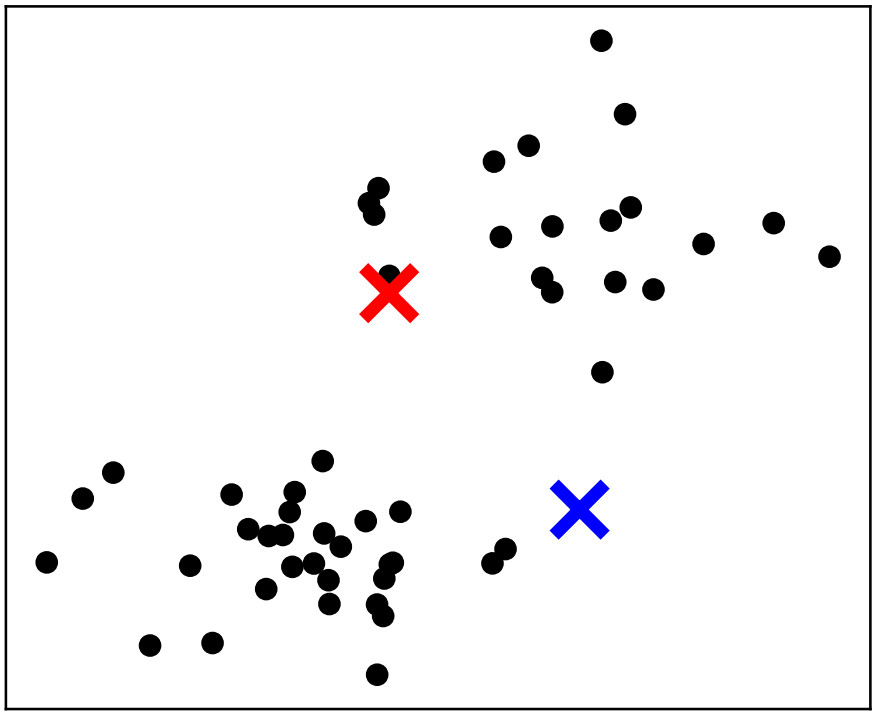
Formation des galaxies

Plan

Principe de fonctionnement



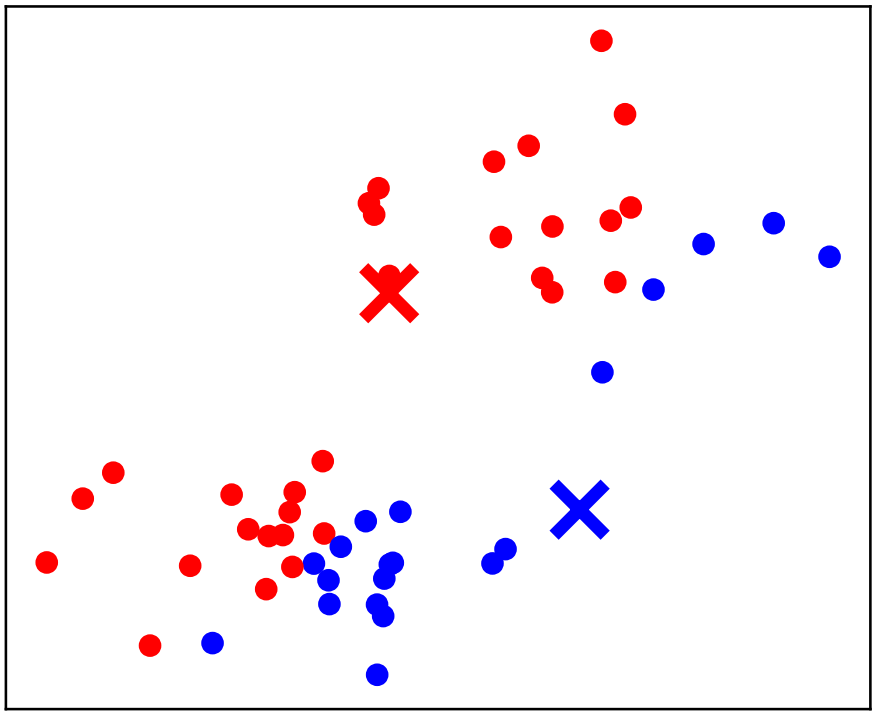
Principe de fonctionnement



Étape 0 : initialiser des centroïdes* de manière aléatoire

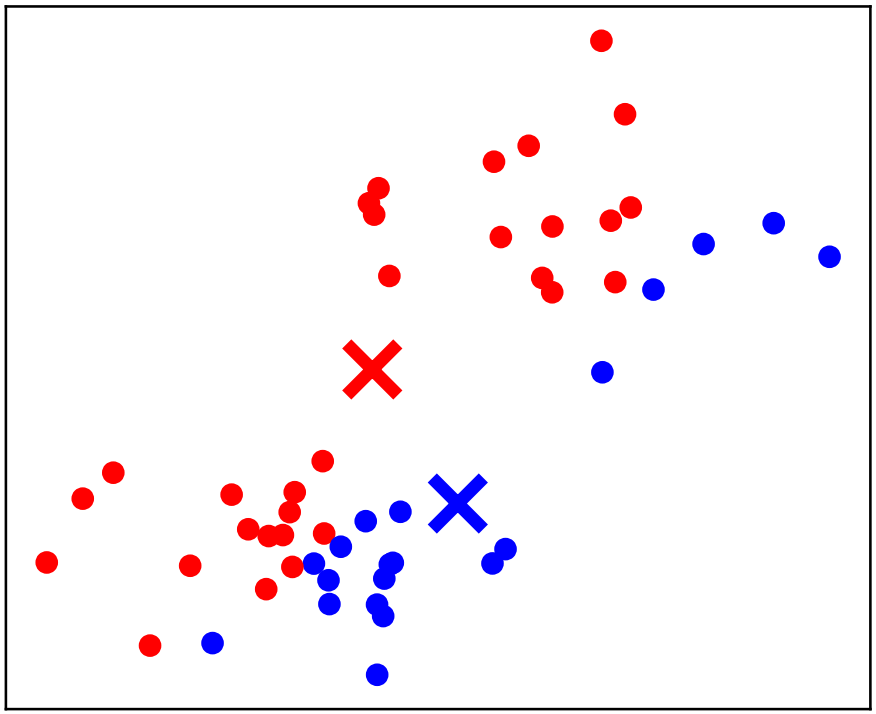
* centroïde := barycentre

Principe de fonctionnement



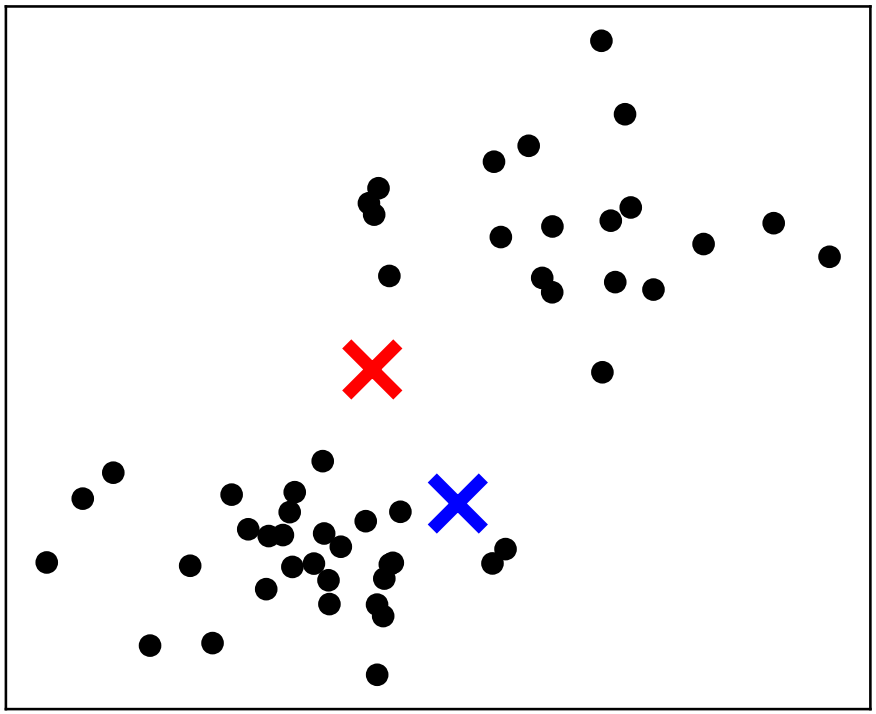
Étape 1a : affecter à chaque observation la classe la plus proche

Principe de fonctionnement



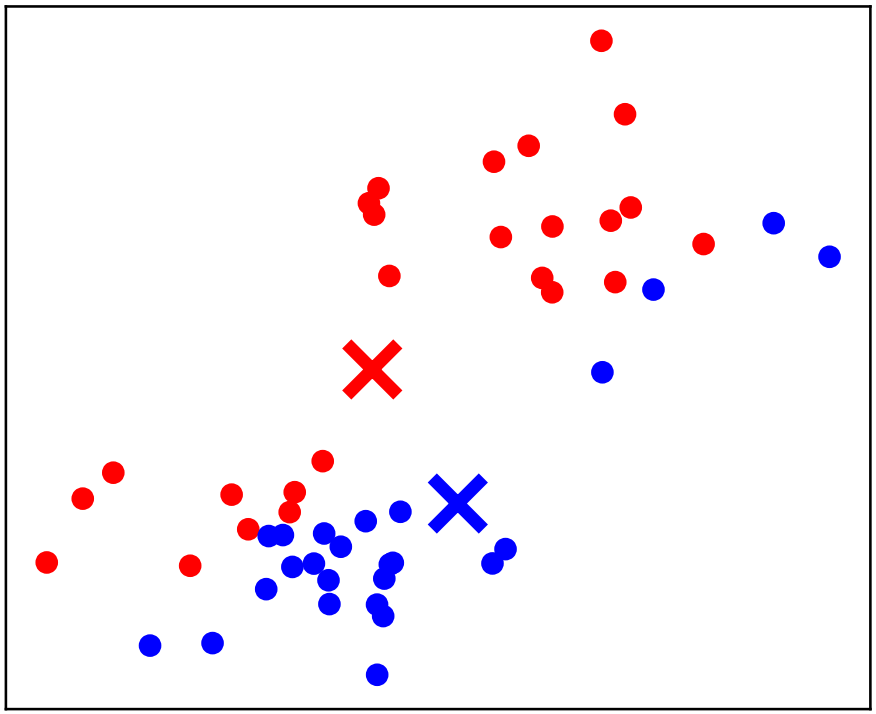
Étape 1b : recalculer la position des centroïdes

Principe de fonctionnement



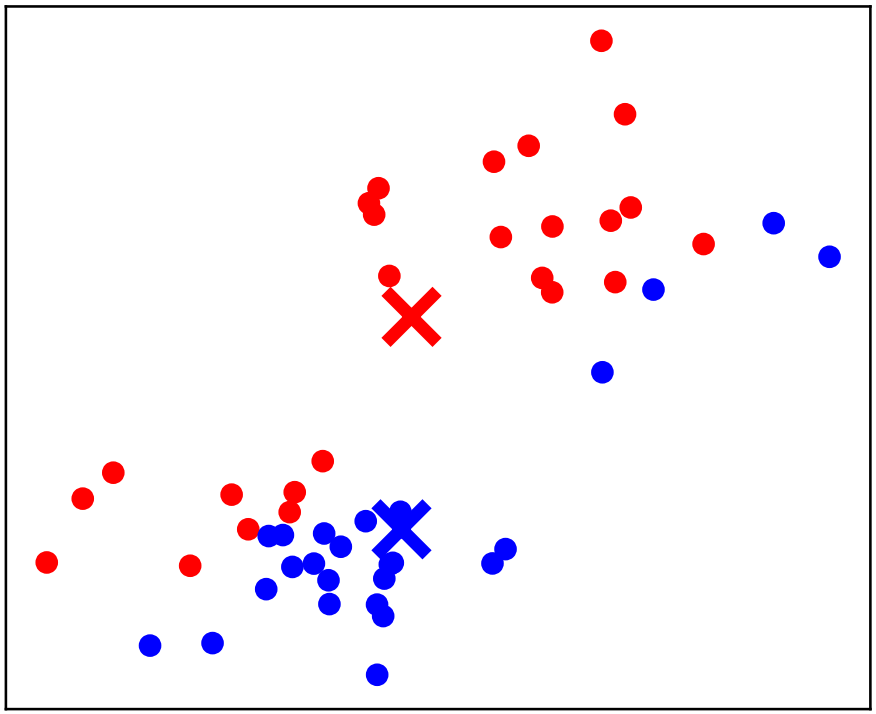
Répéter l'étape 1 avec les nouveaux centroïdes

Principe de fonctionnement



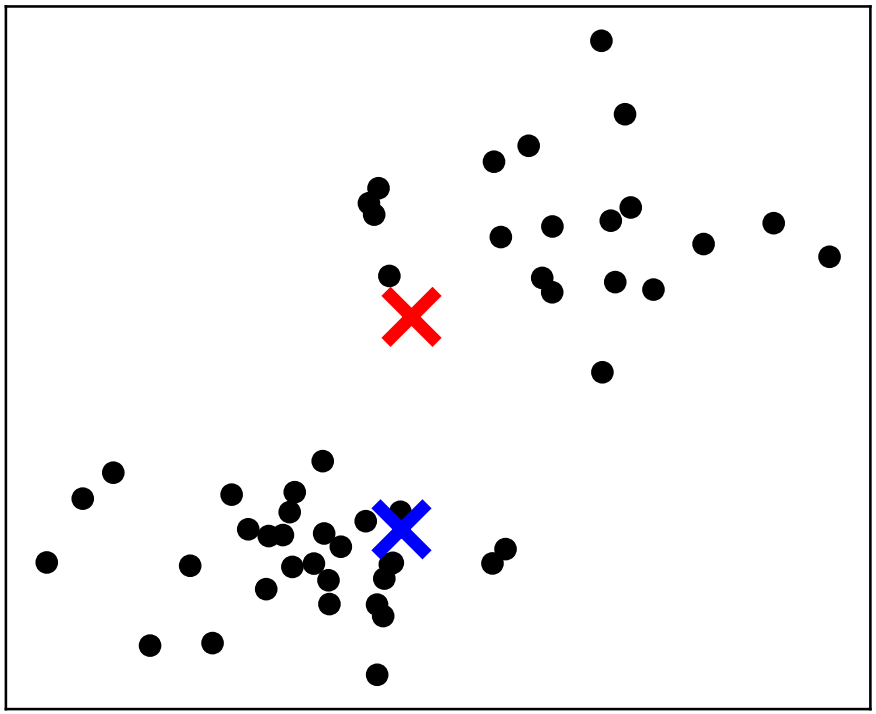
Étape 2a : affecter à chaque observation la classe la plus proche

Principe de fonctionnement



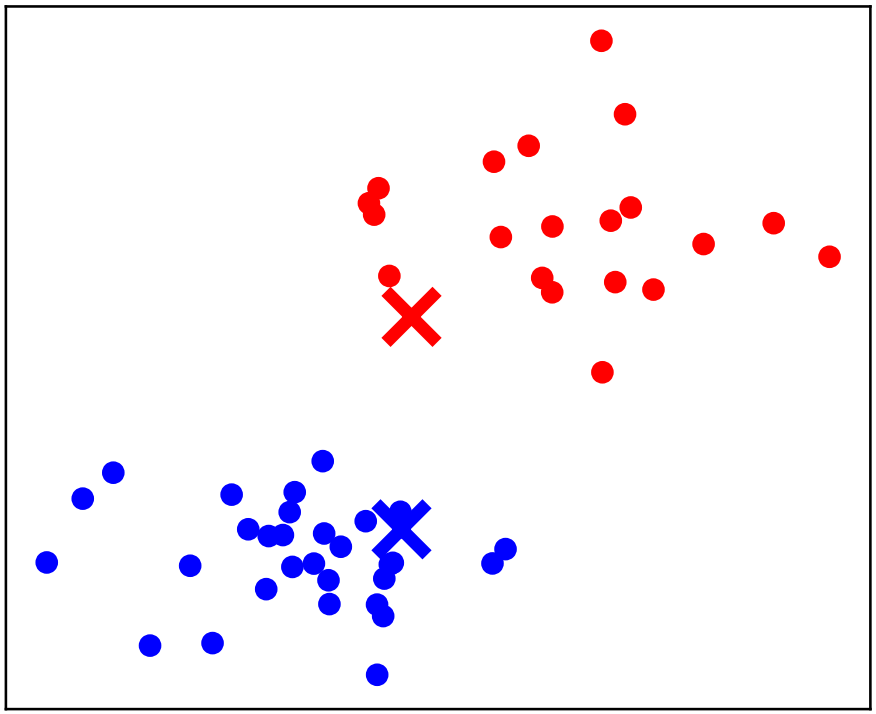
Étape 2b : recalculer la position des centroïdes

Principe de fonctionnement



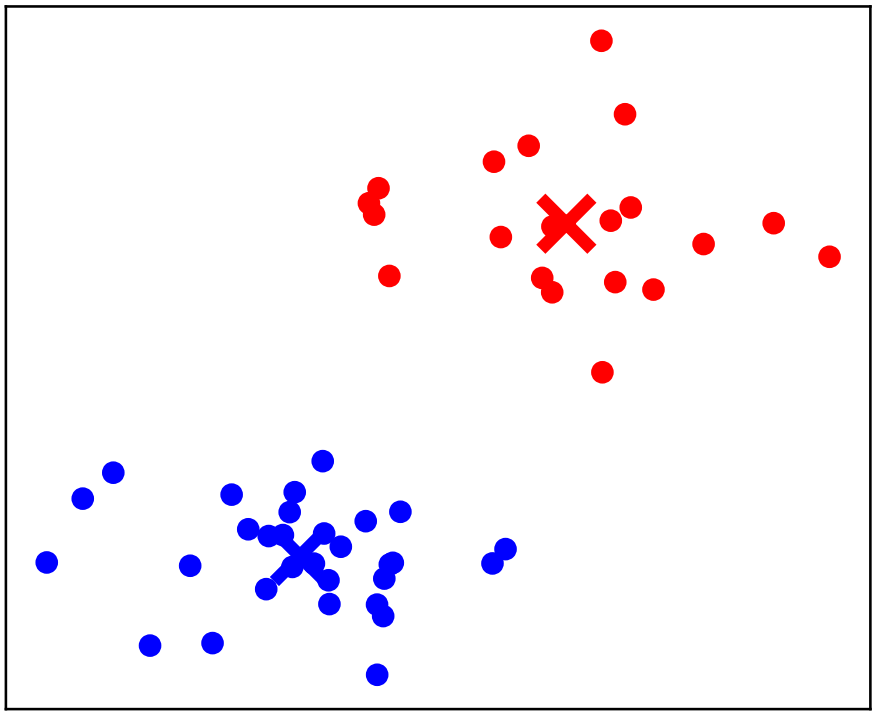
Répéter l'étape 2 pour les nouveaux centroïdes

Principe de fonctionnement



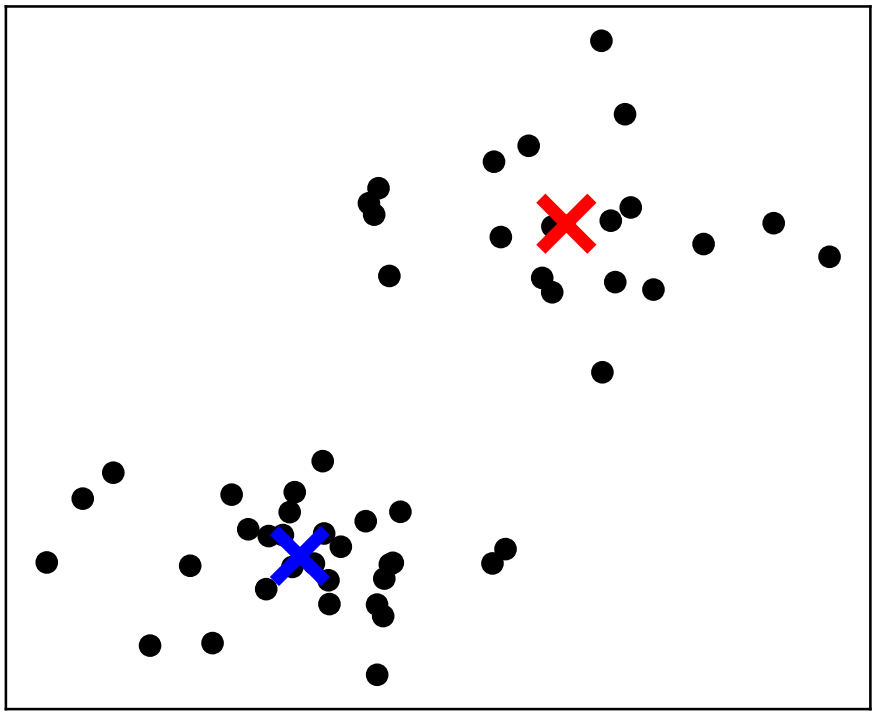
Étape 3a : affecter à chaque observation la classe la plus proche

Principe de fonctionnement



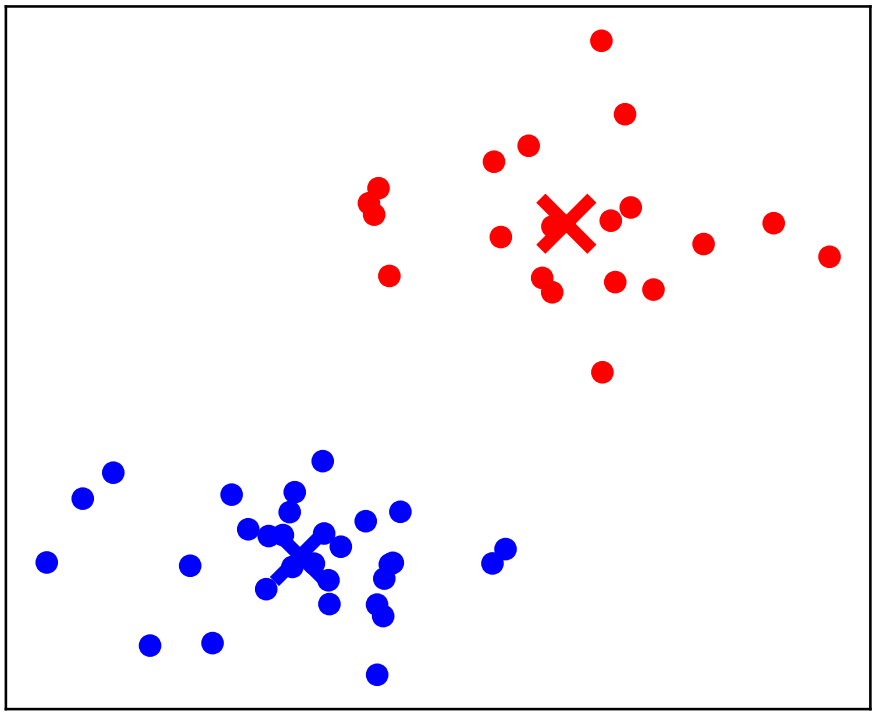
Étape 3b : recalculer la position des centroïdes

Principe de fonctionnement



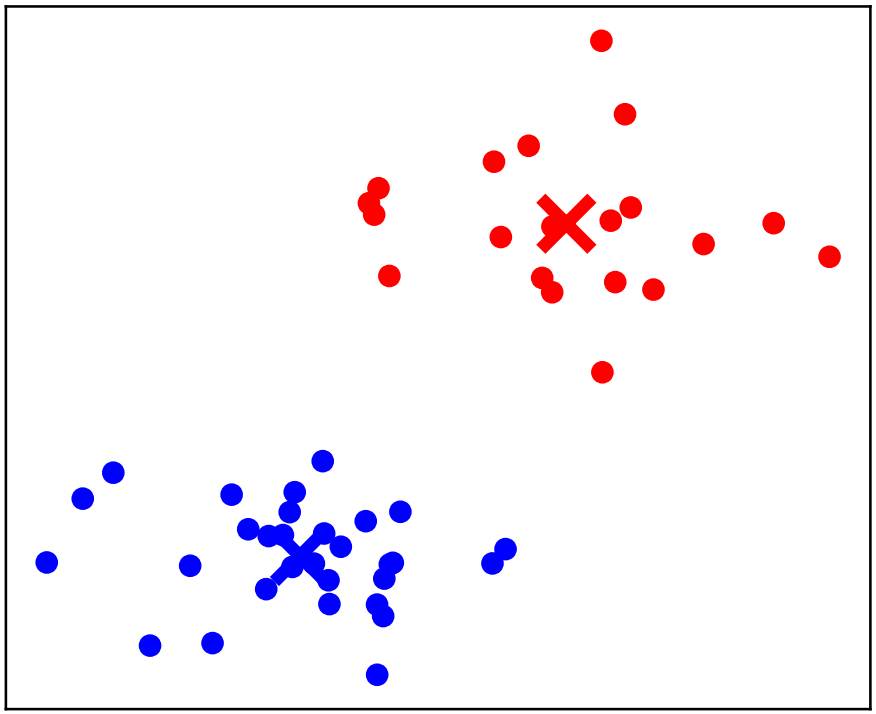
Répéter l'étape 3 pour les nouveaux centroïdes

Principe de fonctionnement



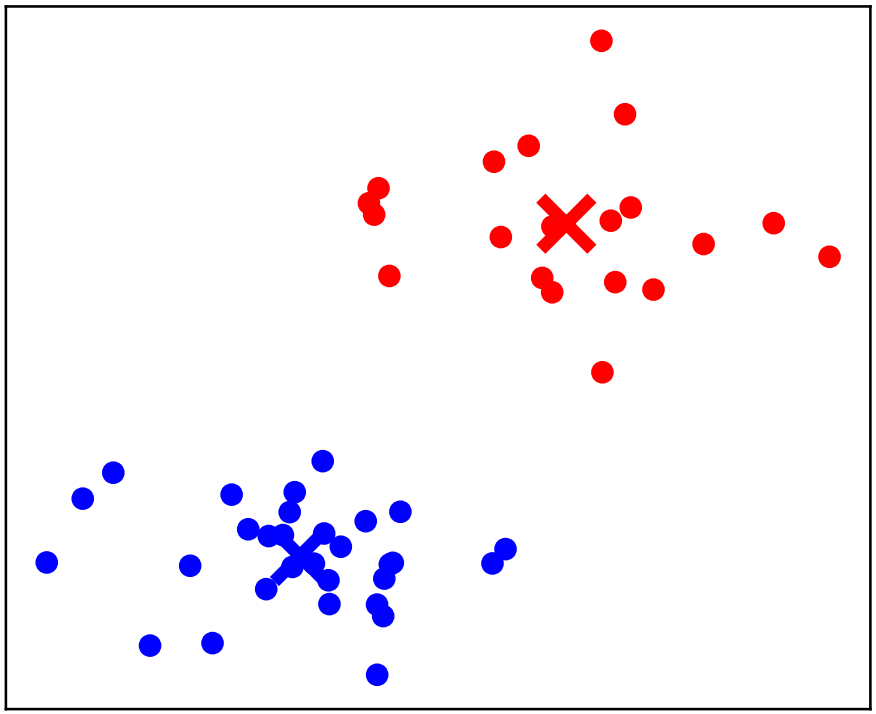
Étape 4a : affecter à chaque observation la classe la plus proche

Principe de fonctionnement



Étape 4b : recalculer la position des centroïdes

Principe de fonctionnement



Convergence : les centroïdes sont identiques à ceux calculés précédemment

Algorithme

Entrées

- k nombre de groupes (*clusters*)
- données d'apprentissage : $\{\mathbf{x}_i\}_{i=1}^{m}$ avec $\mathbf{x}_i \in \mathbb{R}^d$ (centrée-réduite)

Algorithme

Entrées

- k nombre de groupes (*clusters*)
- données d'apprentissage : $\{\mathbf{x}_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$ (centrée-réduite)

Algorithme

Étape 0 Initialiser k centroïdes de manière aléatoire : $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_k\}$

Algorithme

Entrées

- k nombre de groupes (*clusters*)
- données d'apprentissage : $\{\mathbf{x}_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$ (centrée-réduite)

Algorithme

Étape 0 Initialiser k centroïdes de manière aléatoire : $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_k\}$

Répéter jusqu'à convergence

Étape $itera$ pour chaque observation $\mathbf{x}_i$ (pour i allant de 1 à m)

affecter $\hat{y}_i$ (le numéro de cluster de 1 à k) pour l'observation x_i

$\hat{y}_i$ correspond au cluster dont le centroïde est le plus proche de $\mathbf{x}_i$

Algorithme

Entrées

- k nombre de groupes (*clusters*)
- données d'apprentissage : $\{\mathbf{x}_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$ (centrée-réduite)

Algorithme

Étape 0 Initialiser k centroïdes de manière aléatoire : $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_k\}$

Répéter jusqu'à convergence

Étape $itera$ pour chaque observation $\mathbf{x}_i$ (pour i allant de 1 à m)

affecter $\hat{y}_i$ (le numéro de cluster de 1 à k) pour l'observation x_i

$\hat{y}_i$ correspond au cluster dont le centroïde est le plus proche de $\mathbf{x}_i$

$$\hat{y}_i = \underset{c \in \{1, \cdots, k\}}{\arg\min} ||\mathbf{x}_i - \boldsymbol{\mu}_c||^2$$

Algorithme

Entrées

- k nombre de groupes (*clusters*)
- données d'apprentissage : $\{\mathbf{x}_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$ (centrée-réduite)

Algorithme

Étape 0 Initialiser k centroïdes de manière aléatoire : $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_k\}$

Répéter jusqu'à convergence

Étape $iter$**a** pour chaque observation $\mathbf{x}_i$ (pour i allant de 1 à m)

affecter $\hat{y}_i$ (le numéro de cluster de 1 à k) pour l'observation x_i

$\hat{y}_i$ correspond au cluster dont le centroïde est le plus proche de $\mathbf{x}_i$

$$\hat{y}_i = \underset{c \in \{1, \cdots, k\}}{\arg\min} ||\mathbf{x}_i - \boldsymbol{\mu}_c||^2$$

Étape $iter$**b** pour chaque cluster (pour c allant de 1 à k)

recalculer $\boldsymbol{\mu}_c$ le centroïde du cluster c

$\boldsymbol{\mu}_c$ correspond à la moyenne des observations affectées au cluster c

Algorithme

Entrées

- k nombre de groupes (*clusters*)
- données d'apprentissage : $\{\mathbf{x}_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$ (centrée-réduite)

Algorithme

Étape 0 Initialiser k centroïdes de manière aléatoire : $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_k\}$

Répéter jusqu'à convergence

Étape $iter$**a** pour chaque observation $\mathbf{x}_i$ (pour i allant de 1 à m)

affecter $\hat{y}_i$ (le numéro de cluster de 1 à k) pour l'observation x_i

$\hat{y}_i$ correspond au cluster dont le centroïde est le plus proche de $\mathbf{x}_i$

$$\hat{y}_i = \underset{c \in \{1, \cdots, k\}}{\arg\min} ||\mathbf{x}_i - \boldsymbol{\mu}_c||^2$$

Étape $iter$**b** pour chaque cluster (pour c allant de 1 à k)

recalculer $\boldsymbol{\mu}_c$ le centroïde du cluster c

$\boldsymbol{\mu}_c$ correspond à la moyenne des observations affectées au cluster c

$$\boldsymbol{\mu}_c = \frac{1}{m_c} \sum_{\mathbf{x}_i | \hat{y}_i = c} \mathbf{x}_i$$, avec m_c le nombre d'observations qui appartiennent au cluster c

Algorithme

Entrées

- k nombre de groupes (*clusters*)
- données d'apprentissage : $\{\mathbf{x}_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$ (centrée-réduite)

Algorithme

Étape 0 Initialiser k centroïdes de manière aléatoire : $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_k\}$

Répéter jusqu'à convergence

Étape $iter$**a** pour chaque observation $\mathbf{x}_i$ (pour i allant de 1 à m)

affecter $\hat{y}_i$ (le numéro de cluster de 1 à k) pour l'observation x_i

$\hat{y}_i$ correspond au cluster dont le centroïde est le plus proche de $\mathbf{x}_i$

$$\hat{y}_i = \underset{c \in \{1, \cdots, k\}}{\arg\min} ||\mathbf{x}_i - \boldsymbol{\mu}_c||^2$$

Étape $iter$**b** pour chaque cluster (pour c allant de 1 à k)

recalculer $\boldsymbol{\mu}_c$ le centroïde du cluster c

$\boldsymbol{\mu}_c$ correspond à la moyenne des observations affectées au cluster c

$$\boldsymbol{\mu}_c = \frac{1}{m_c} \sum_{\mathbf{x}_i | \hat{y}_i = c} \mathbf{x}_i$$, avec m_c le nombre d'observations qui appartiennent au cluster c

Critère d'arrêt (convergence)

- lorsque les centroïdes sont identiques : les affectations ne changent plus
- on a atteint un nombre pré-défini d'itérations : $iter > ITER\_MAX$

Activité 1

Implémentez votre version des k-Moyennes.

Inertie

L'**inertie** $\mathcal{I}_T$ d'un nuage des points est représentée par la distance au carré des points à leur centroïde

- données (d'apprentissage) : $\{\mathbf{x}_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$
- centroïde : $\boldsymbol{\mu} = \frac{1}{m} \sum_i^m \mathbf{x}_i$
- inertie : $\mathcal{I}_T = \sum_{\mathbf{x}_i} ||\mathbf{x}_i - \boldsymbol{\mu}||^2$

Inertie

L'**inertie** $\mathcal{I}_T$ d'un nuage des points est représentée par la distance au carré des points à leur centroïde

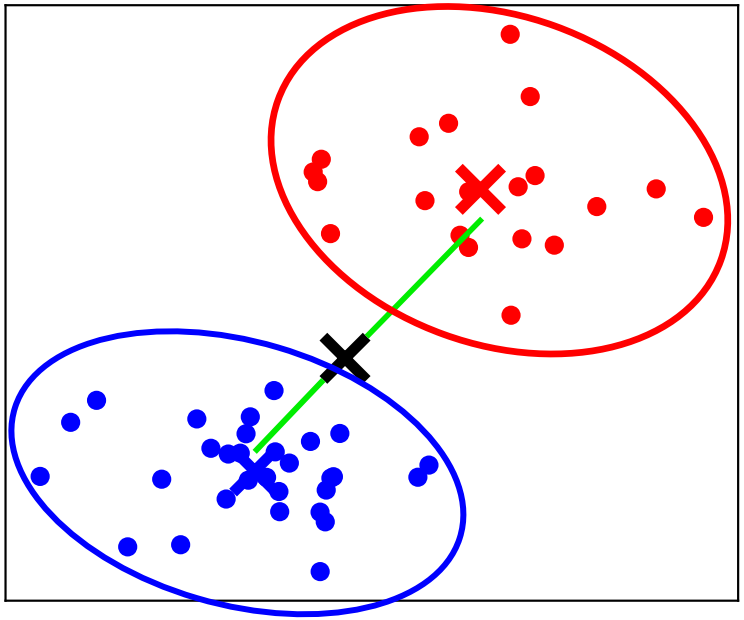
- données (d'apprentissage) : $\{\mathbf{x}_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$
- centroïde : $\boldsymbol{\mu} = \frac{1}{m} \sum_i^m \mathbf{x}_i$
- inertie : $\mathcal{I}_T = \sum_{\mathbf{x}_i} ||\mathbf{x}_i - \boldsymbol{\mu}||^2$

Remarque : l'**inertie totale** dépend uniquement des données (et pas des clusters auxquels appartiennent les observaitions)

Formule de décomposition de l'inertie : théorème de Huygens

Inertie totale = Inertie inter-classe
+ Inertie intra-classe

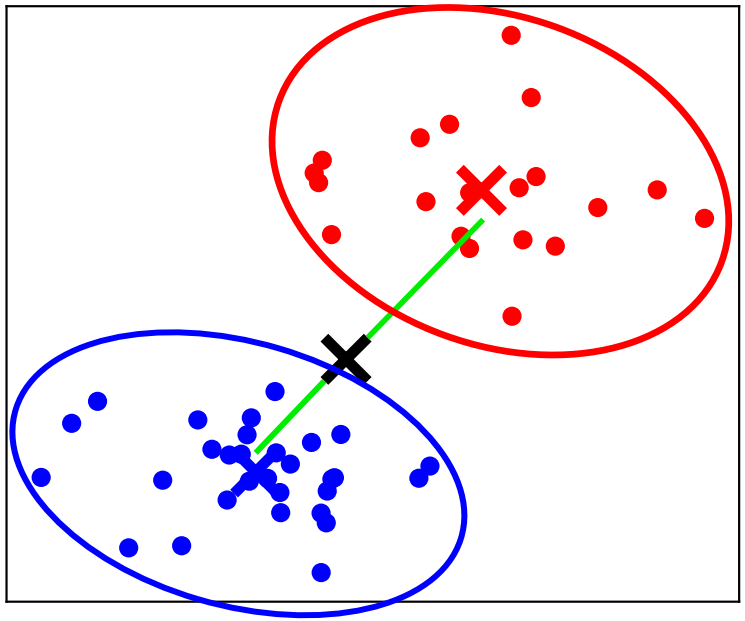
$$\mathcal{I}_T = \mathcal{I}_B + \mathcal{I}_W$$



Formule de décomposition de l'inertie : théorème de Huygens

Inertie totale = Inertie inter-classe
+ Inertie intra-classe

$$\mathcal{I}_T = \mathcal{I}_B + \mathcal{I}_W$$

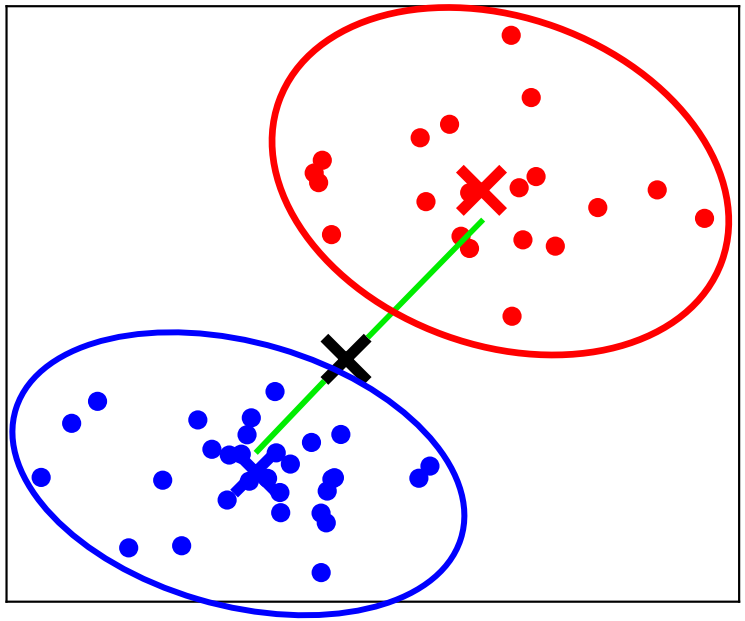


$$\underbrace{\sum_{\mathbf{x}_i} ||\mathbf{x}_i - \boldsymbol{\mu}||^2}_{\text{Inertie totale}} = \underbrace{\sum_{c=1}^{k} m_c ||\boldsymbol{\mu}_c - \boldsymbol{\mu}||^2}_{\text{Indicateur de séparabilité des classes : dispersion des centroïdes des clusters autour du centroïde global}} + \underbrace{\sum_{c=1}^{k} \sum_{\mathbf{x}_i | \hat{y}_i = c} ||\mathbf{x}_i - \boldsymbol{\mu}_c||^2}_{\textbf{Indicateur de compacité des classes : dispersion à l'intérieur de chaque cluster}}$$

Formule de décomposition de l'inertie : théorème de Huygens

Inertie totale = Inertie inter-classe
+ Inertie intra-classe

$$\mathcal{I}_T = \mathcal{I}_B + \mathcal{I}_W$$



$$\underbrace{\sum_{\mathbf{x}_i} ||\mathbf{x}_i - \boldsymbol{\mu}||^2}_{\text{Inertie totale}} = \underbrace{\sum_{c=1}^{k} m_c ||\boldsymbol{\mu}_c - \boldsymbol{\mu}||^2}_{\text{Indicateur de séparabilité des classes : dispersion des centroïdes des clusters autour du centroïde global}} + \underbrace{\sum_{c=1}^{k} \sum_{\mathbf{x}_i | \hat{y}_i = c} ||\mathbf{x}_i - \boldsymbol{\mu}_c||^2}_{\textbf{Indicateur de compacité des classes : dispersion à l'intérieur de chaque cluster}}$$

Double objectif

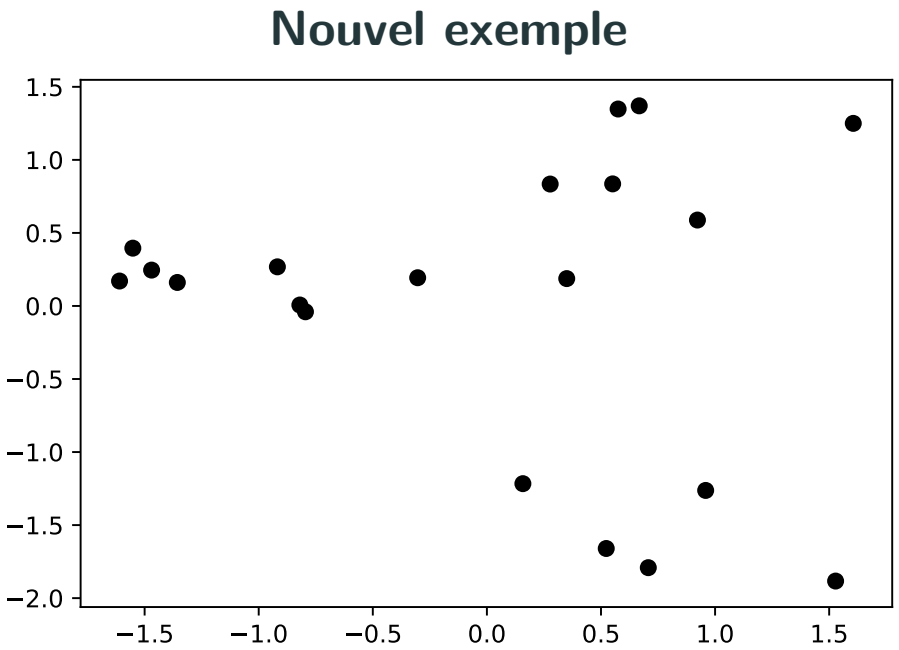
- avoir des groupes homogènes : $\min \mathcal{I}_W$
- avoir des groupes séparés les uns des autres : $\max \mathcal{I}_B$

Activité 2

Affinez votre version de l'algorithme des k-Moyennes pour qu'elle retourne la valeur de l'inertie intra-classe I_W.

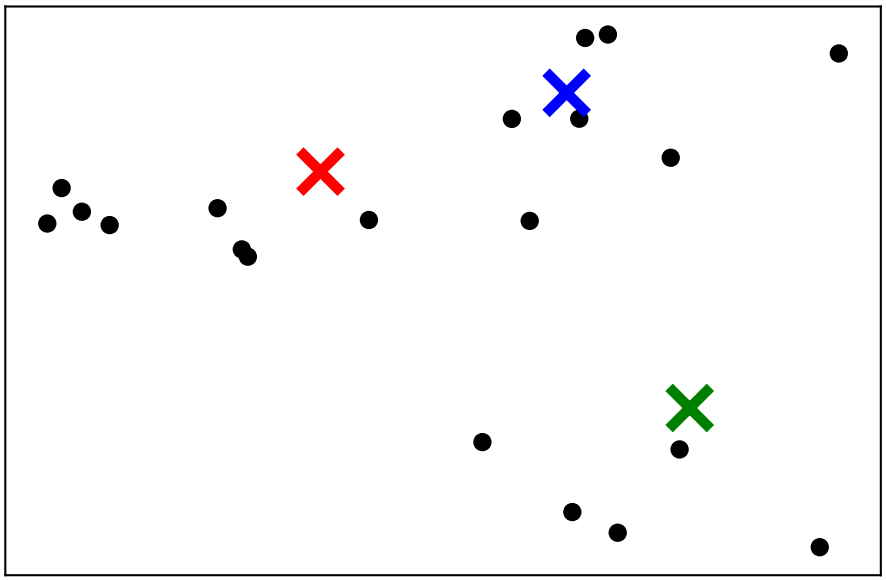
Plan

Un minimum local ?



Un minimum local ?

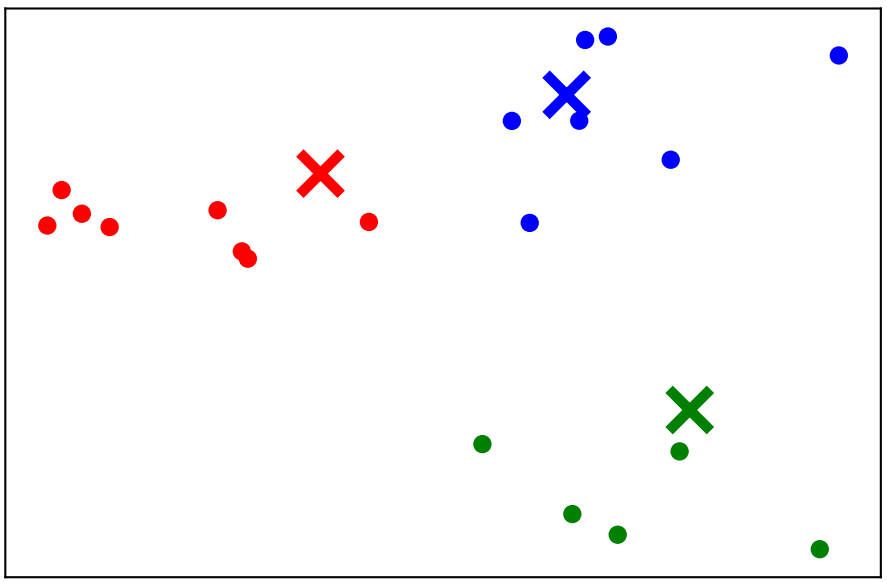
Nouvel exemple : 3 clusters



Étape 0 : initialiser des centroïdes de manière aléatoire

Un minimum local ?

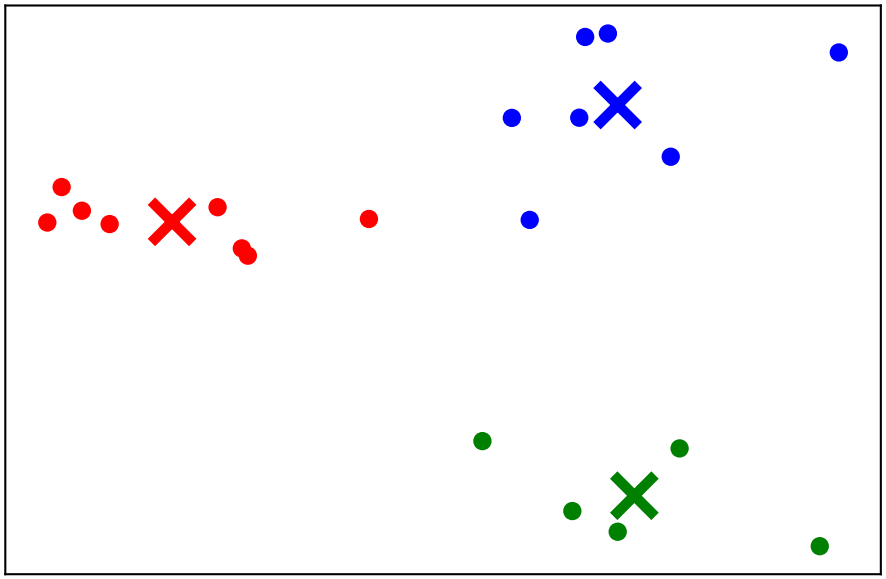
Nouvel exemple : 3 clusters



Étape 1a : affecter à chaque observation la classe la plus proche

Un minimum local ?

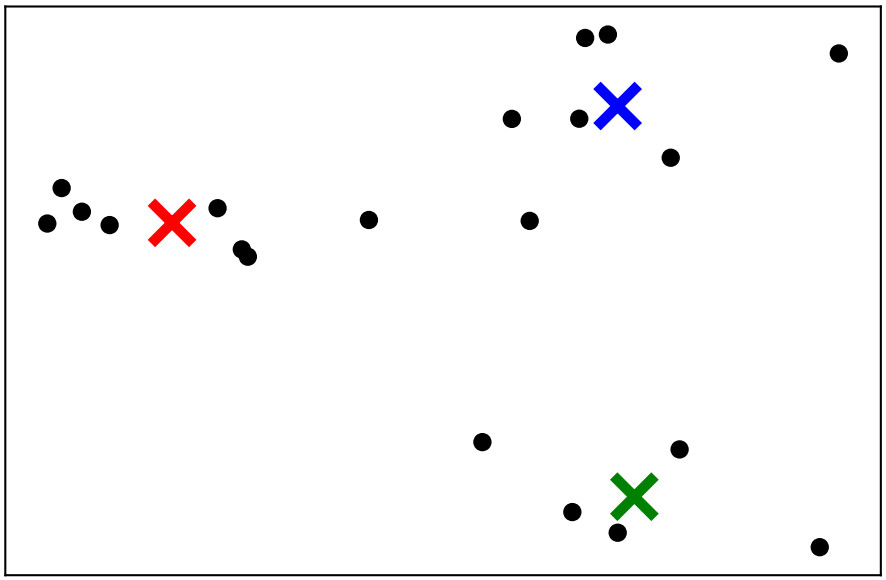
Nouvel exemple : 3 clusters



Étape 1b : recalculer la position des centroïdes

Un minimum local ?

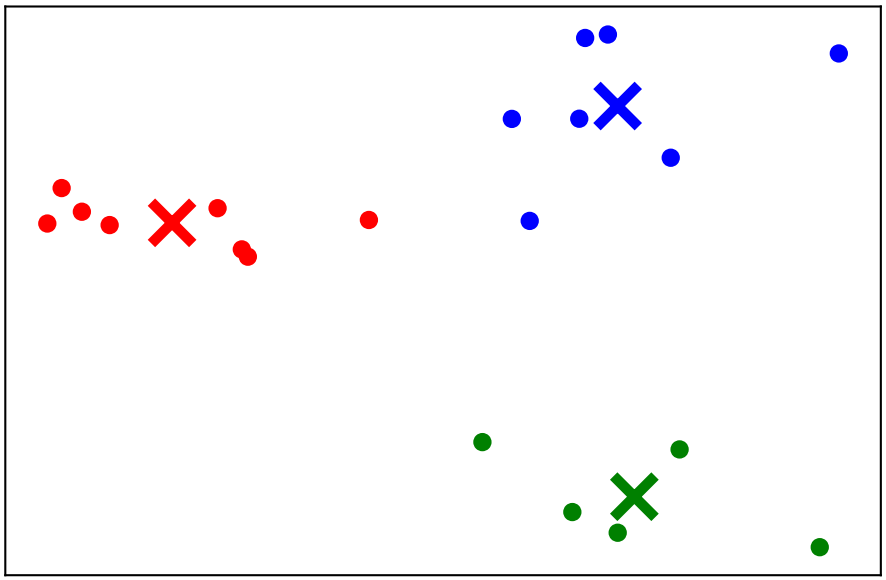
Nouvel exemple : 3 clusters



Répéter l'étape 1 avec les nouveaux centroïdes

Un minimum local ?

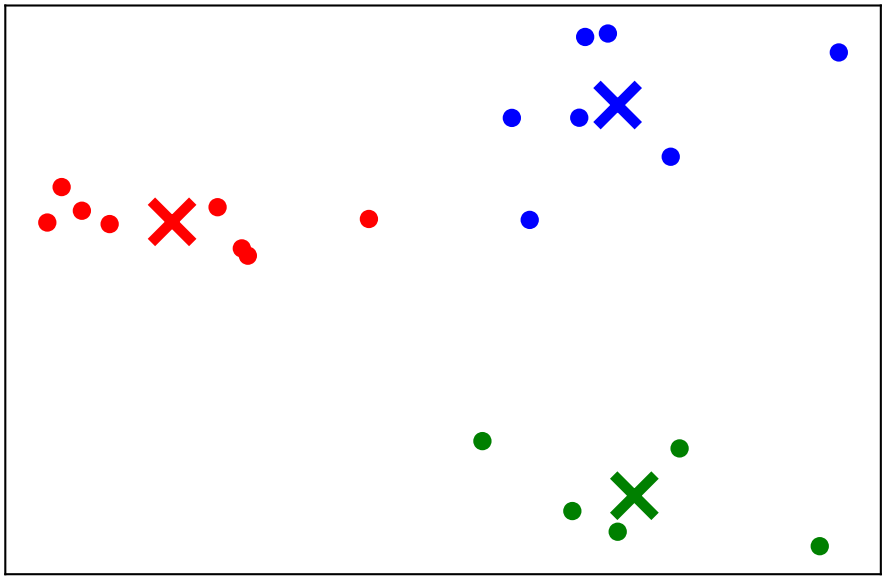
Nouvel exemple : 3 clusters



Étape 2a : affecter à chaque observation la classe la plus proche

Un minimum local ?

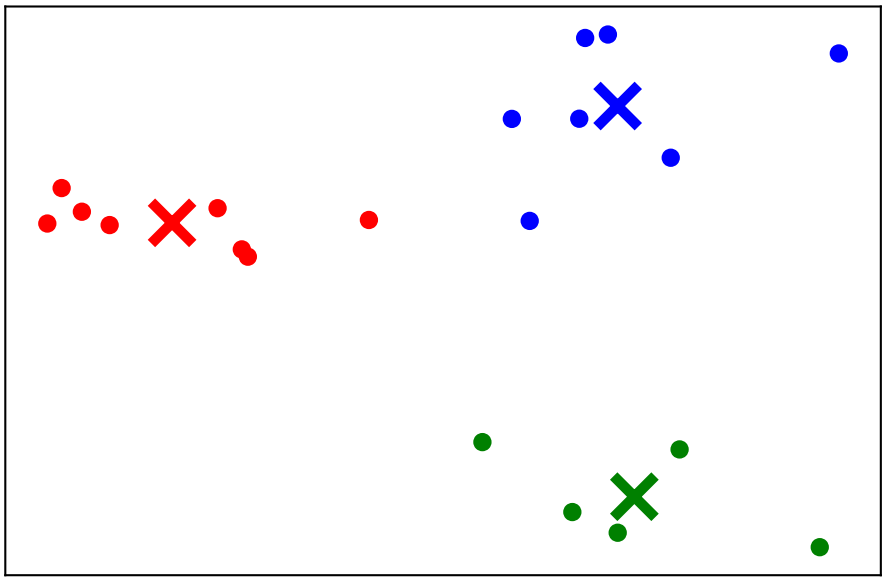
Nouvel exemple : 3 clusters



Étape 2b : recalculer la position des centroïdes

Un minimum local ?

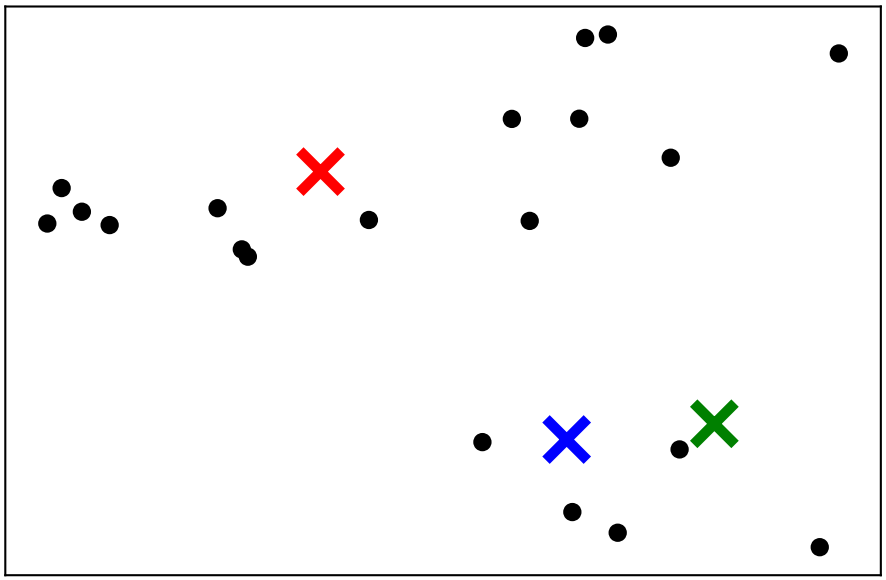
Nouvel exemple : 3 clusters



Convergence : les centroïdes sont identiques à ceux calculés précédemment

Un minimum local ?

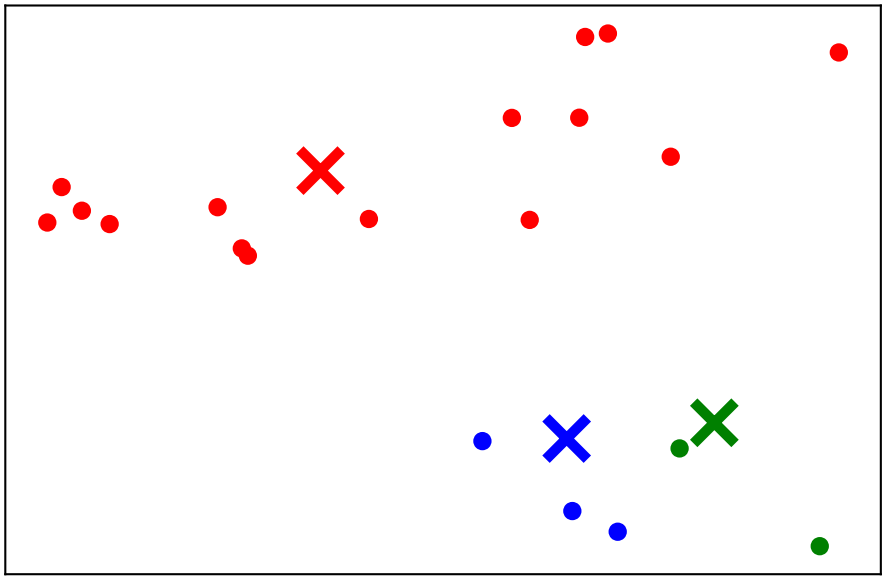
Nouvel exemple : 3 clusters



Étape 0 : initialiser des centroïdes de manière aléatoire

Un minimum local ?

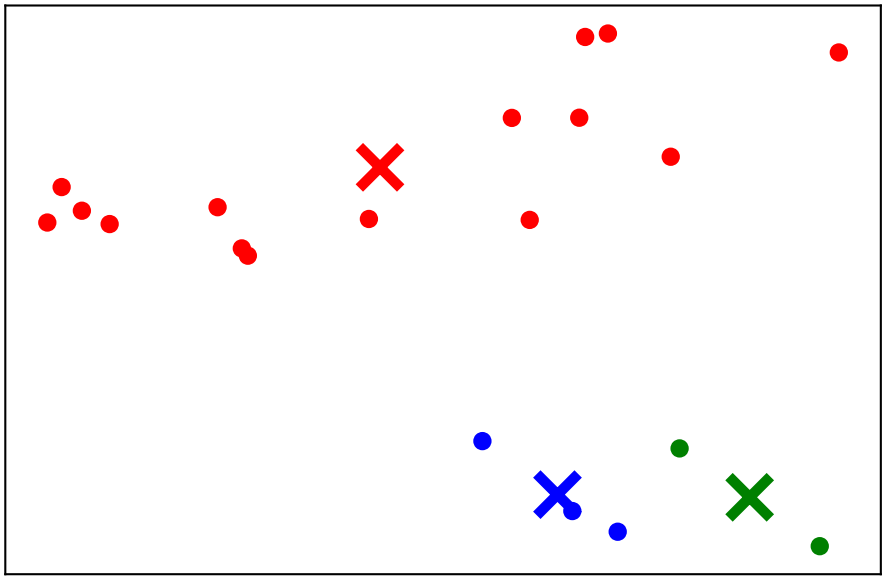
Nouvel exemple : 3 clusters



Étape 1a : affecter à chaque observation la classe la plus proche

Un minimum local ?

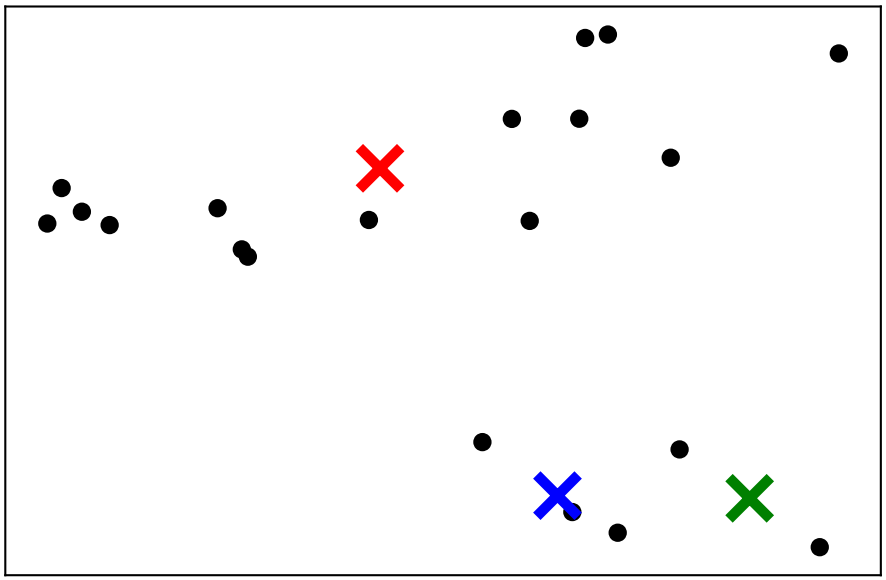
Nouvel exemple : 3 clusters



Étape 1b : recalculer la position des centroïdes

Un minimum local ?

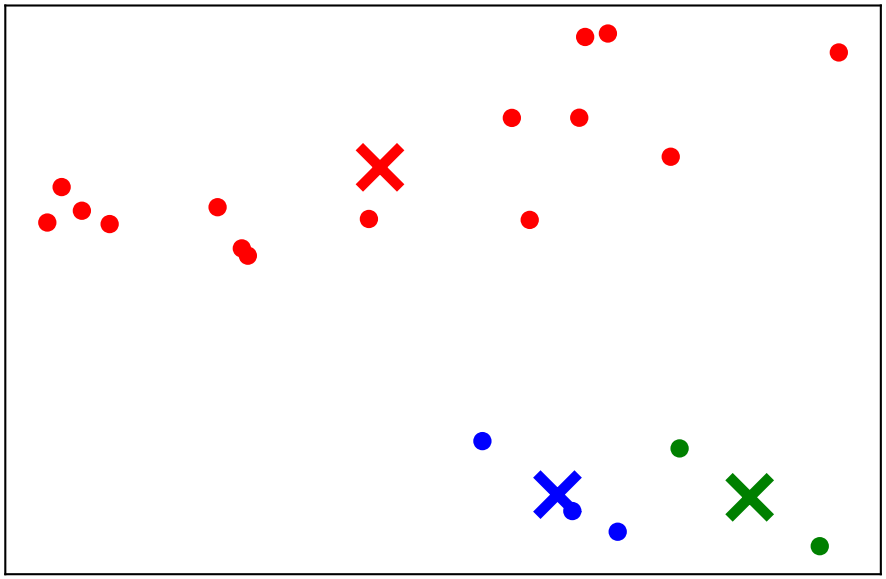
Nouvel exemple : 3 clusters



Répéter l'étape 1 avec les nouveaux centroïdes

Un minimum local ?

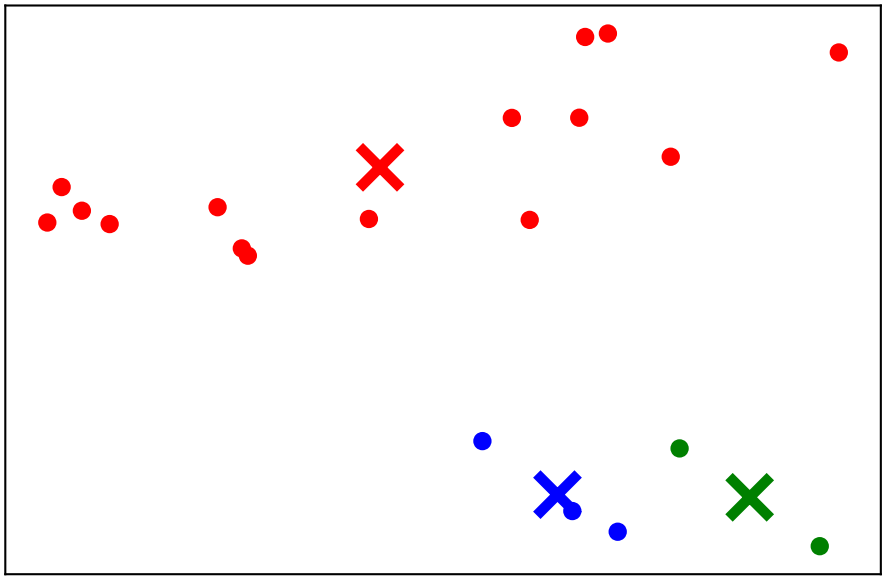
Nouvel exemple : 3 clusters



Étape 2a : affecter à chaque observation la classe la plus proche

Un minimum local ?

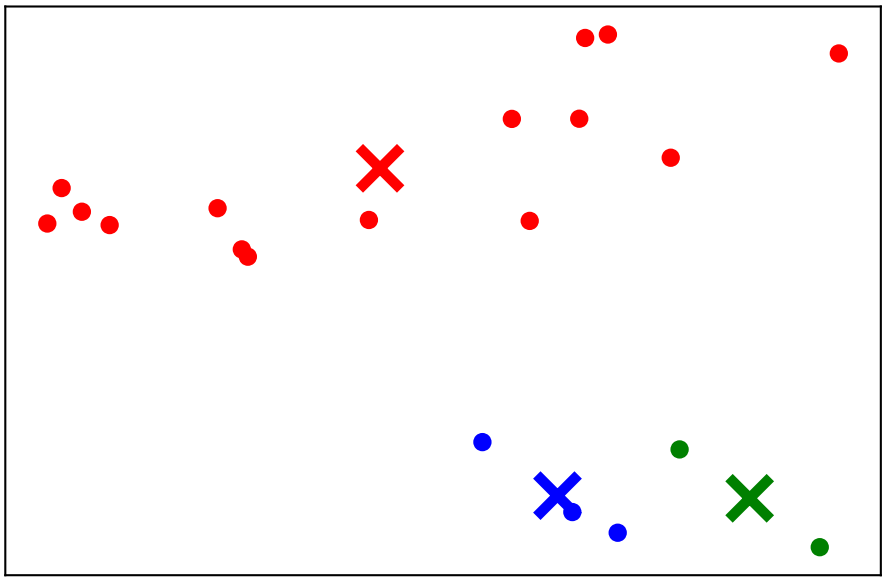
Nouvel exemple : 3 clusters



Étape 2b : recalculer la position des centroïdes

Un minimum local ?

Nouvel exemple : 3 clusters



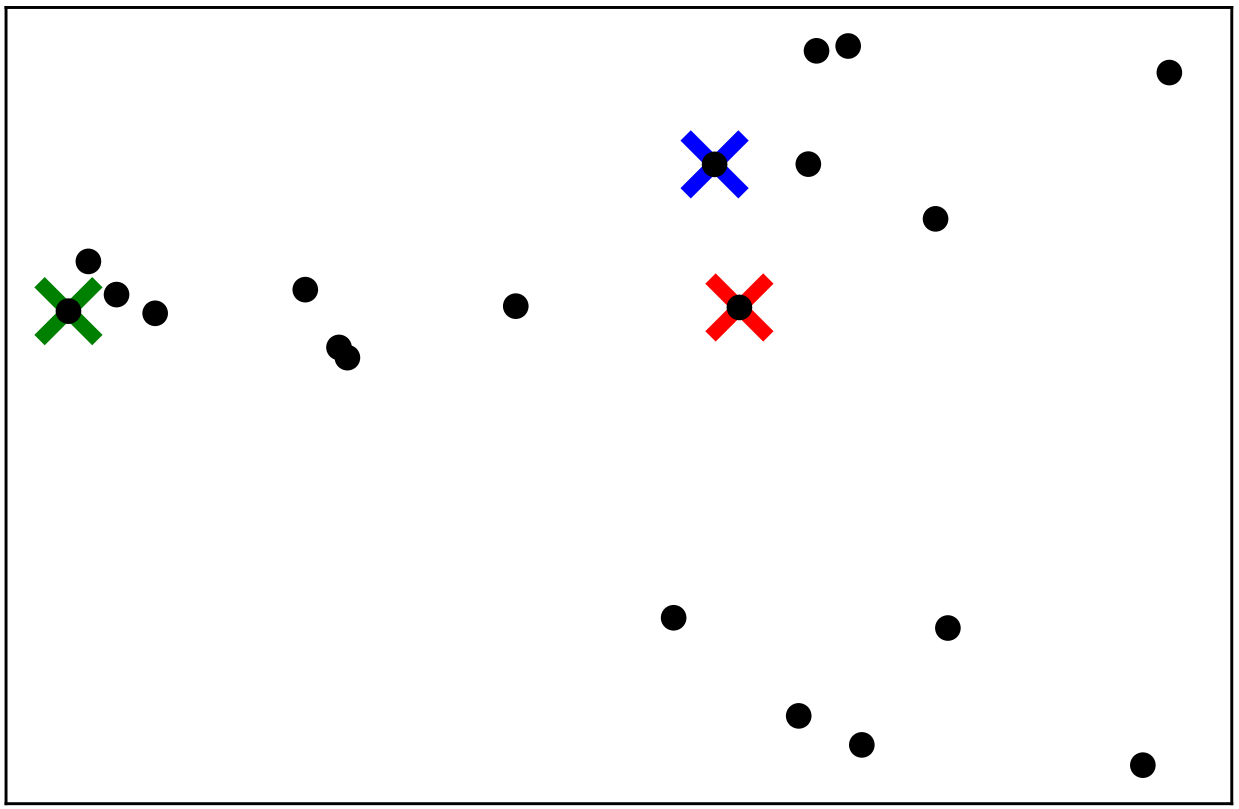
Convergence : les centroïdes sont identiques à ceux calculés précédemment

Activité 3

Dans votre implémentation, modifiez l'initialisation des centroïdes pour qu'ils soient sélectionnés aléatoirement parmi les données d'apprentissage.

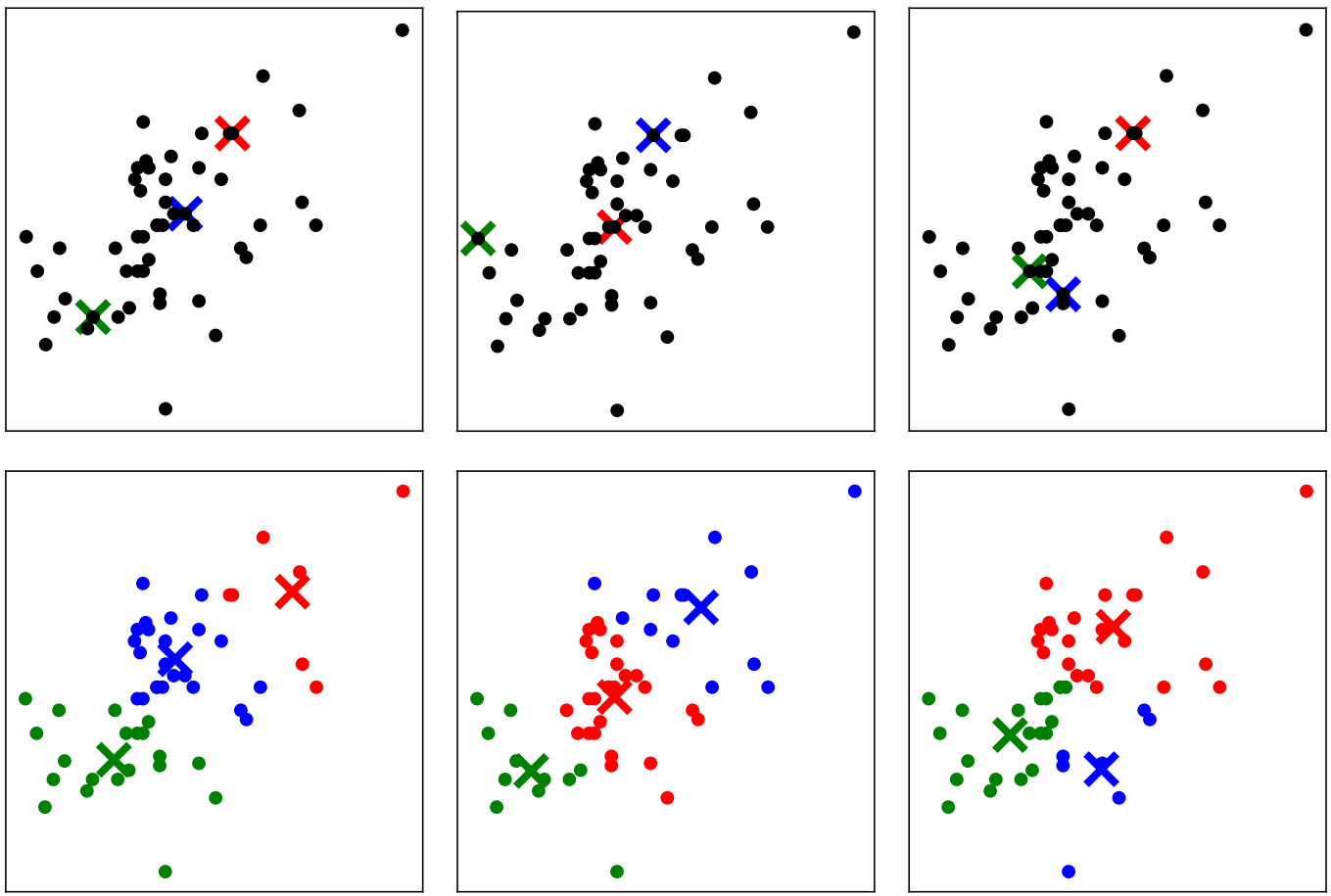
Choix des centroïdes initiaux

- Le choix des centroïdes se fait parmi les données d'apprentissage



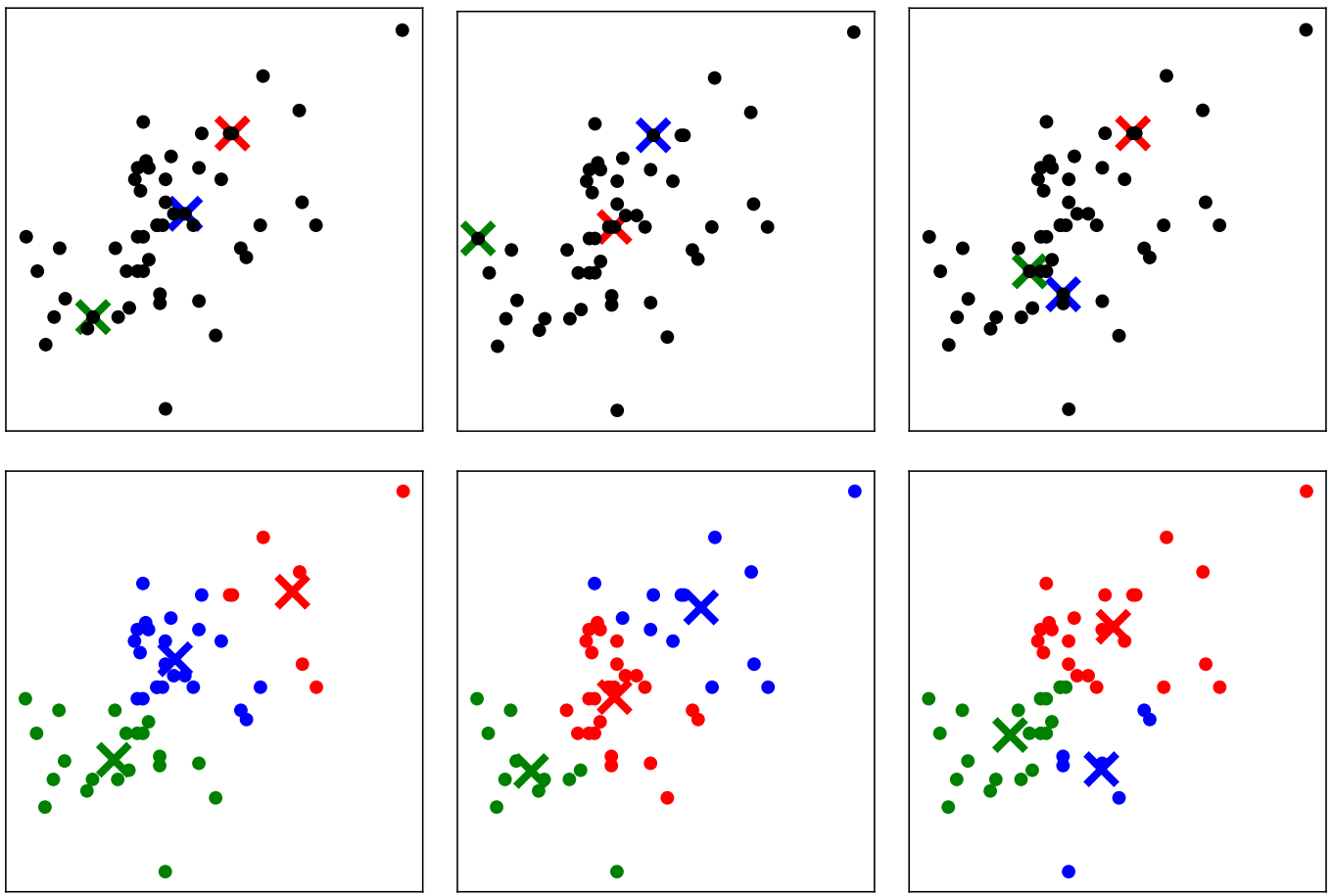
Éviter un minimum local

- **Répéter** t fois l'algorithme des k-Moyennes avec différentes initialisations
- **Calculer** l'inertie inter-classe $\mathcal{I}_W^t$ pour chaque itération t
- **Sélectionner** le partitionnement pour lequel $\mathcal{I}_W$ est minimal : $\mathcal{I}_W = \min_t \mathcal{I}_W^t$



Éviter un minimum local

- **Répéter** t fois l'algorithme des k-Moyennes avec différentes initialisations
- **Calculer** l'inertie inter-classe $\mathcal{I}_W^t$ pour chaque itération t
- **Sélectionner** le partitionnement pour lequel $\mathcal{I}_W$ est minimal : $\mathcal{I}_W = \min_t \mathcal{I}_W^t$



$\mathcal{I}_W = 35.8$ $\mathcal{I}_W = 39.4$ $\mathcal{I}_W = 46.1$

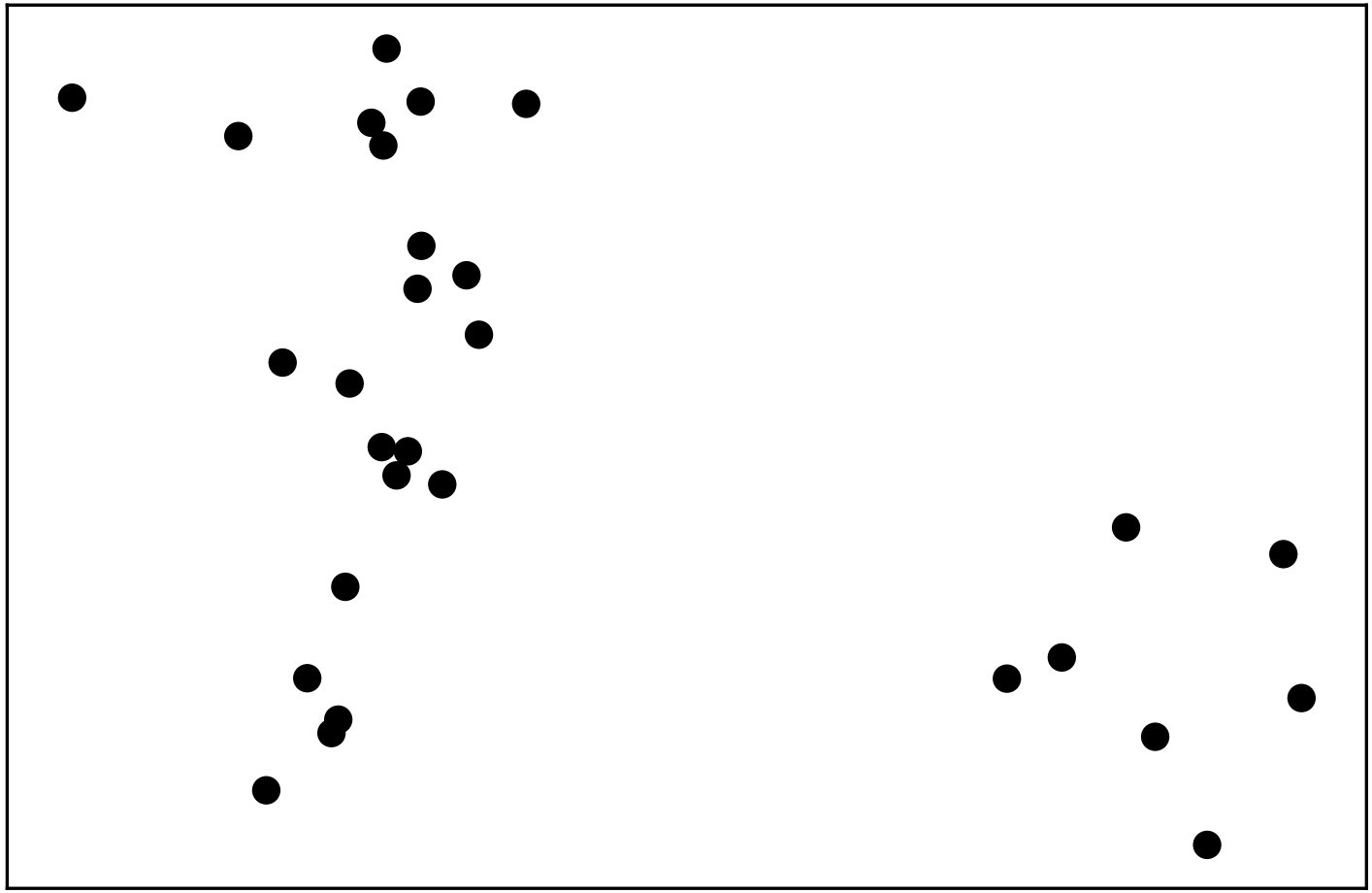
Activité 4

Répéter plusieurs fois l'algorithme des k-Moyennes et garder le résultat qui donne l'inertie intra-classe minimale.

Plan

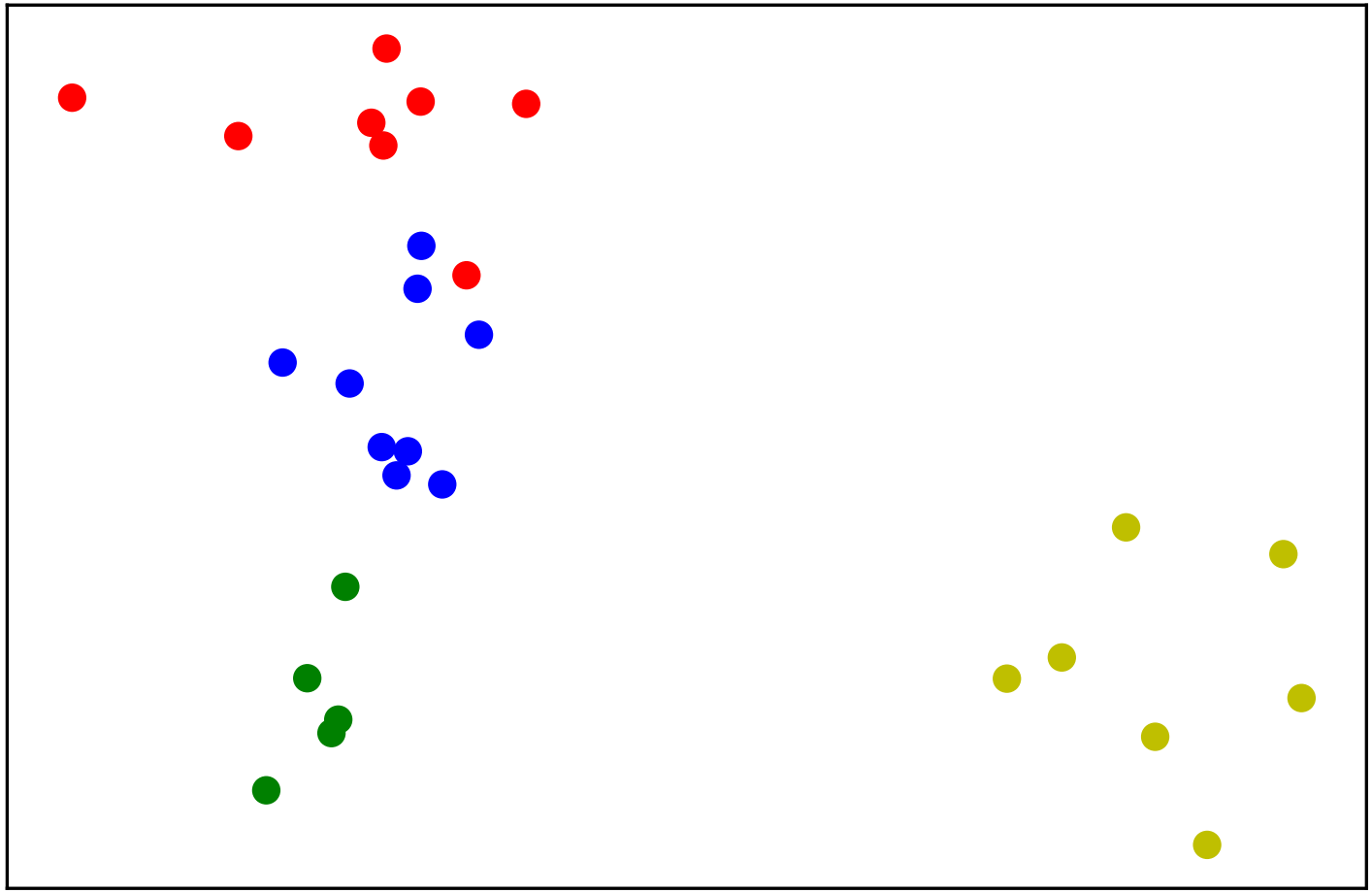
Comment choisir le nombre de clusters ?

Combien de clusters voyez-vous ?



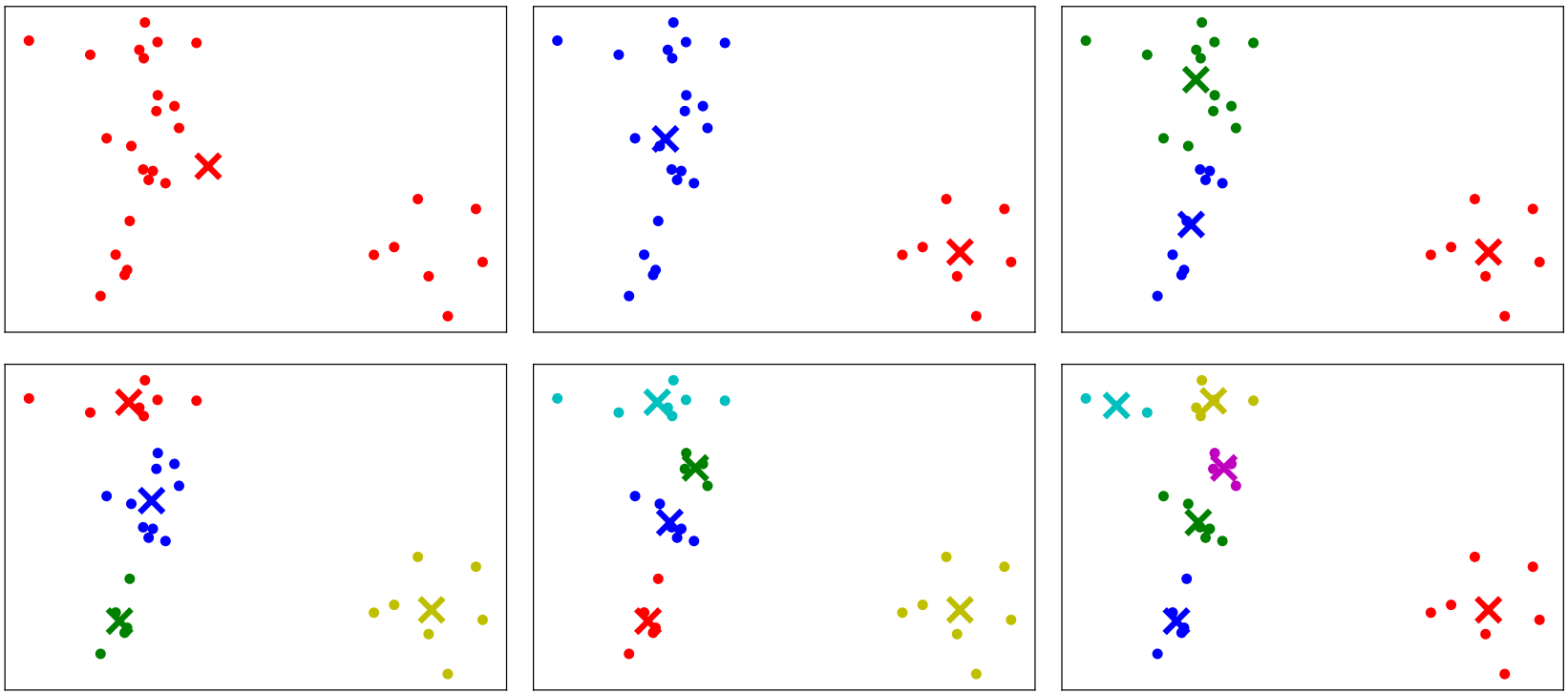
Comment choisir le nombre de clusters ?

Combien de clusters voyez-vous ?



Comment choisir le nombre de clusters ?

Les partitionnements obtenus pour différentes valeurs de k



Comment choisir le nombre de clusters ?

- **Manuellement** : en pratique souvent la meilleure des techniques

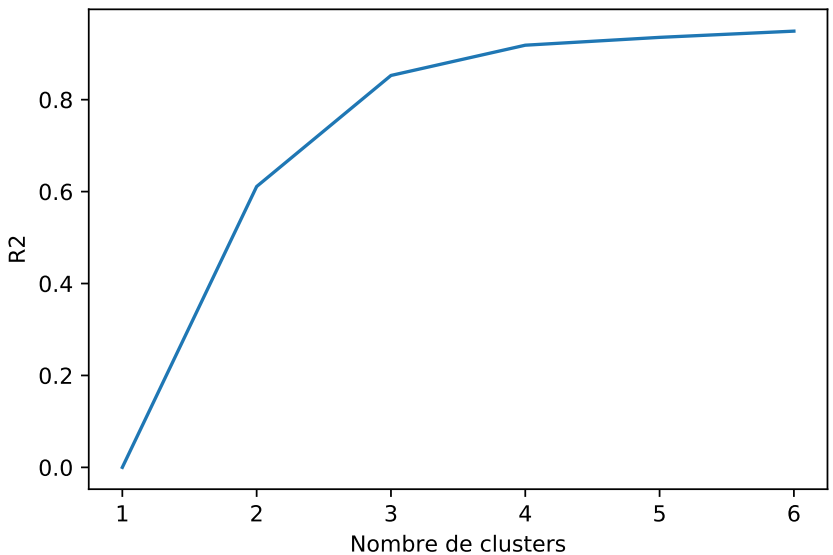
Comment choisir le nombre de clusters ?

- **Manuellement** : en pratique souvent la meilleure des techniques
- **Automatiquement** : étudier les valeurs d'inertie intra-classe en fonction du nombre de clusters k

Automatiquement

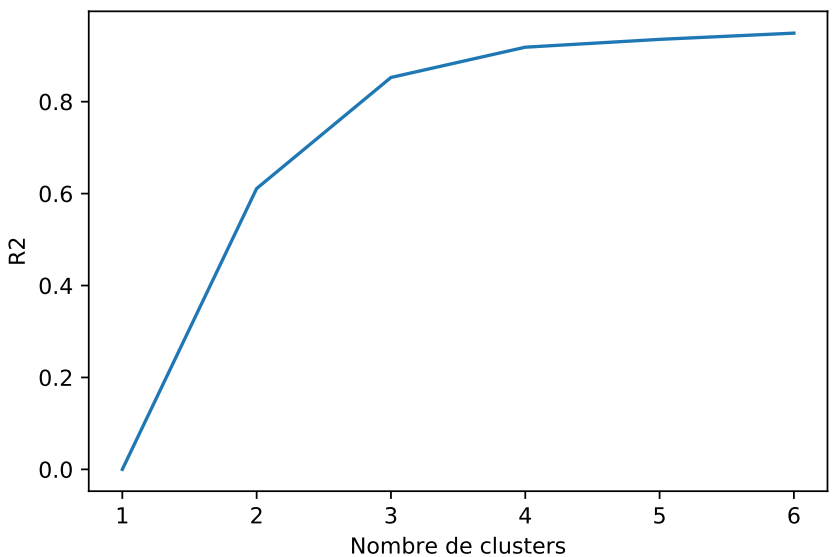
R^2 : la proportion de variance expliquée par les clusters

$$R^2 = \frac{\mathcal{I}_B}{\mathcal{I}_T} = 1 - \frac{\mathcal{I}_W}{\mathcal{I}_T}$$

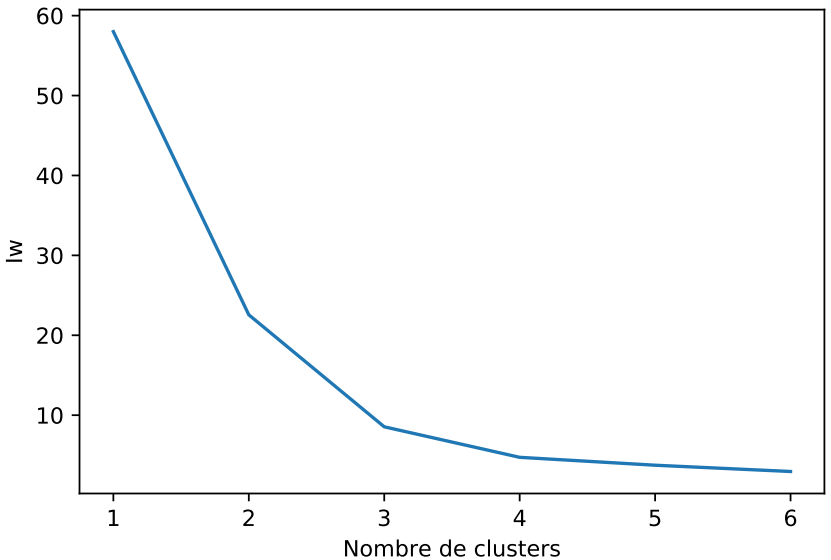


R^2 : la proportion de variance expliquée par les clusters

$$R^2 = \frac{\mathcal{I}_B}{\mathcal{I}_T} = 1 - \frac{\mathcal{I}_W}{\mathcal{I}_T}$$

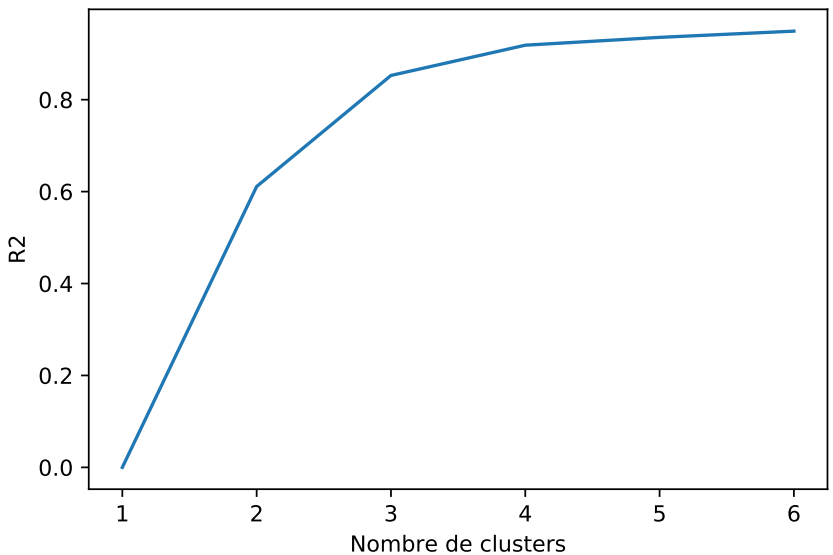


Méthode d'Elbow (le coude)

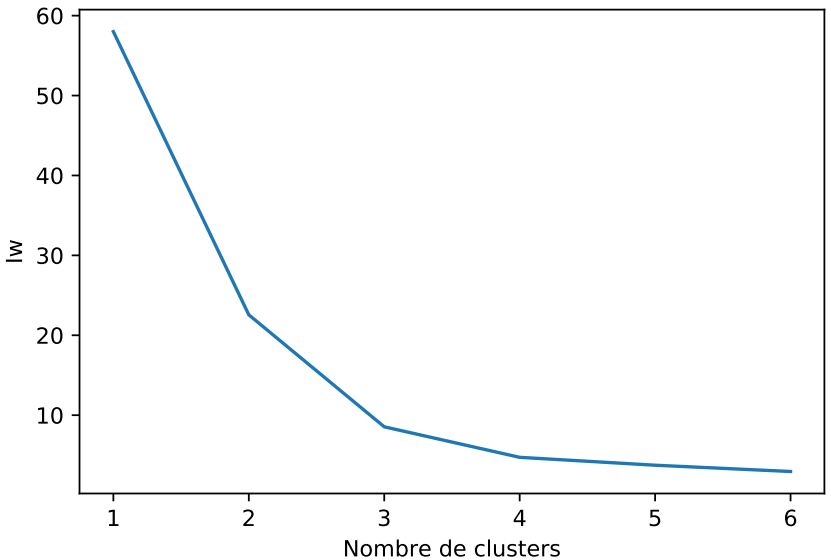


R^2 : la proportion de variance expliquée par les clusters

$$R^2 = \frac{\mathcal{I}_B}{\mathcal{I}_T} = 1 - \frac{\mathcal{I}_W}{\mathcal{I}_T}$$



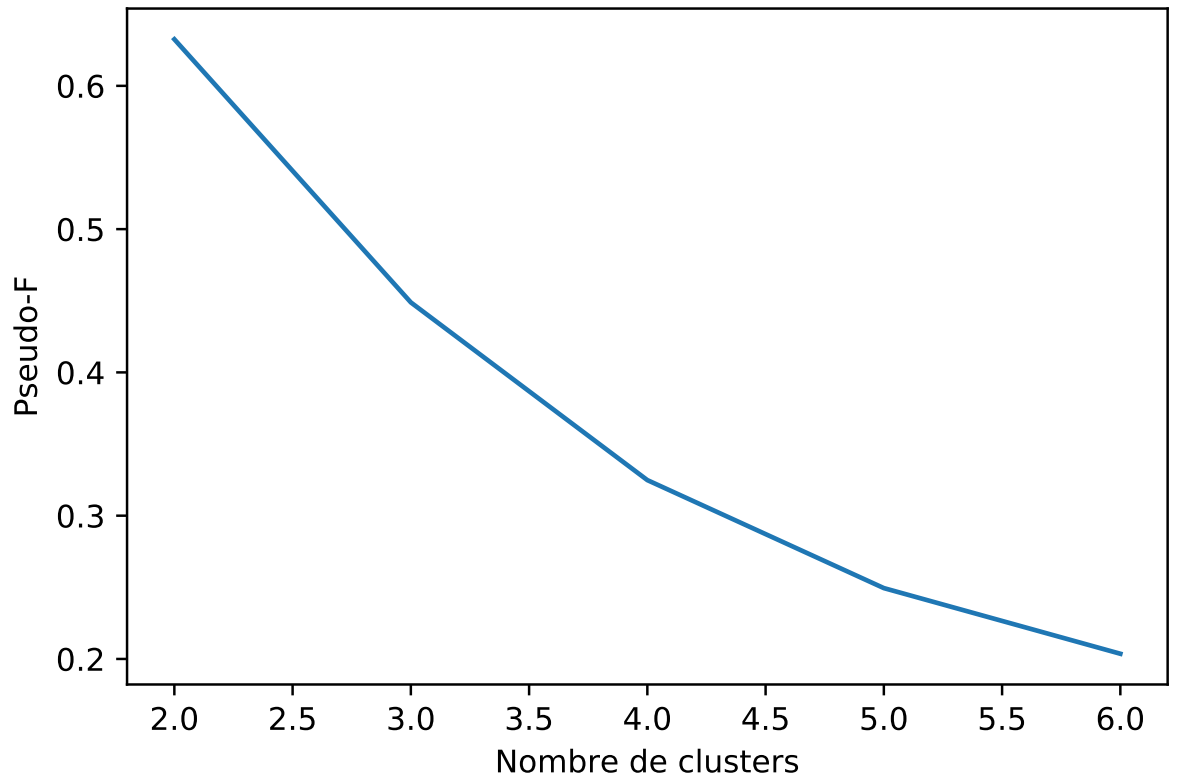
Méthode d'Elbow (le coude)



Limitation : généralement aucun *coude* n'est visible

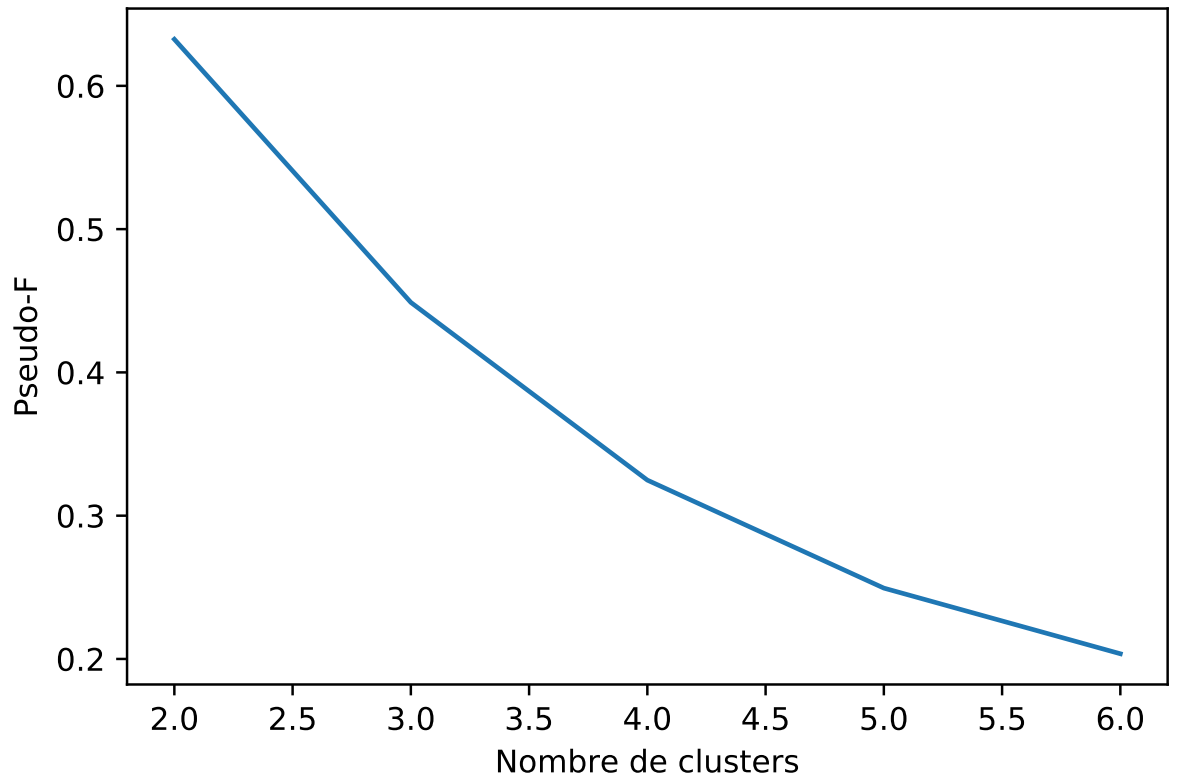
Comment choisir le nombre de clusters ?

Pseudo-F : mesure de séparation entre toutes les classes Pseudo-F $= \frac{\frac{R^2}{k-1}}{\frac{1-R^2}{m-k}}$



Comment choisir le nombre de clusters ?

Pseudo-F : mesure de séparation entre toutes les classes Pseudo-F $= \frac{\frac{R^2}{k-1}}{\frac{1-R^2}{m-k}}$



Et d'autres encore :

- Cubic Clustering Criterion (CCC)
- le coefficient de silhouette (cohésion et séparation)

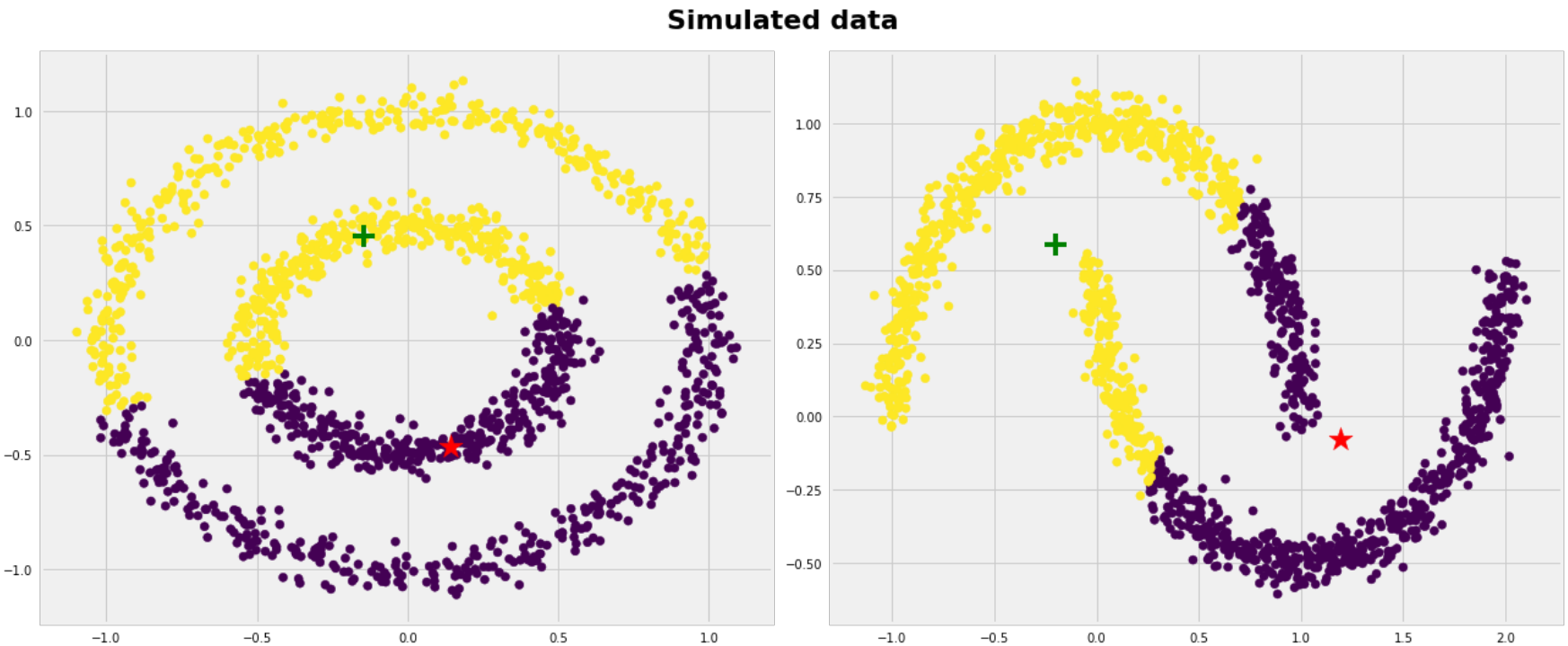
Activité 5

- Testez un ou plusieurs critères pour sélectionnez au mieux le nombre de clusters.
- Qu'en pensez-vous ?

Plan

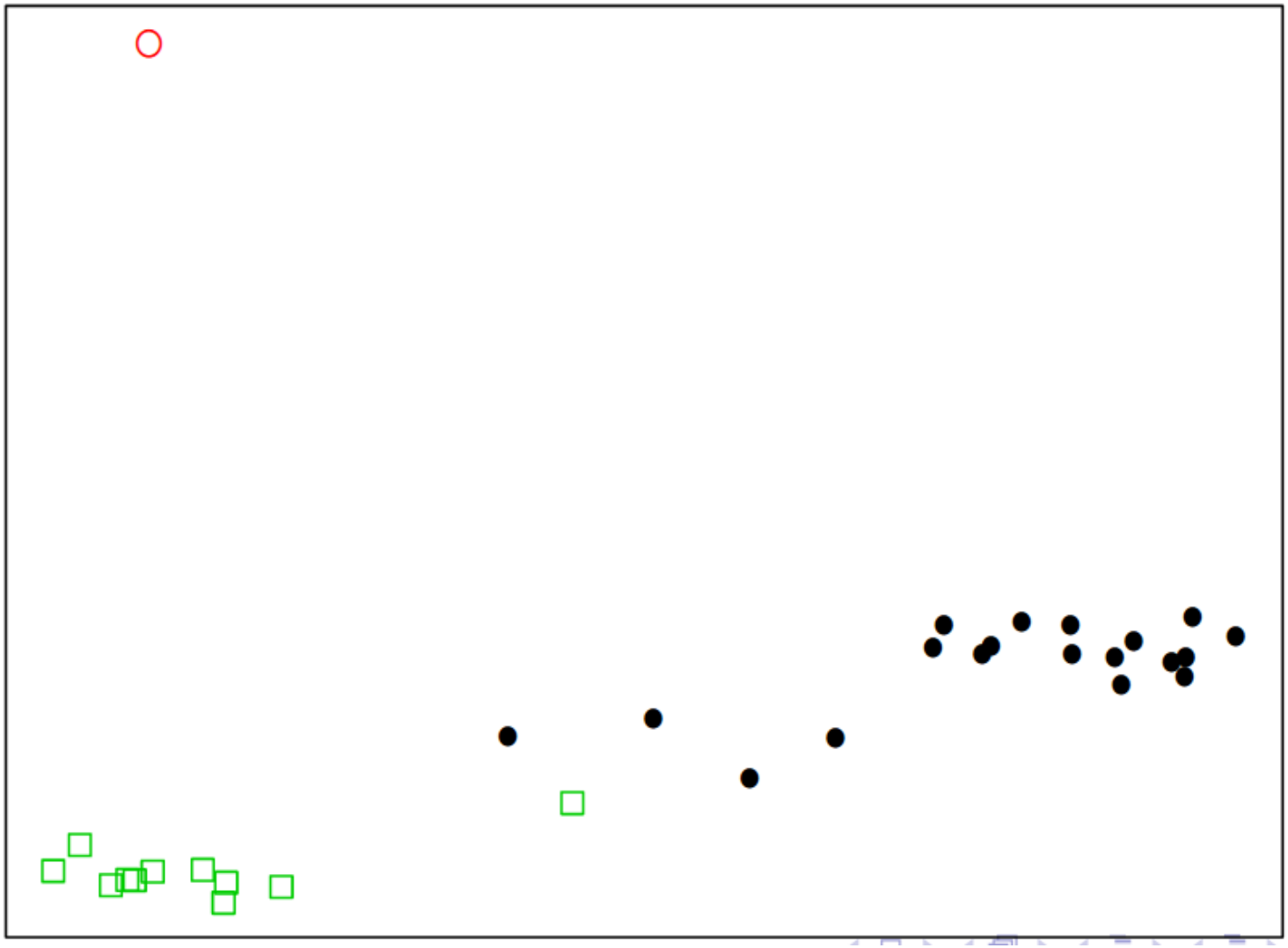
Limitations

Détection uniquement de formes sphériques



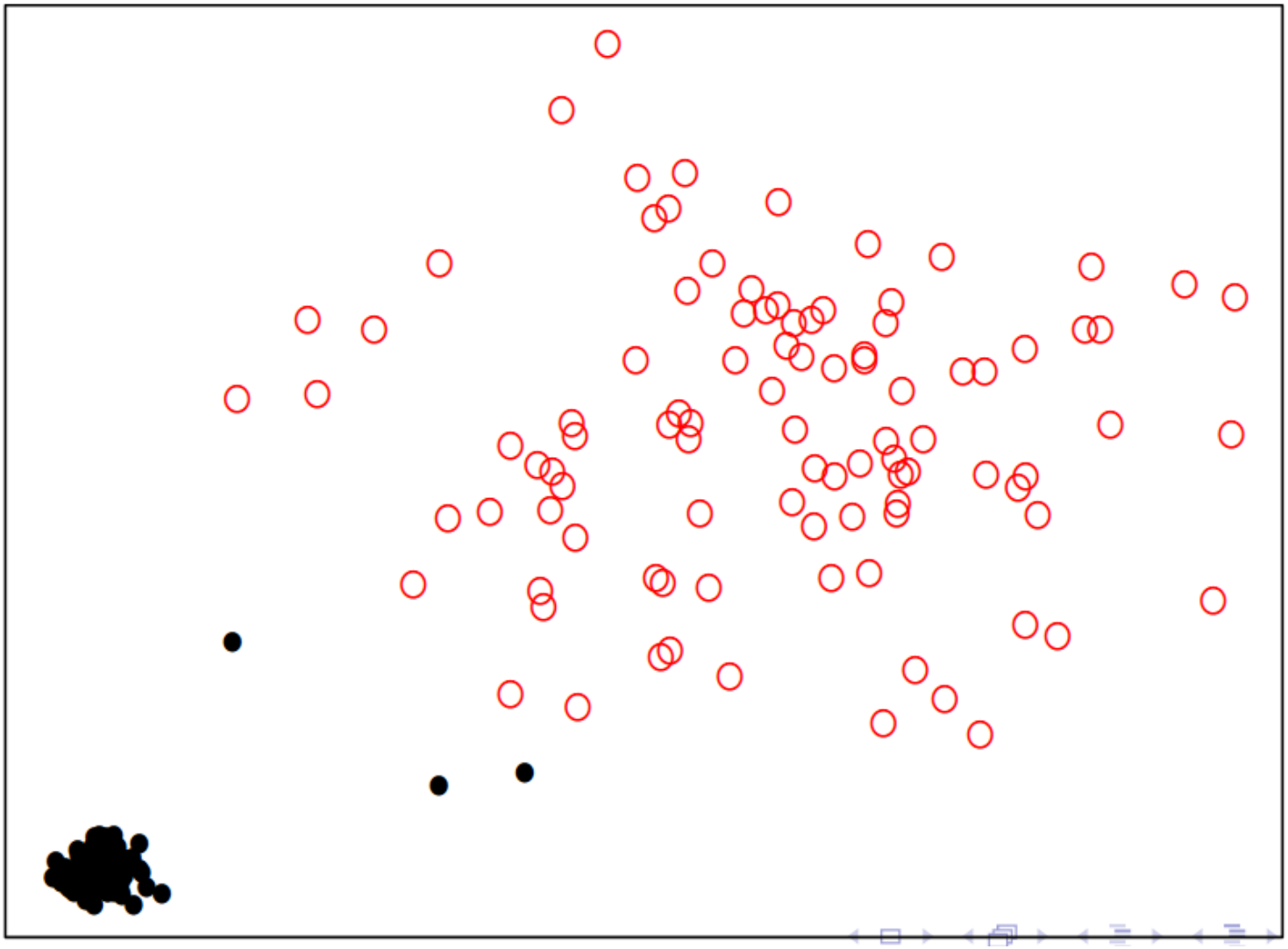
Limitations

Sensible aux données aberrantes



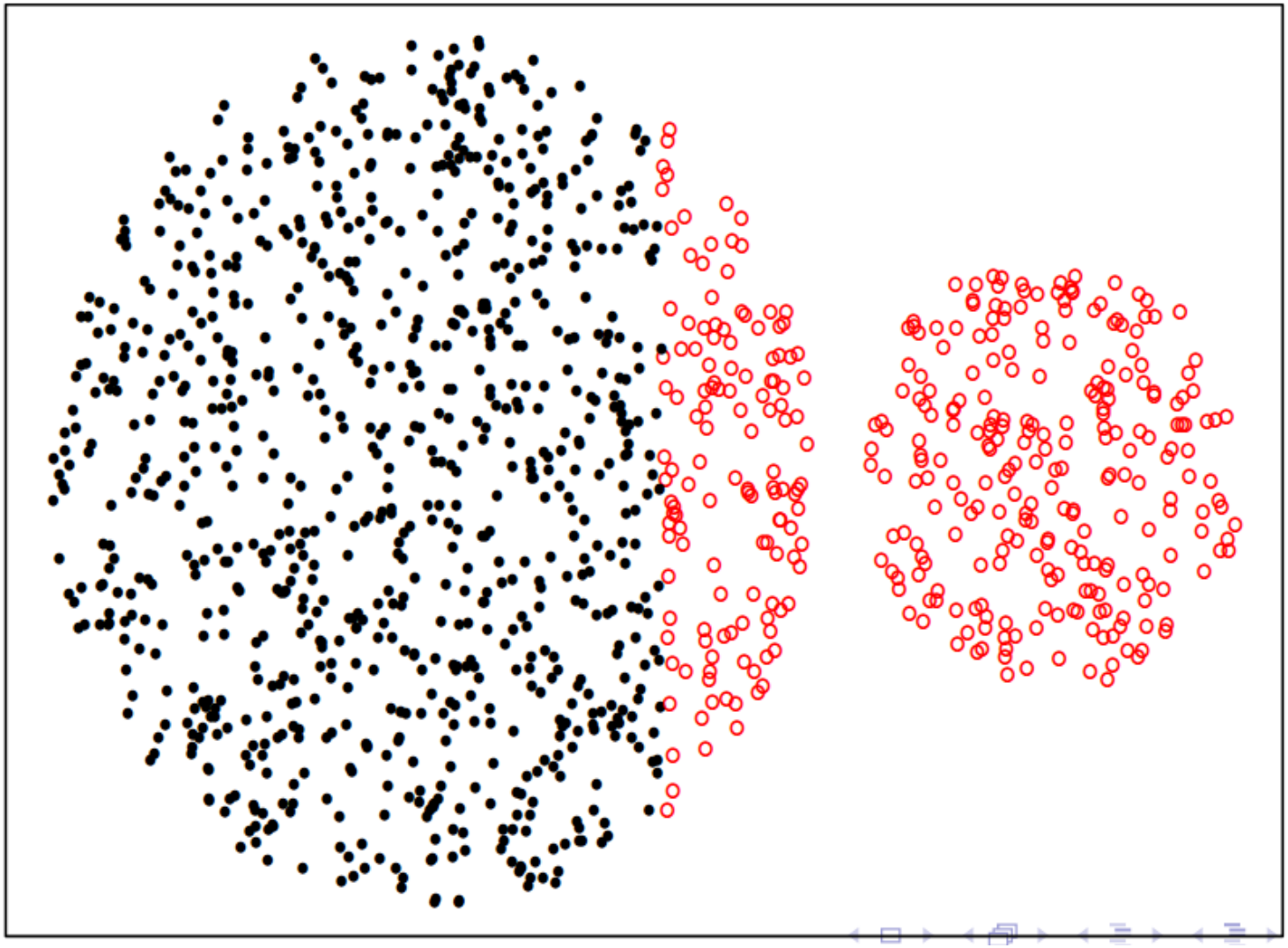
Limitations

Détection de clusters avec des densités proches



Limitations

Détection de clusters avec des tailles proches



Conclusions

Avantages

- Algorithme simple à mettre en oeuvre
- Utilisable pour m grand (mais pas trop grand)
- Complexité calculatoire en $O(k \cdot m \cdot d)$

Limitations

- le nombre de clusters k doit être fixé *a priori*
- détecte uniquement des formes sphériques
- des clusters peuvent être vides ou ne contenir que des données aberrantes
- ne marche pas bien si les clusters ont des densités différentes ou si ils sont de tailles différentes

Conclusions

Variantes

- pour des données non-quantitatives
- pour trouver plus rapidement une solution otpimale
- pour trouver des propriétés particulières recherchées

Activité 6

Comparez les résultats à ceux obtenus avec la librairie Scikit-Learn.