

01-02

Partitionnement en K-moyennes

420-A58-SF — Algorithmes d'apprentissage non supervisé — Hiver 2023
Spécialisation technique en intelligence artificielle — M. Swawola, M.Sc.

**NOUS ÉCLAIRON.
VOUS BRILLEZ.**

FORMATION CONTINUE
ET SERVICES AUX ENTREPRISES



Sommaire

1. K-moyennes: intuition
2. Algorithme des K-moyennes
3. Initialisation de l'algorithme
4. Choix du nombre de clusters
5. Mise à l'échelle
6. K-moyennes avec scikit-learn
7. Ateliers
8. Lectures et références

Sommaire

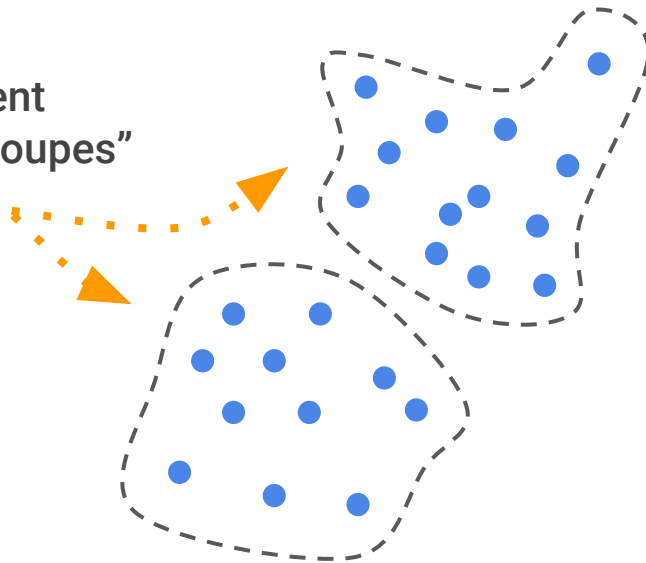
1. K-moyennes: intuition
2. Algorithme des K-moyennes
3. Initialisation de l'algorithme
4. Choix du nombre de clusters
5. Mise à l'échelle
6. K-moyennes avec scikit-learn
7. Ateliers
8. Lectures et références

Rappel des principaux types de partitionnement

- Partitionnement basé sur
 - les centroïdes (**K-moyennes**, CURE, ...) 🖱️
 - la connectivité (hiérarchique, ...)
 - la distribution (BFR, ...)
 - la densité (DBSCAN, OPTICS, ...)
 - les grilles
- Et d'autres

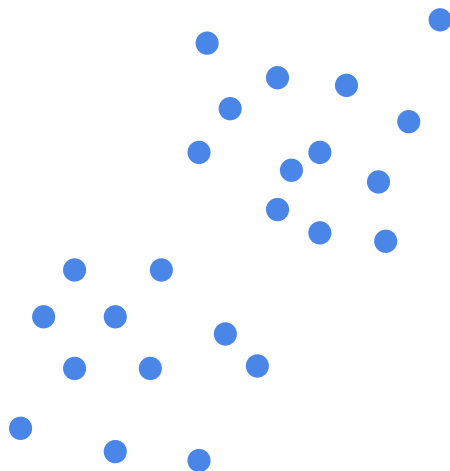
K-moyennes: intuition

Ces données semblent
partitionnées en deux “groupes”



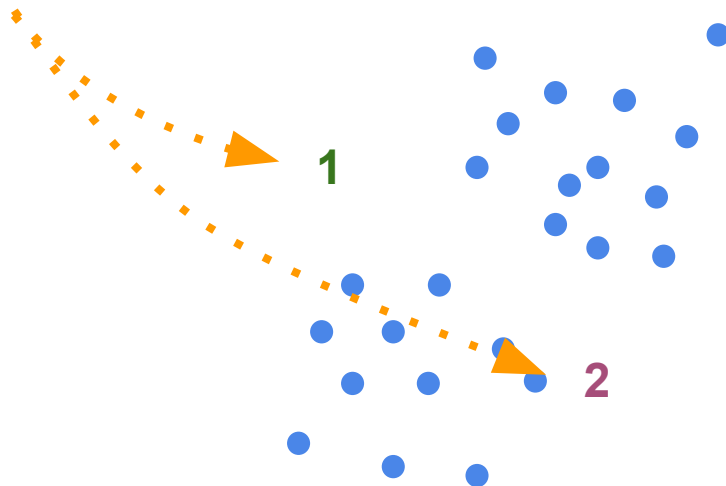
→ Comment découvrir ces partitions, ou clusters ?

K-moyennes: intuition



K-moyennes: intuition

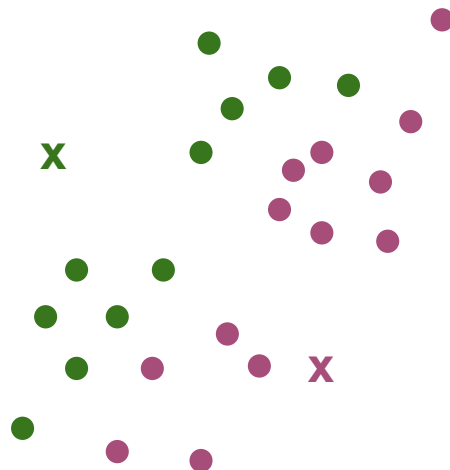
On place au hasard K centroïdes initiaux (ici $K = 2$)



K-moyennes: intuition

On place au hasard K centroïdes initiaux (ici $K = 2$)

On assigne les observations au centroïde le plus proche

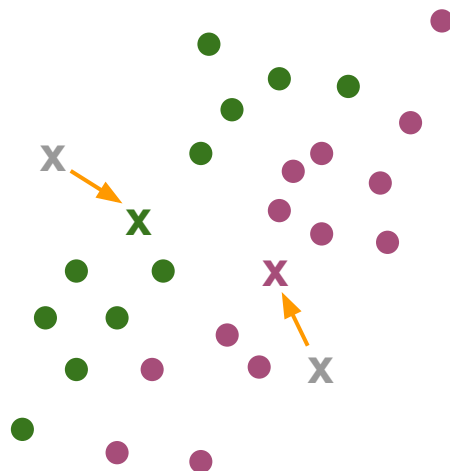


K-moyennes: intuition

On place au hasard K centroïdes initiaux (ici $K = 2$)

On assigne les observations au centroïde le plus proche

On déplace les centroïdes

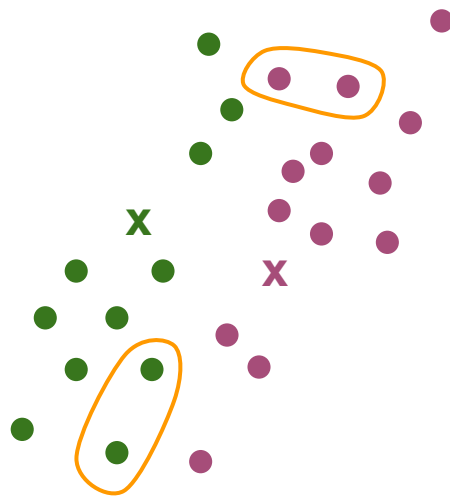


K-moyennes: intuition

On place au hasard K centroïdes initiaux (ici $K = 2$)

On assigne les observations au centroïde le plus proche

On déplace les centroïdes

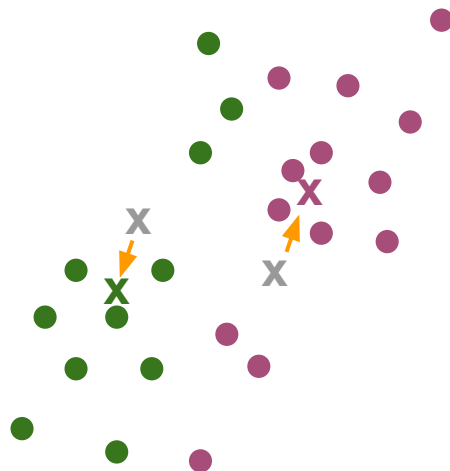


K-moyennes: intuition

On place au hasard K centroïdes initiaux (ici $K = 2$)

On assigne les observations au centroïde le plus proche

On déplace les centroïdes

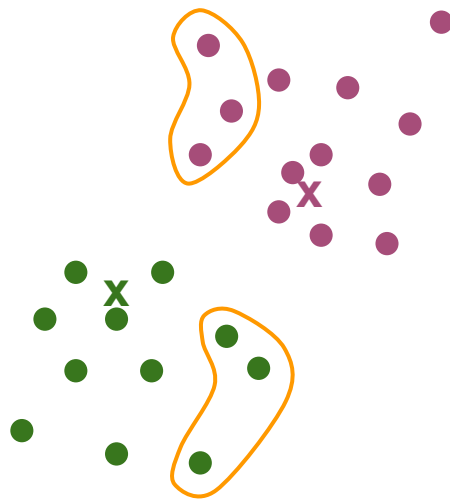


K-moyennes: intuition

On place au hasard K centroïdes initiaux (ici $K = 2$)

On assigne les observations au centroïde le plus proche

On déplace les centroïdes

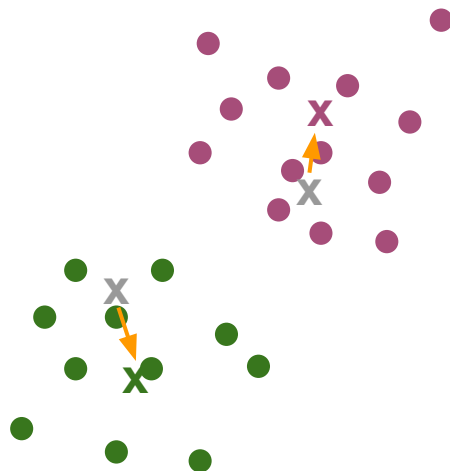


K-moyennes: intuition

On place au hasard K centroïdes initiaux (ici $K = 2$)

On assigne les observations au centroïde le plus proche

On déplace les centroïdes



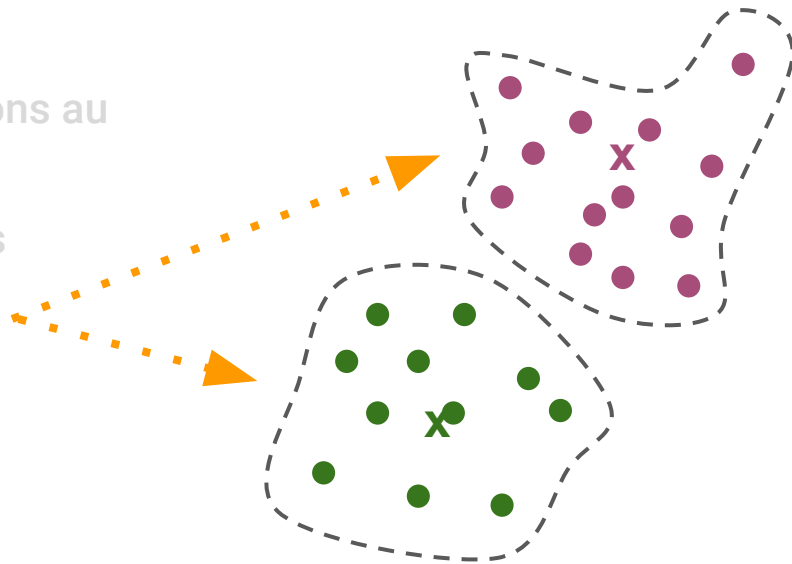
K-moyennes: intuition

On place au hasard K centroïdes initiaux (ici $K = 2$)

On assigne les observations au centroïde le plus proche

On déplace les centroïdes

On obtient deux clusters

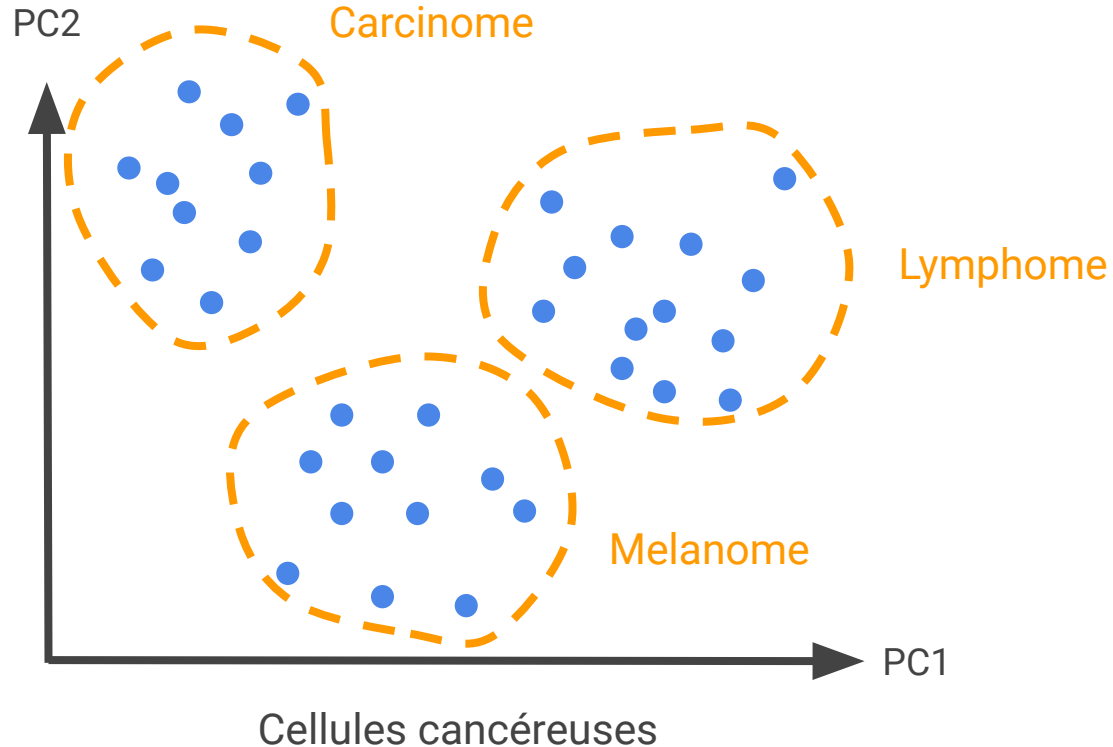


K-moyennes: intuition

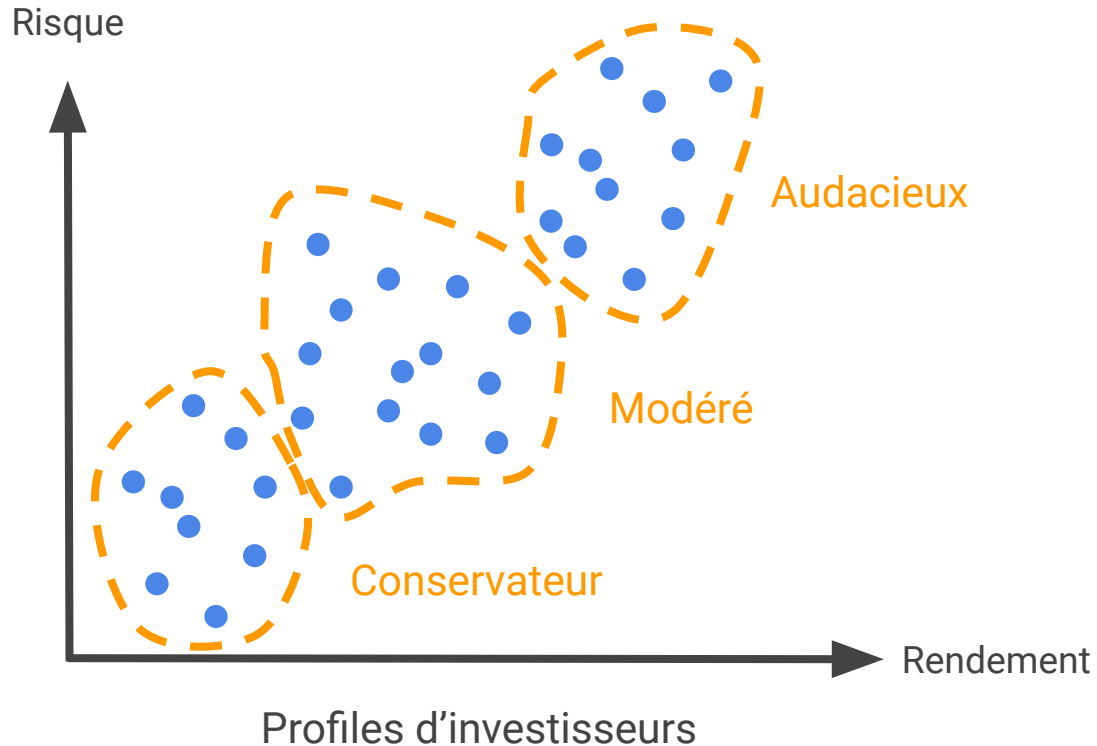
La procédure précédente se réduit à un problème mathématique simple et intuitif: en considérant K clusters, si C_1, \dots, C_K dénotent les ensembles contenant les indices des observations pour chaque cluster, alors les deux propriétés suivantes sont satisfaites:

- $C_1 \cup C_2 \cup \dots \cup C_K = \{ 1, \dots, m \} \rightarrow$ Chaque observation appartient à au moins un des K clusters
- $C_k \cap C_{k'} = \emptyset \ \forall \ k \neq k' \rightarrow$ Les clusters ne se chevauchent pas, car aucune observation n'appartient à plus d'un cluster

K-moyennes pour clusters séparés



K-moyennes pour clusters non séparés



Sommaire

1. K-moyennes: intuition
2. **Algorithme des K-moyennes**
3. Initialisation de l'algorithme
4. Choix du nombre de clusters
5. Mise à l'échelle
6. K-moyennes avec scikit-learn
7. Ateliers
8. Lectures et références

Ce que l'on cherche à optimiser

Soit les notations suivantes:

- $c^{(i)}$ index du cluster (1, 2, ..., K) auquel l'observation $x^{(i)}$ est assignée
- μ_k centroïde du cluster k
- $\mu_{c^{(i)}}$ centroïde du cluster auquel l'observation $x^{(i)}$ est assignée

L'objectif est de minimiser la **fonction de coût** ci-dessous

$$J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K) = \frac{1}{m} \sum_{i=1}^m \|x^{(i)} - \mu_{c^{(i)}}\|^2$$



Distance
euclidienne

En d'autres termes, la distance entre les observations d'un même cluster doit être petite, tandis que la distance entre les clusters doit être grande

Ce que l'on cherche à optimiser

Minimiser la fonction de coût

$$\min_{\substack{c^{(1)}, \dots, c^{(m)} \\ \mu_1, \dots, \mu_K}} J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K)$$

Est un problème **NP-hard** !



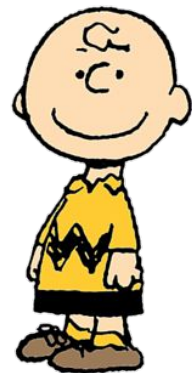
Nécessité de trouver une
solution approchée

Solution approchée

- **Algorithme de Lloyd (K-moyennes)**
- Aussi connu sous le nom d'itérations de Voronoï, il s'agit d'un algorithme itératif issu du génie électrique
- La fonction de coût

$$J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K) = \frac{1}{m} \sum_{i=1}^m \|x^{(i)} - \mu_{c^{(i)}}\|^2$$

est aussi appelée **distorsion**



Algorithme des K-moyennes: pseudo-code

Choix de K

Initialisation aléatoire de K centroïdes $\mu[1]$,
 $\mu[2]$, ..., $\mu[K]$

Répéter {

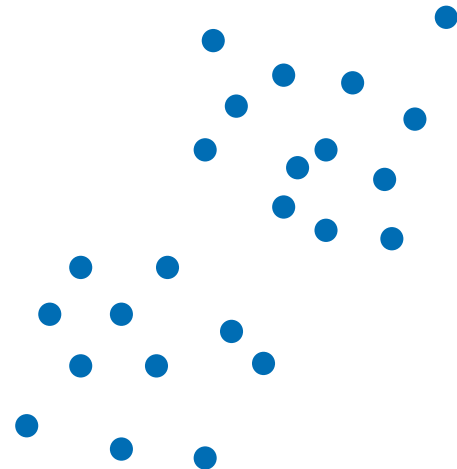
 for $i = 1$ to m

$c[i] :=$ index (de 1 à K) du centroïde le
 plus proche de $x[i]$

 for $k = 1$ to K

$\mu[k] :=$ moyenne des points assignés au
 cluster k

}



Algorithme des K-moyennes: pseudo-code

Choix de K

Initialisation aléatoire de K centroïdes $\mu[1]$,
 $\mu[2]$, ..., $\mu[K]$

Répéter {

 for $i = 1$ to m

$c[i] :=$ index (de 1 à K) du centroïde le
 plus proche de $x[i]$

 for $k = 1$ to K

$\mu[k] :=$ moyenne des points assignés au
 cluster k

}

$$\min_{c^{(1)}, \dots, c^{(m)}} J(c^{(1)}, \dots, c^{(m)}, \overbrace{\mu_1, \dots, \mu_K}^{\text{constant}})$$

$$\min_{\mu_1, \dots, \mu_K} J(\underbrace{c^{(1)}, \dots, c^{(m)}}_{\text{constant}}, \mu_1, \dots, \mu_K)$$

Sommaire

1. K-moyennes: intuition
2. Algorithme des K-moyennes
3. Initialisation de l'algorithme
4. Choix du nombre de clusters
5. Mise à l'échelle
6. K-moyennes avec scikit-learn
7. Ateliers
8. Lectures et références

Initialisation de l'algorithme

Choix de K

Initialisation aléatoire de K centroïdes $\mu[1]$, $\mu[2]$, ..., $\mu[K]$

Répéter {

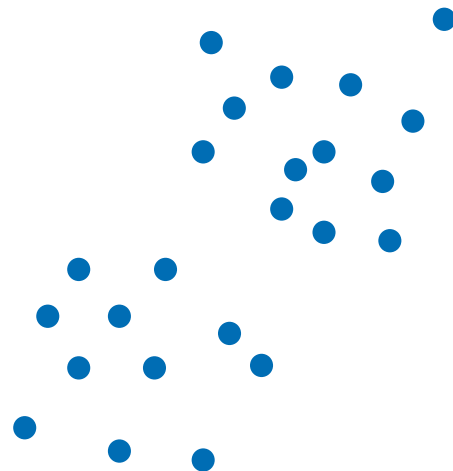
 for $i = 1$ to m

$c[i] :=$ index (de 1 à K) du centroïde le plus proche de $x[i]$

 for $k = 1$ to K

$\mu[k] :=$ moyenne des points assignés au cluster k

}



Initialisation de l'algorithme

Choix de K

Initialisation aléatoire de K centroïdes $\mu[1]$, $\mu[2]$, ..., $\mu[K]$

Répéter {

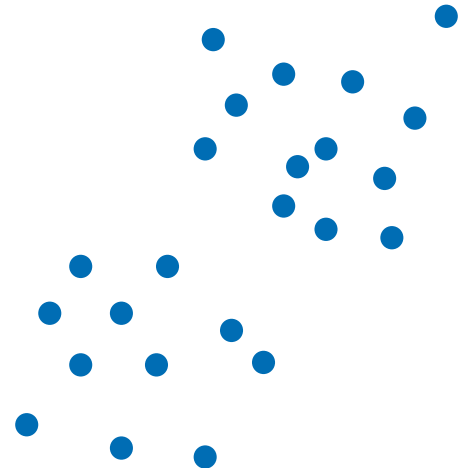
 for $i = 1$ to m

$c[i] :=$ index (de 1 à K) du centroïde le plus proche de $x[i]$

 for $k = 1$ to K

$\mu[k] :=$ moyenne des points assignés au cluster k

}



Initialisation de l'algorithme

Choix de K

Initialisation aléatoire de K centroïdes $\mu[1]$, $\mu[2]$, ..., $\mu[K]$

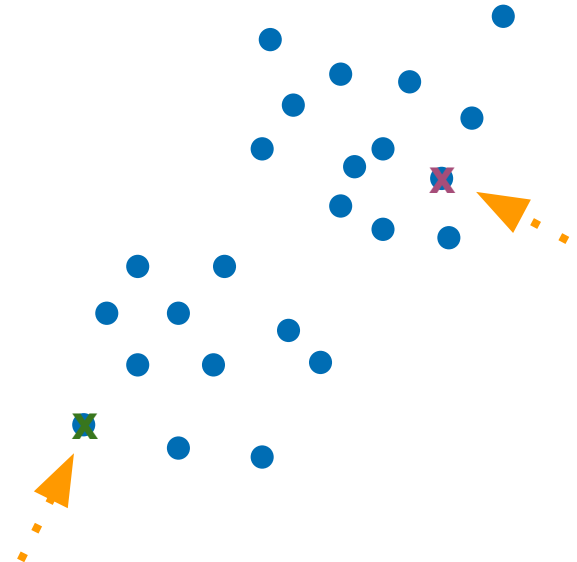
Répéter {

for $i = 1$ to m

- K doit être inférieur à m (nombre d'observations)

- Choisir au hasard K observations

- Les K centroïdes $\mu_1, \mu_2, \dots, \mu_K$ sont égaux à ces observations



Initialisation de l'algorithme

Choix de K

Initialisation aléatoire de K centroïdes $\mu[1]$, $\mu[2]$, ..., $\mu[K]$

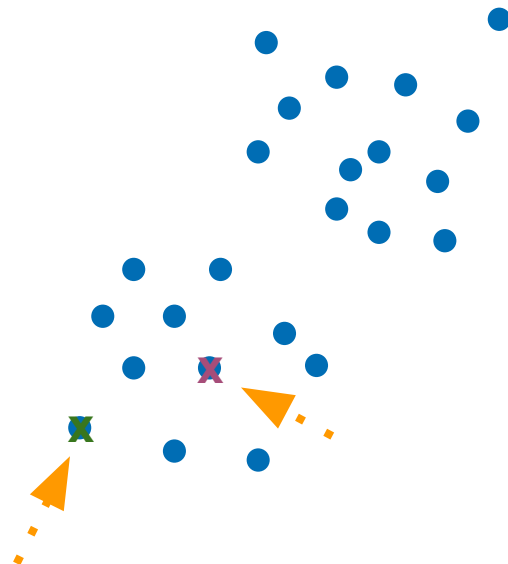
Répéter {

for $i = 1$ to m

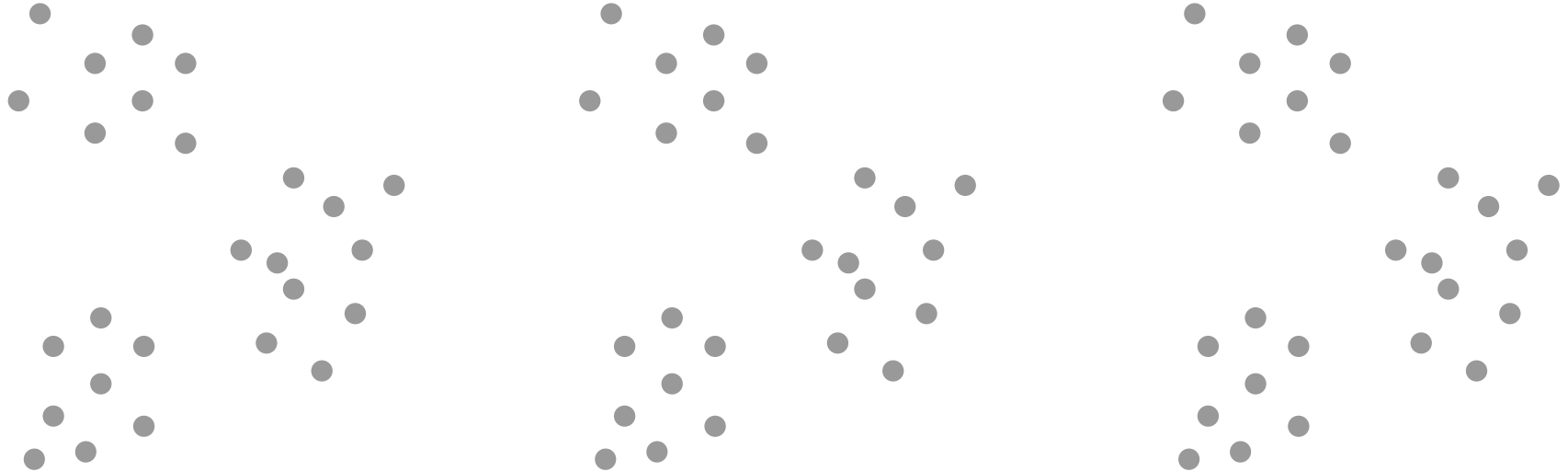
- K doit être inférieur à m (nombre d'observations)

- Choisir au hasard K observations

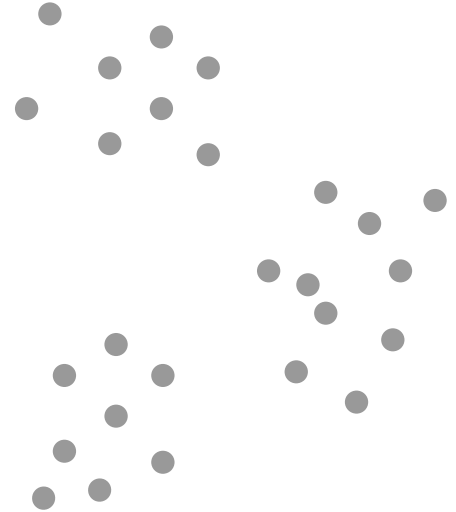
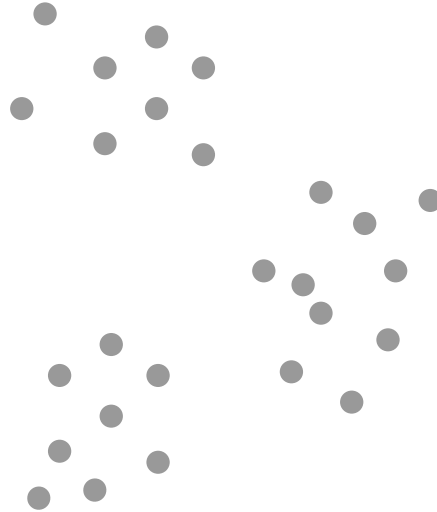
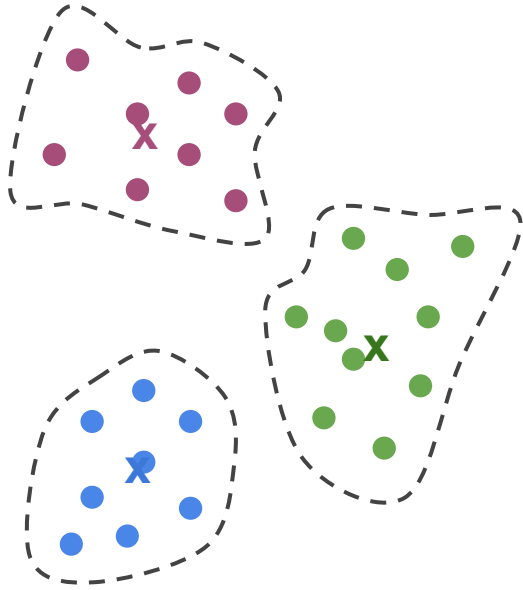
- Les K centroïdes $\mu_1, \mu_2, \dots, \mu_K$ sont égaux à ces observations



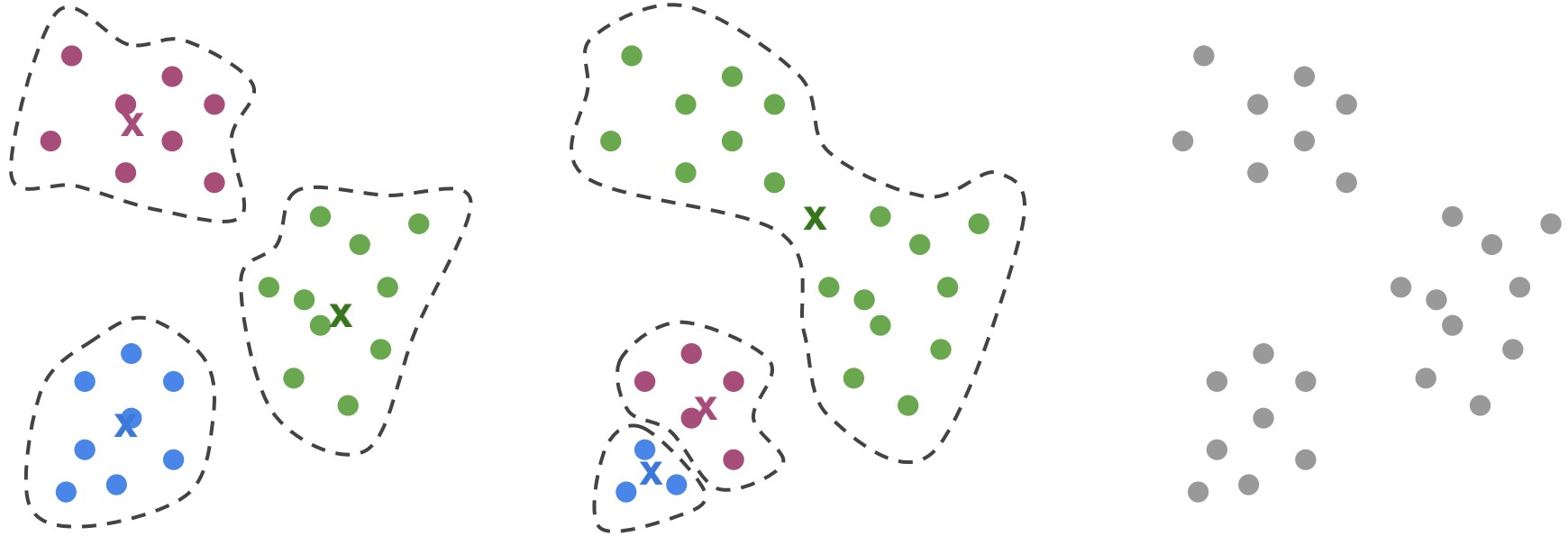
Algorithme des K-moyennes - optimum local



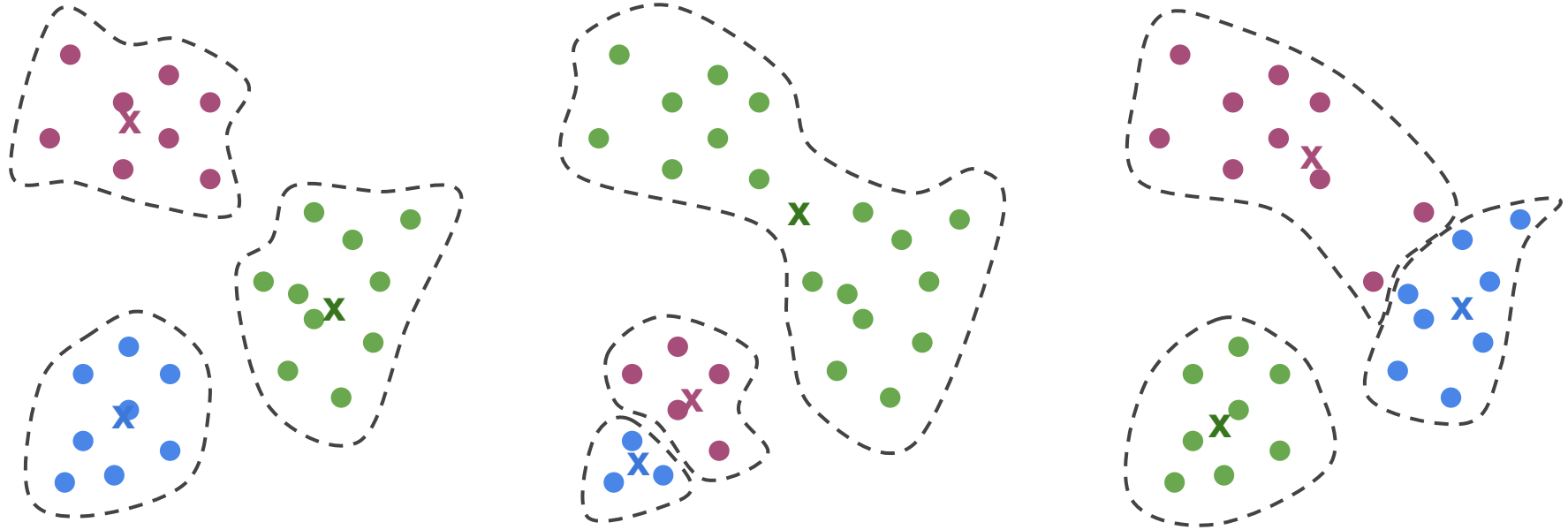
Algorithme des K-moyennes - optimum local



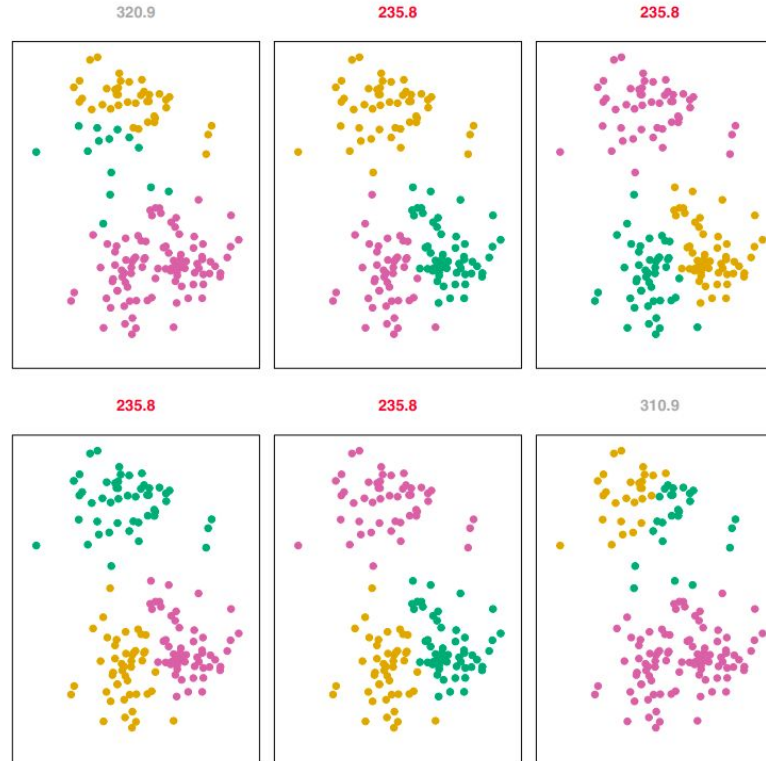
Algorithme des K-moyennes - optimum local



Algorithme des K-moyennes - optimum local



Algorithme des K-moyennes - optimum local



Extrait de [2] - Ici l'algorithme des K-moyennes est exécuté 6 fois avec $K = 3$

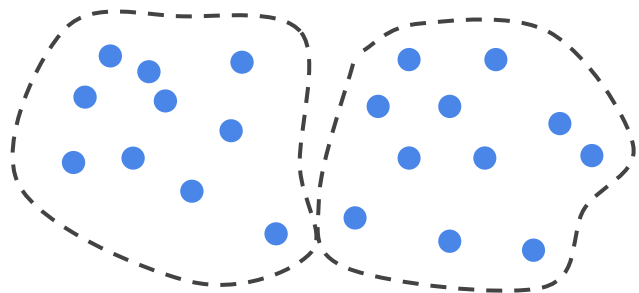
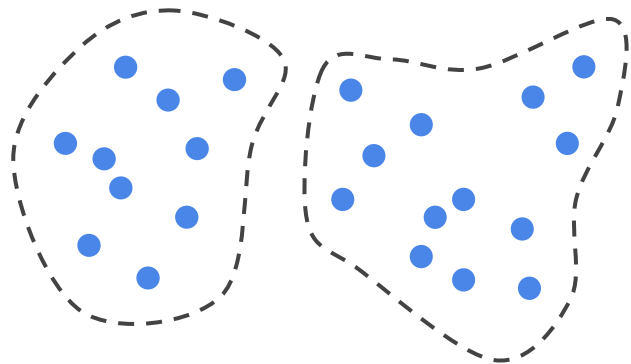
Algorithme des K-moyennes - exécutions multiples

```
for i = 1 to n_init {  
    Initialisation aléatoire des K centroïdes  $\mu[1]$ ,  $\mu[2]$ , ...,  $\mu[K]$   
    Exécution de l'algorithme des K-moyennes  
    Sauvegarder dans un historique  $\mu[1]$ ,  $\mu[2]$ , ...,  $\mu[K]$   
    Calculer la distorsion  $J$   
}  
Choisir le résultat donnant la distorsion la plus faible
```

Sommaire

1. K-moyennes: intuition
2. Algorithme des K-moyennes
3. Initialisation de l'algorithme
4. Choix du nombre de clusters
5. Mise à l'échelle
6. K-moyennes avec scikit-learn
7. Ateliers
8. Lectures et références

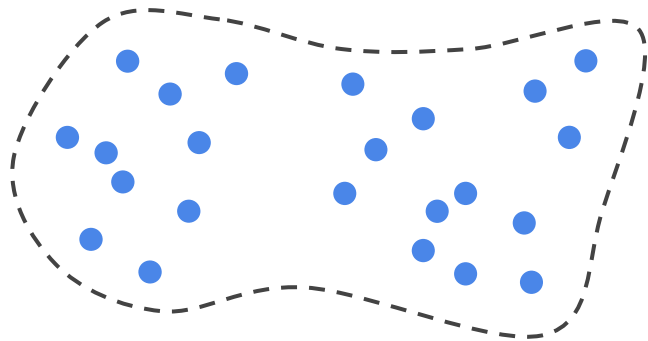
Le choix de K est souvent subjectif ...



Combien de clusters voyez vous ?

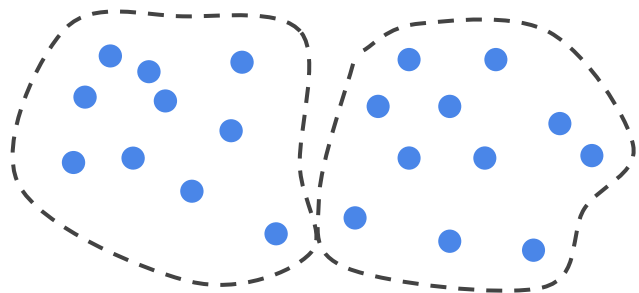
4 ?

Le choix de K est souvent subjectif ...

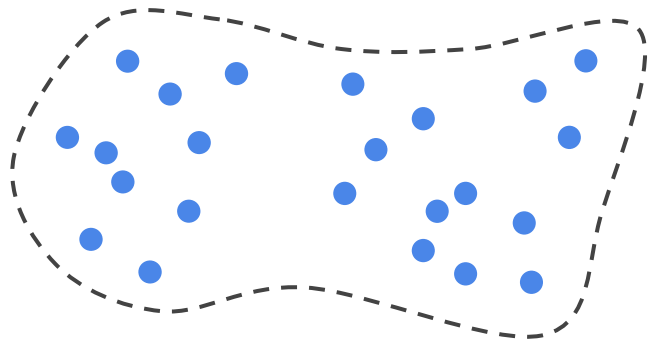


Combien de clusters voyez vous ?

3 ?

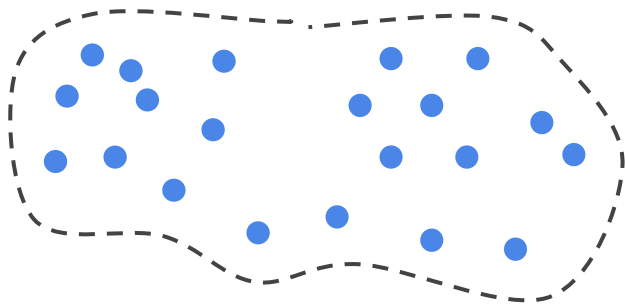


Le choix de K est souvent subjectif ...

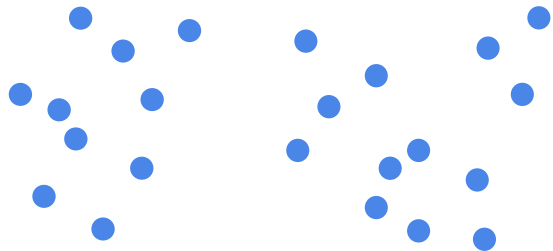


Combien de clusters voyez vous ?

2 ?

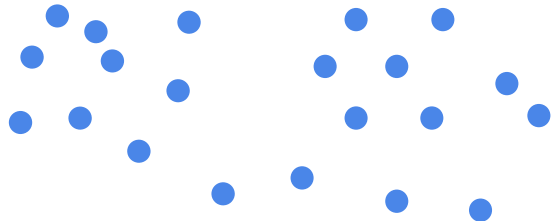


Le choix de K est souvent subjectif ...



Combien de clusters voyez vous ?

Toutes les réponses peuvent être valables !

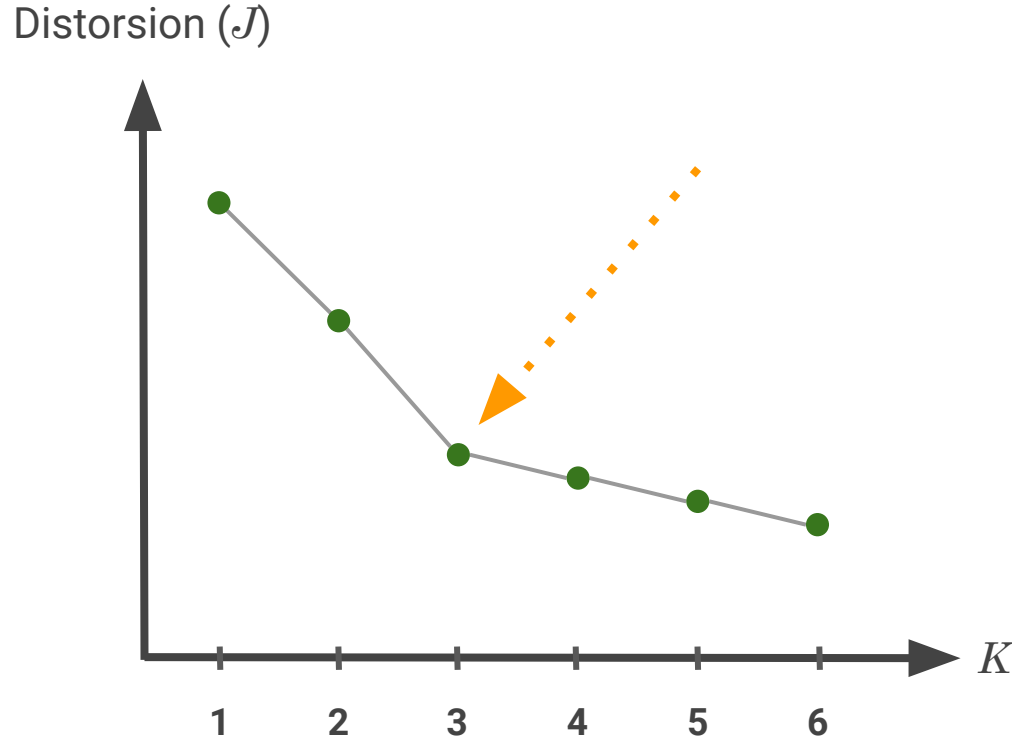


Le choix de K est souvent subjectif ...

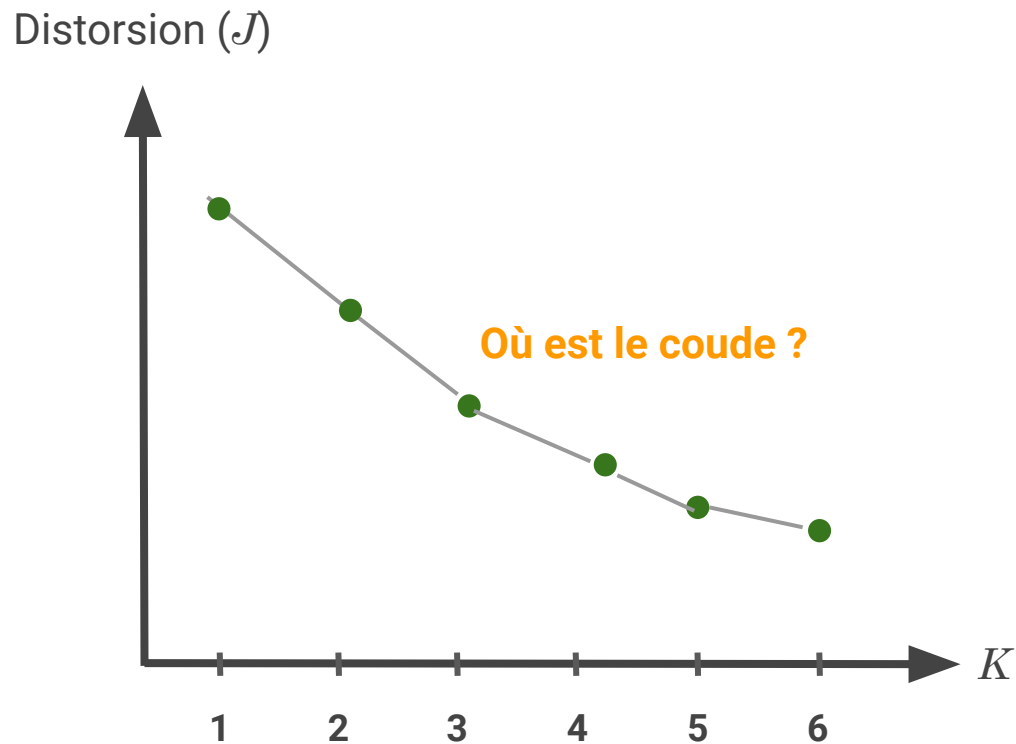
Il n'existe toujours pas de méthode "miracle" et la meilleure reste un choix... manuel !

Combien de clusters voyez vous ?
Il existe néanmoins quelques techniques pouvant fonctionner (parfois) ...
Toutes les réponses peuvent être valables !

La méthode du “coude” ... en théorie



Et en pratique !



Algorithme des K-moyennes - “GridSearch”

for k = 1 to K { ◀ **Exemple: 2 à 15**

 for i = 1 to n_iter { ◀ **Exemple: 1 à 1000**

 Initialisation aléatoire des k centroïdes $\mu[1]$, $\mu[2]$, ..., $\mu[K]$

 Exécution de l'algorithme des K-moyennes

 Sauvegarder dans un historique $\mu[1]$, $\mu[2]$, ..., $\mu[K]$

 Calculer la distorsion J

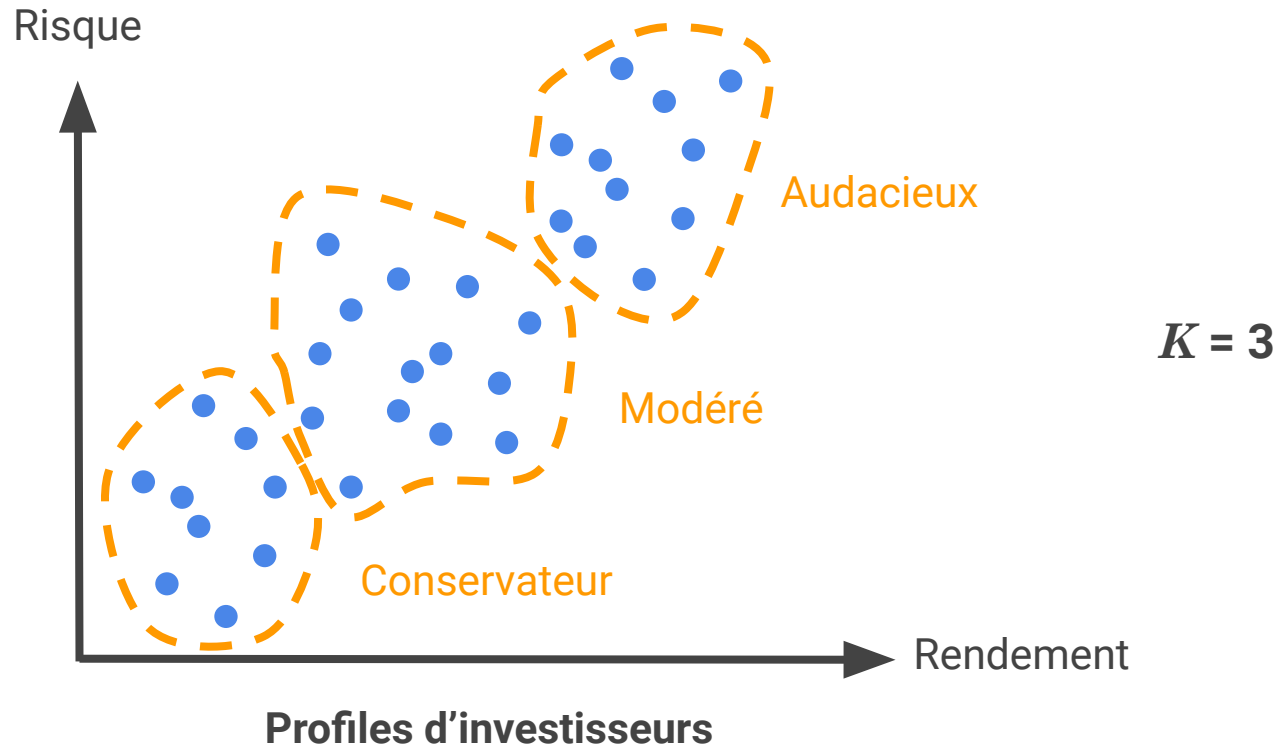
 }

}

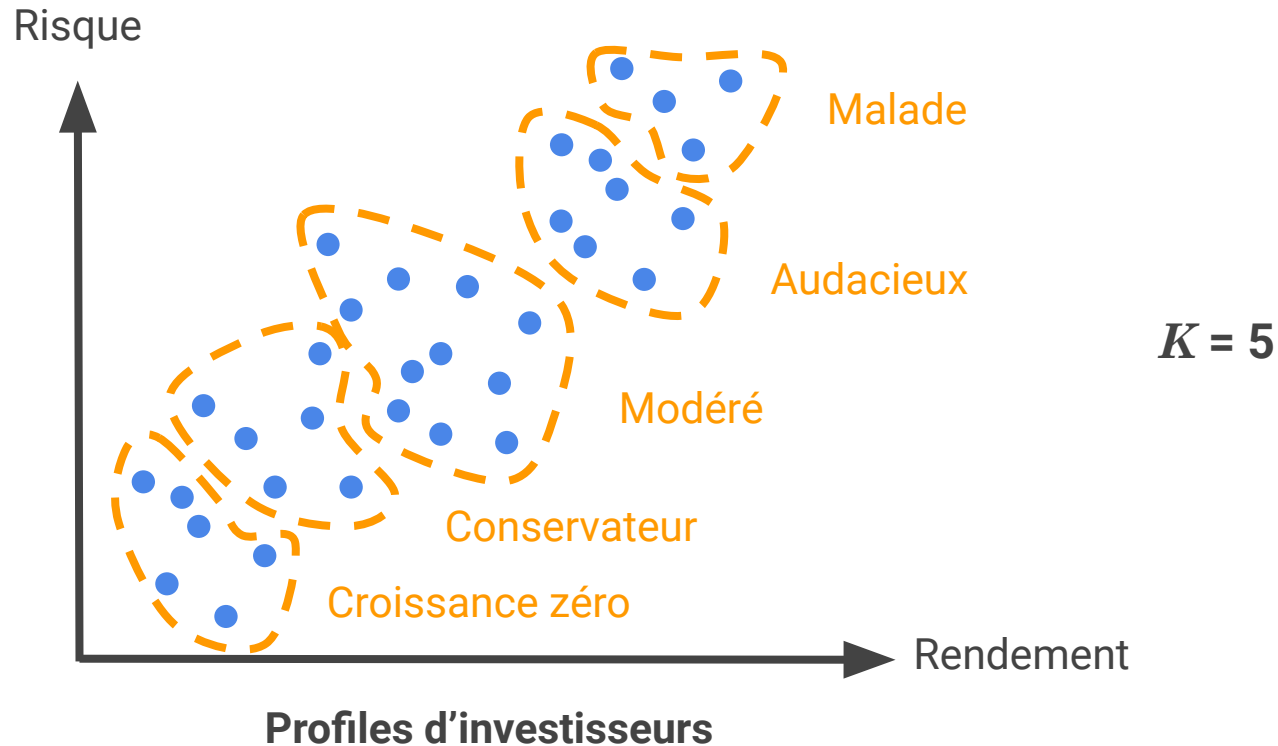
Choisir la valeur de K donnant la distorsion la plus faible



Choix de K à posteriori



Choix de K à posteriori



Sommaire

1. K-moyennes: intuition
2. Algorithme des K-moyennes
3. Initialisation de l'algorithme
4. Choix du nombre de clusters
5. **Mise à l'échelle**
6. K-moyennes avec scikit-learn
7. Ateliers
8. Lectures et références

Mise à l'échelle

Choix de K

Initialisation aléatoire de K centroïdes $\mu[1]$, $\mu[2]$, ..., $\mu[K]$

Répéter {

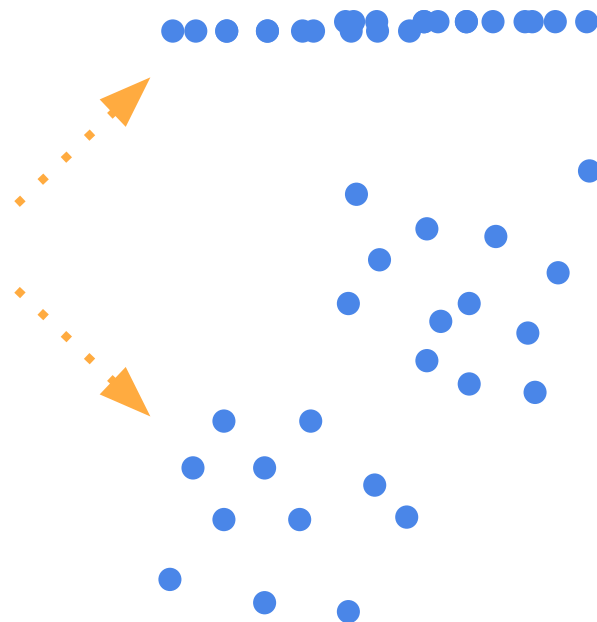
 for $i = 1$ to m

$c[i] := \text{index (de 1 à K) du centroïde le plus proche de } x[i]$

 for $k = 1$ to K

$\mu[k] := \text{moyenne des points assignés au cluster } k$

}



Mise à l'échelle

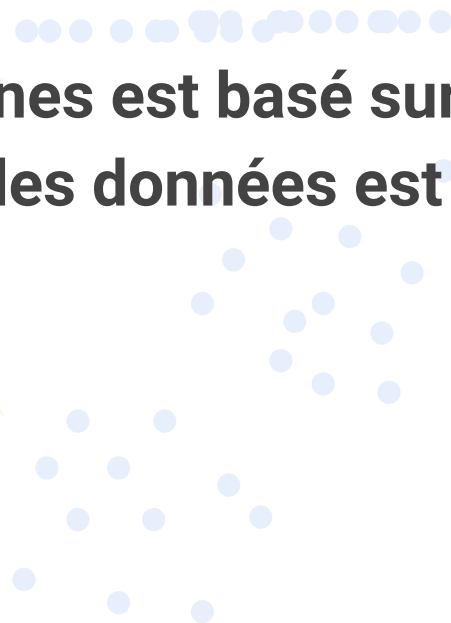
Choix de K

Initialisation aléatoire de K centroïdes $\mu[1]$,
 $\mu[2]$, ..., $\mu[K]$

Étant donné que l'algorithme des K-moyennes est basé sur le calcul de distances, la mise à l'échelle des données est très recommandée !

for $i = 1$ to n
 $c[i] := \text{index (de } 1 \text{ à } K) \text{ du centroïde le plus proche de } x[i]$

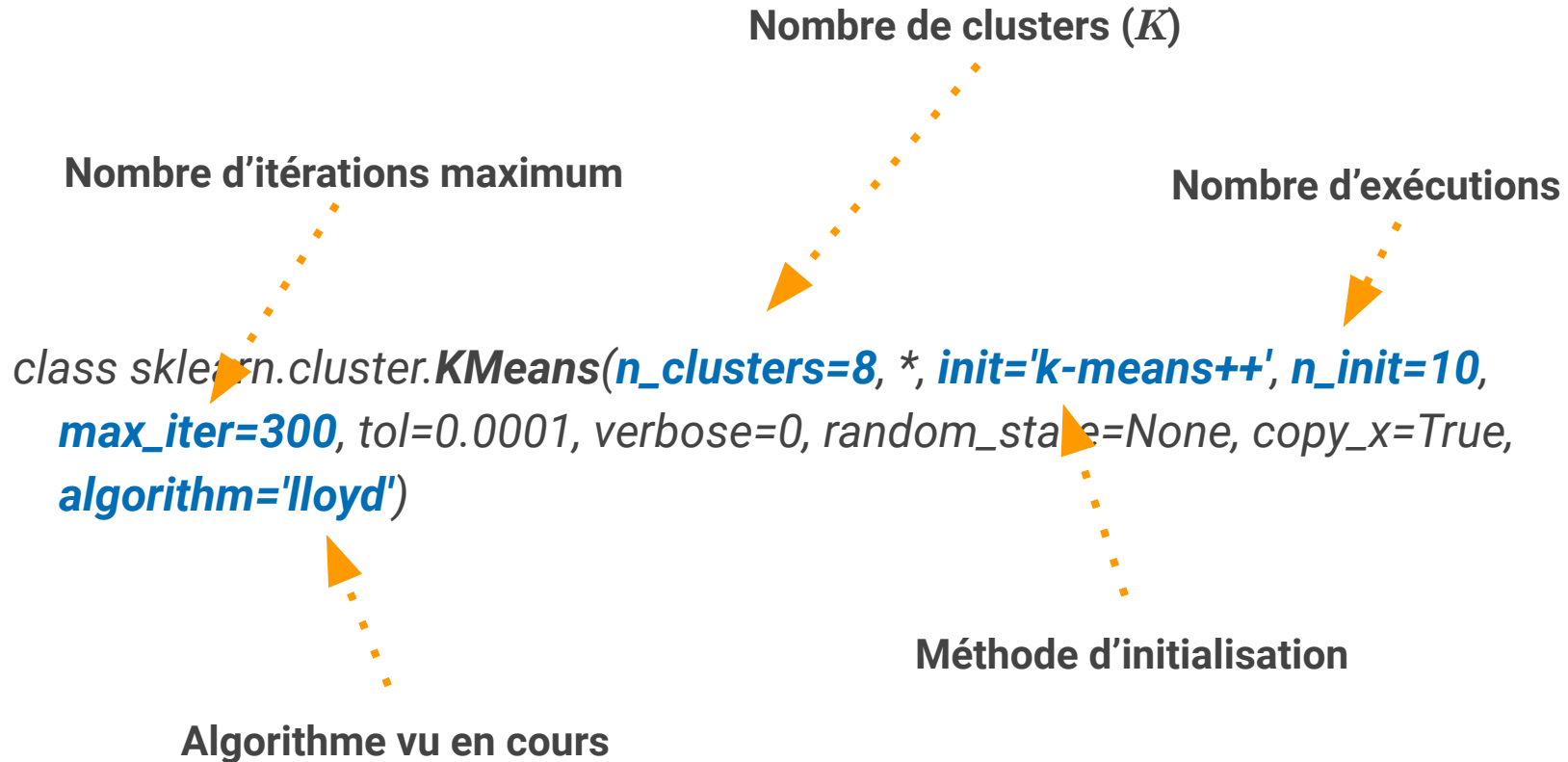
for $k = 1$ to K
 $\mu[k] := \text{moyenne des points assignés au cluster } k$
}



Sommaire

1. K-moyennes: intuition
2. Algorithme des K-moyennes
3. Initialisation de l'algorithme
4. Choix du nombre de clusters
5. Mise à l'échelle
6. K-moyennes avec scikit-learn
7. Ateliers
8. Lectures et références

KMeans (scikit-learn 1.2.1)



Sommaire

1. K-moyennes: intuition
2. Algorithme des K-moyennes
3. Initialisation de l'algorithme
4. Choix du nombre de clusters
5. Mise à l'échelle
6. K-moyennes avec scikit-learn
7. Ateliers
8. Lectures et références



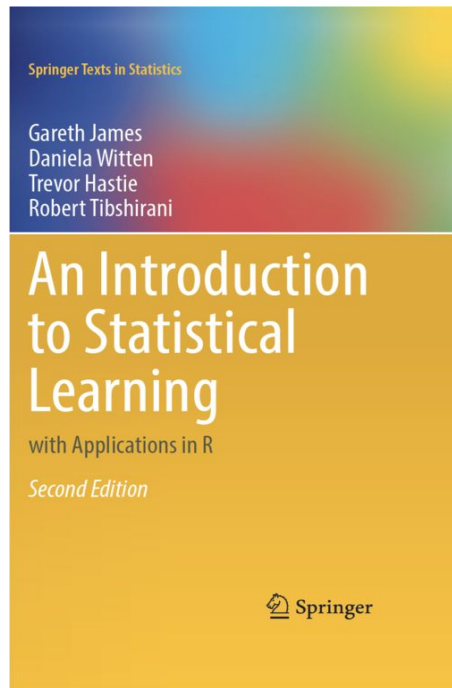
<https://github.com/mswawola-cegep/420-a58-sf-gr-12060.git>

01-02-A1 et 01-02-A2

Sommaire

1. K-moyennes: intuition
2. Algorithme des K-moyennes
3. Initialisation de l'algorithme
4. Choix du nombre de clusters
5. Mise à l'échelle
6. K-moyennes avec scikit-learn
7. Ateliers
8. Lectures et références

Lectures



- Introduction to Statistical Learning with Applications in R
Second edition (2021)
→ 12.4 Clustering Methods

Références

[1] CS229: Machine Learning - Stanford University

[2] Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani, "Introduction to Statistical Learning with Applications in R - Second edition"