Data Science

M2 Actuariat

Session 6 : Introduction à l'apprentissage non supervisé - Découvrir la structure cachée des données

François HU

Responsable du pôle Intelligence Artificielle, Milliman R&D

Enseignant ISFA

2025

ISFA
Institut de Science Financière et d'Assurances
GRANDE ÉCOLE D'ACTUARIAT & GESTION DES RISQUES

Milliman

Sommaire du cours Data Science

Session apprentissage non supervisé – Zoom sur le clustering

1. Fondements et clustering par partitionnement
2. Clustering avancé : hiérarchie et densité
3. Approche probabiliste et les modèles à variables latentes
4. Conclusion

Objectif de la session

- **Maîtriser les 3 grandes familles du non supervisé** :
  clustering, réduction de dimension, détection d'anomalies.

- **Comprendre les algorithmes clés** :
  k-means, clustering hiérarchique, DBSCAN, GMM...

- **Savoir appliquer ces techniques pour résoudre des problèmes actuariels concrets**

1. Fondements et clustering par partitionnement

- **Introduction au non supervisé**
- Préparation des données pour le clustering
- Algorithme k-means et son évaluation

ISFA
Institut de Science Financière et d'Assurances

Le paradigme : apprendre sans variable cible (1/3)

De la prédiction d'une cible à la découverte d'une structure

Exemple illustratif

Observations :

Variable cible : $Y = \{\circ, \circ\}$ (décès)

Apprentissage Supervisé (Rappel) :

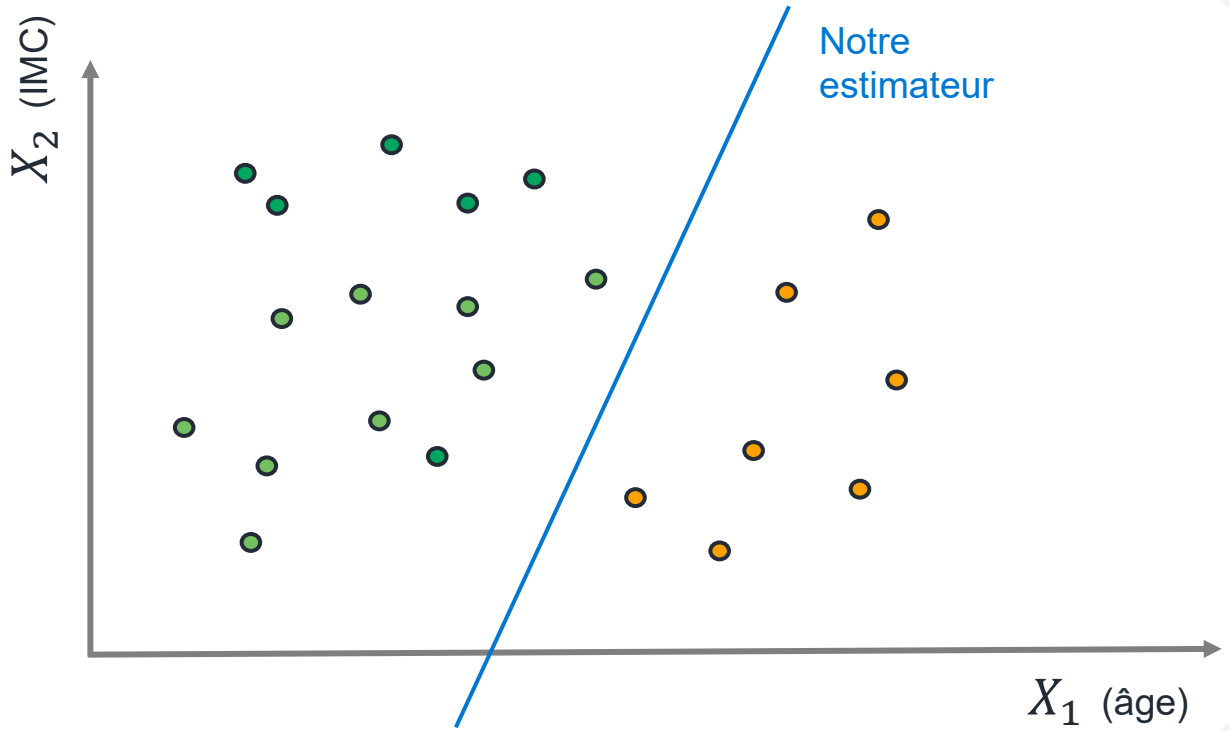
- **Données** : $\{(x_1, y_1), \dots, (x_n, y_n)\}$.
- **Objectif** : Estimer la relation $P(y|X)$ pour prédire y à partir de X.

Apprentissage Non Supervisé :

- **Données** : $\{x_1, \dots, x_n\}$.
- **Objectif** : Découvrir des propriétés intéressantes de la distribution $P(X)$ elle-même.

Les 3 grandes questions du non supervisé :

- **Clustering** : *Mes données forment-elles des groupes distincts ?*
- **Réduction de Dimension** : *Puis-je résumer mes données avec moins de variables ?*
- **Détection d'Anomalies** : *Certaines de mes observations sont-elles "étranges" ?*



Le paradigme : apprendre sans variable cible (2/3)

De la prédiction d'une cible à la découverte d'une structure

Exemple illustratif

Apprentissage Supervisé (Rappel) :

- **Données** : $\{(x_1, y_1), \dots, (x_n, y_n)\}$.
- **Objectif** : Estimer la relation $P(y|X)$ pour prédire y à partir de X.

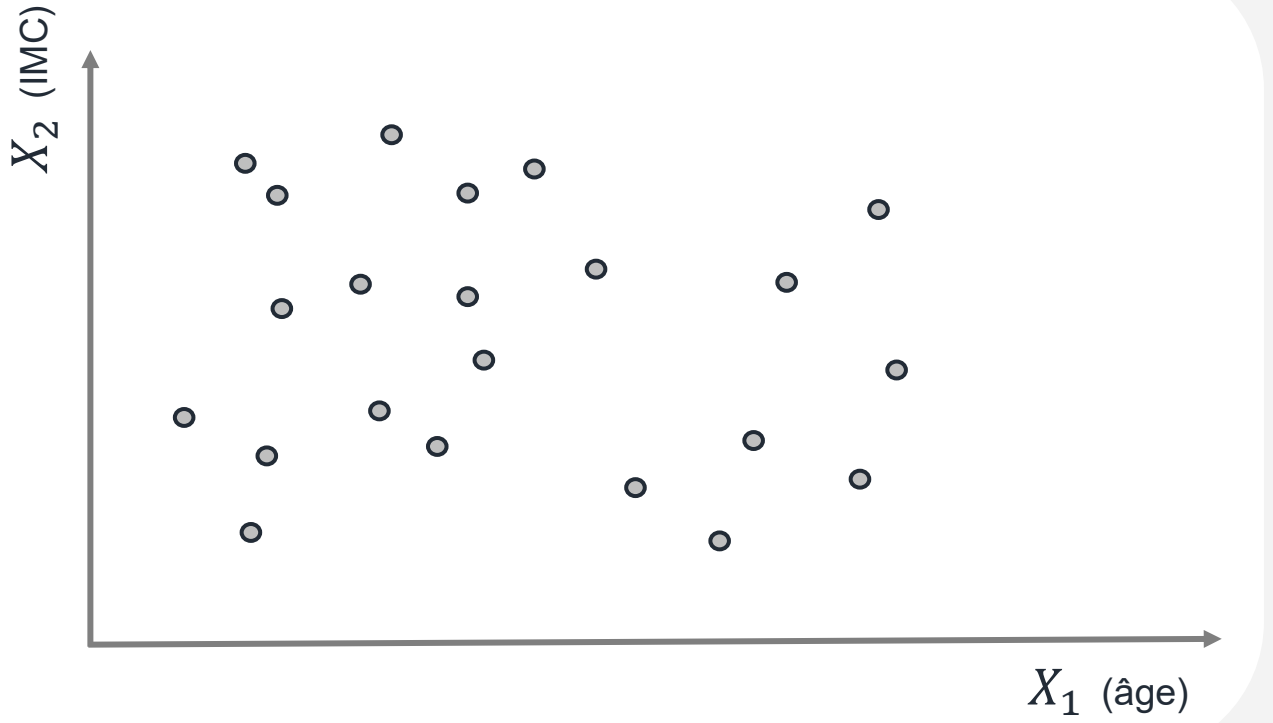
Apprentissage Non Supervisé :

- **Données** : $\{x_1, \dots, x_n\}$.
- **Objectif** : Découvrir des propriétés intéressantes de la distribution $P(X)$ elle-même.

Les 3 grandes questions du non supervisé :

- **Clustering** : *Mes données forment-elles des groupes distincts ?*
- **Réduction de Dimension** : *Puis-je résumer mes données avec moins de variables ?*
- **Détection d'Anomalies** : *Certaines de mes observations sont-elles "étranges" ?*

Observations :



Le paradigme : apprendre sans variable cible (3/3)

De la prédiction d'une cible à la découverte d'une structure

Exemple illustratif

Apprentissage Supervisé (Rappel) :

- **Données** : $\{(x_1, y_1), \dots, (x_n, y_n)\}$.
- **Objectif** : Estimer la relation $P(y|X)$ pour prédire y à partir de X.

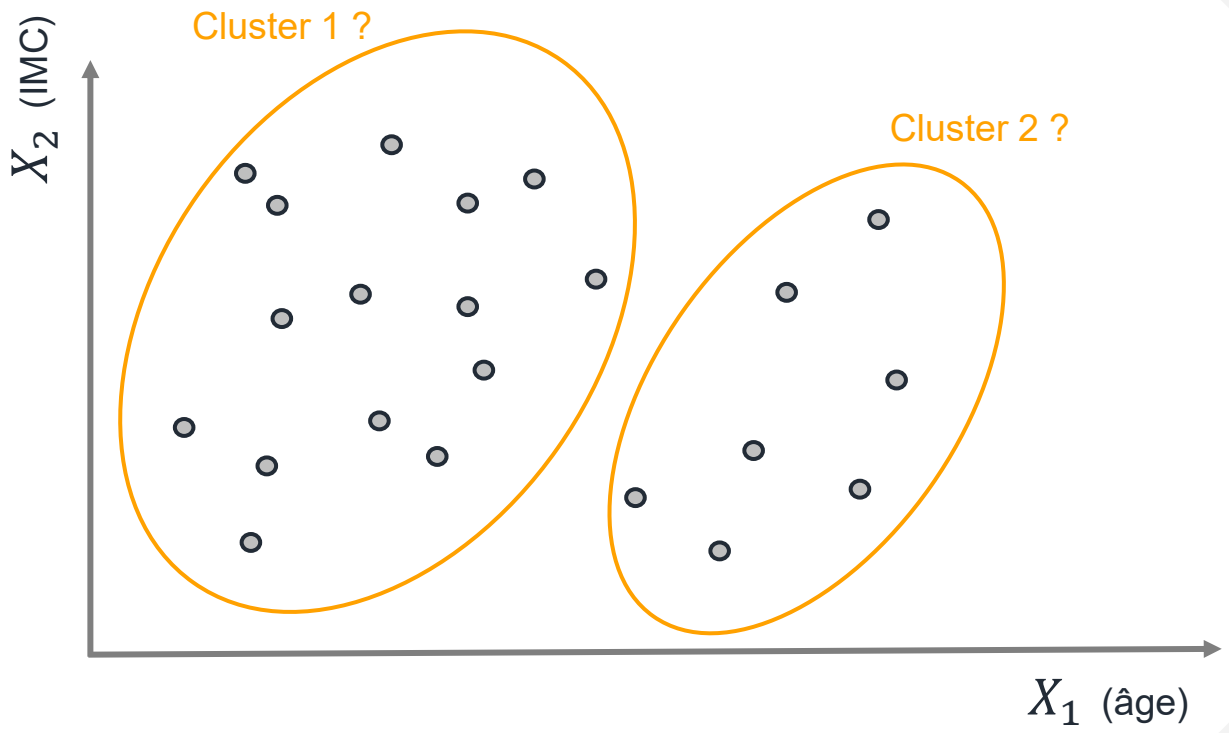
Apprentissage Non Supervisé :

- **Données** : $\{x_1, \dots, x_n\}$.
- **Objectif** : Découvrir des propriétés intéressantes de la distribution $P(X)$ elle-même.

Les 3 grandes questions du non supervisé :

- **Clustering** : *Mes données forment-elles des groupes distincts ?*
- **Réduction de Dimension** : *Puis-je résumer mes données avec moins de variables ?*
- **Détection d'Anomalies** : *Certaines de mes observations sont-elles "étranges" ?*

Observations :



Applications actuarielles du non supervisé

Créer de la valeur par l'exploration et la segmentation

Segmentation de portefeuille (Clustering)

- **Problème** : Identifier des profils de clients homogènes pour personnaliser les offres marketing ou les actions de prévention.
- **Exemple** : Découvrir un segment de "jeunes conducteurs prudents en milieu rural" qui méritent un tarif plus bas que la moyenne de leur classe d'âge.
- **Objectif : Créer des segments de clients ("personas")** pertinents et actionnables pour le marketing ou la tarification.

Typologie des sinistres (Clustering)

- **Problème** : Créer des groupes de sinistres (ex: "fraude probable", "corporel complexe", "matériel simple") pour optimiser leur gestion.
- **Exemple** : Un cluster de sinistres avec "déclaration tardive + absence de tiers + blessures légères déclarées" peut être suspect.
- **Objectif** : **Créer des catégories de sinistres** actionnables pour les équipes opérationnelles.

Feature Engineering (Réduction de Dimension)

- **Problème** : Résumer des centaines de variables corrélées en quelques "super-variables" pour alimenter un modèle supervisé.
- **Exemple** : Créer un "indice de fragilité financière" à partir de 10 variables économiques pour prédire le risque de rachat en assurance vie.
- **Objectif :** Créer un jeu de variables synthétiques, non corrélées et informatives pour **améliorer la performance et la stabilité d'un modèle supervisé**.

Lutte Anti-Fraude (Détection d'Anomalies)

- **Problème** : Identifier des comportements inhabituels qui pourraient signaler une fraude organisée ou des pratiques abusives.
- **Exemple** : En assurance santé, regrouper les prestataires (médecins, kinés...) en fonction de leurs pratiques de facturation. Un prestataire qui se retrouve isolé ou dans un très petit cluster avec des pratiques très différentes de ses pairs (ex: facture systématiquement les actes les plus chers) est une anomalie à investiguer.
- **Objectif**: **Identifier les observations isolées (outliers)** et les petits groupes atypiques qui s'écartent du comportement "normal" pour une investigation ciblée.

1. Fondements et clustering par partitionnement

Introduction au non supervisé

Préparation des données pour le clustering

Algorithme k-means et son évaluation

ISFA
Institut de Science Financière et d'Assurances

La notion de distance

Distance intra- inter- cluster

Exemple illustratif

Définition : Le clustering est un problème d'optimisation dont l'objectif est de trouver une partition $C = \{C_1, \dots, C_K\}$ des données.

Cet objectif est presque toujours formulé en termes de distances :

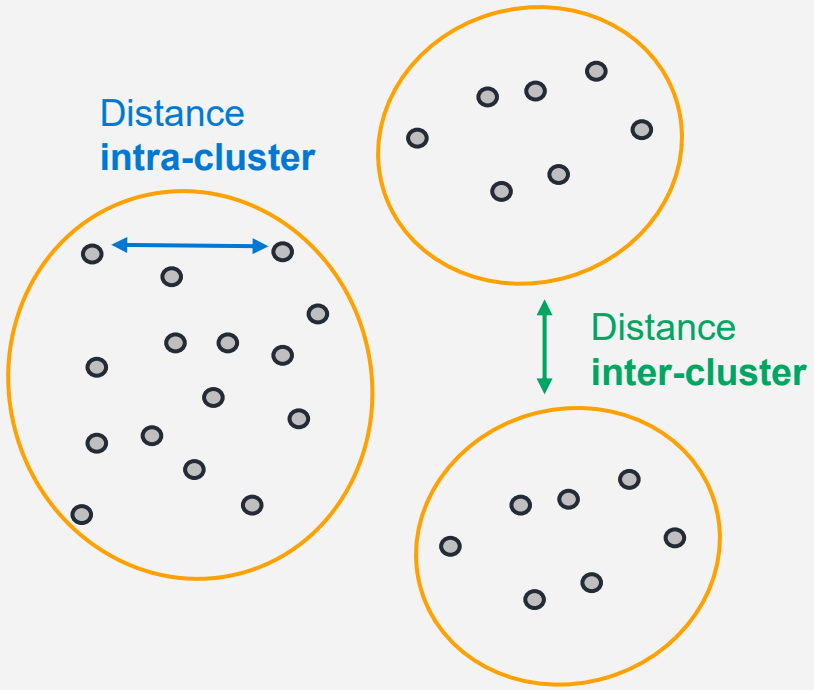
- **Minimiser la distance intra-cluster :** Les points d'un même cluster doivent être proches les uns des autres.
- **Maximiser la distance inter-cluster :** Les points de clusters différents doivent être éloignés.

Le choix de la métrique de distance $d(x_i, x_j)$ est donc une hypothèse fondamentale du modèle.

Distance pour les variables quantitatives (dans $\mathbb{R}^p$) de Minkowski (Généralisation) :

$$d(x_i, x_j) = \left(\Sigma_k \left| x_{ik} - x_{jk} \right|^p \right)^{\frac{1}{p}}$$

- $p = 2$: **Distance Euclidienne** (Norme L2)
  - **Interprétation :** La distance « à vol d'oiseau »
  - **Sensibilité :** Très sensible aux outliers en raison de la mise au carré des écarts.
- $p = 1$: **Distance de Manhattan** (Norme L1)
  - **Interprétation :** La distance « en taxi » dans une grille.
  - **Robustesse :** Moins sensible aux outliers que la distance euclidienne.



La notion de distance et la standardisation

Comment mesurer la similarité de manière équitable

Exemple illustratif

Distance :

Le clustering repose sur une mesure de (dis)similarité. La plus courante est la **distance euclidienne (L2)** :

$$d(A,B) = \sqrt{(Age_A - Age_B)^2 + (Salaire_A - Salaire_B)^2 + \dots}$$

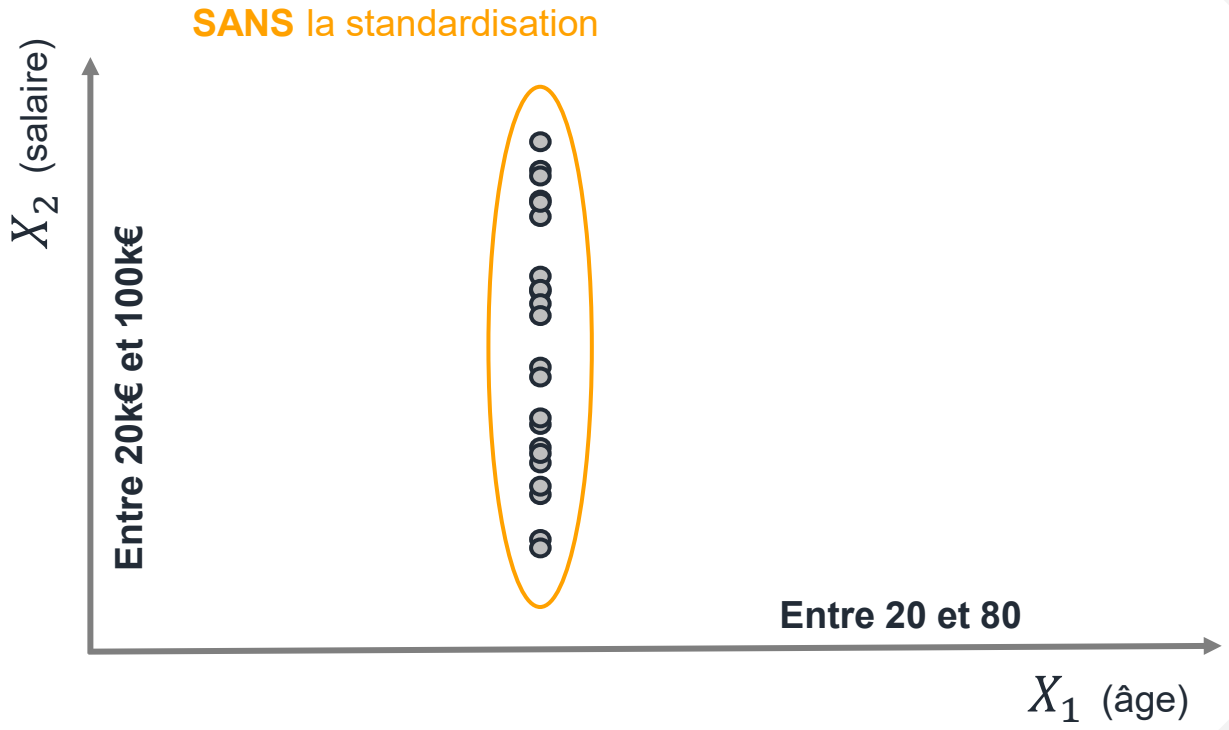
Le problème de l'échelle :

Si l'âge varie de 20 à 80 (amplitude 60) et le salaire de 20k€ à 100k€ (amplitude 80 000), le terme (Salaire_A - Salaire_B)² va complètement dominer le calcul. L'âge n'aura quasiment aucun poids.

La solution : la Standardisation (Z-score)

On transforme chaque variable X_j en $Z_j = \frac{X_j - \mu_j}{\sigma_j}$.

Après standardisation, toutes les variables ont une moyenne de 0 et un écart-type de 1. Elles contribuent de manière égale au calcul de la distance.



La notion de distance et la standardisation

Comment mesurer la similarité de manière équitable

Exemple illustratif

Distance :

Le clustering repose sur une mesure de (dis)similarité. La plus courante est la **distance euclidienne (L2)** :

$$d(A,B) = \sqrt{(Age_A - Age_B)^2 + (Salaire_A - Salaire_B)^2 + \dots}$$

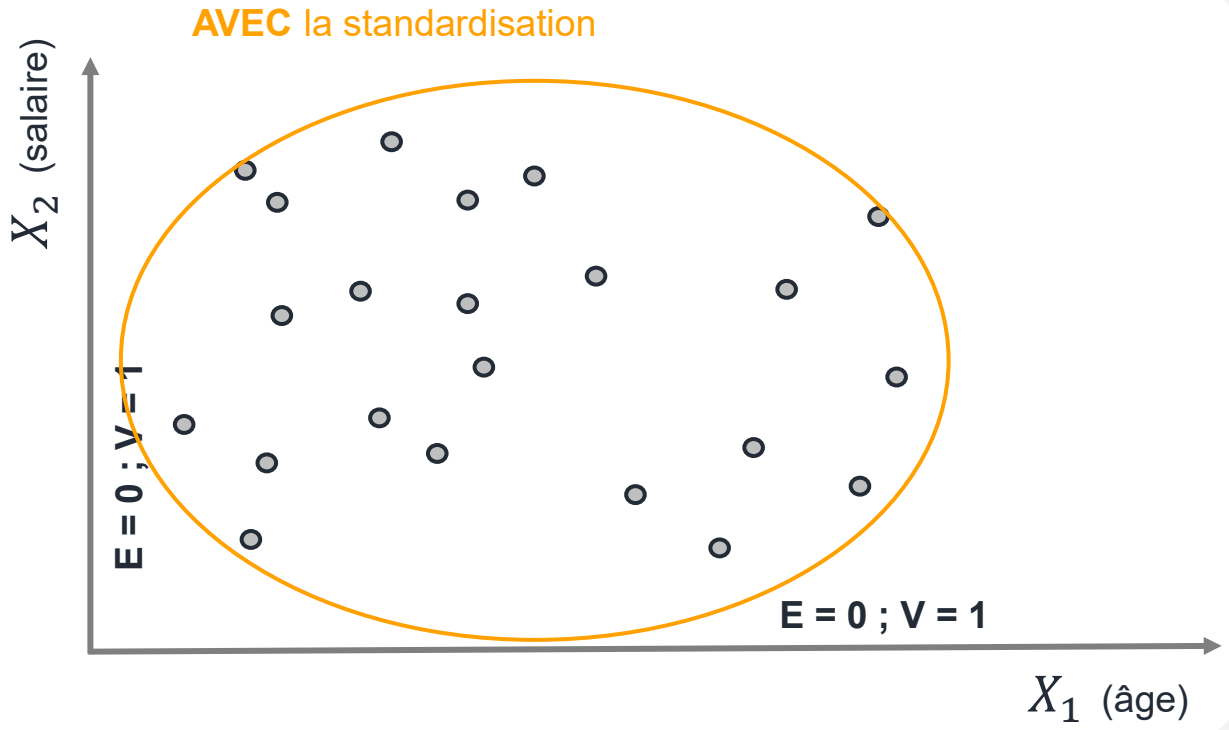
Le problème de l'échelle :

Si l'âge varie de 20 à 80 (amplitude 60) et le salaire de 20k€ à 100k€ (amplitude 80 000), le terme (Salaire_A - Salaire_B)² va complètement dominer le calcul. L'âge n'aura quasiment aucun poids.

La solution : la Standardisation (Z-score)

On transforme chaque variable X_j en $Z_j = \frac{X_j - \mu_j}{\sigma_j}$.

Après standardisation, toutes les variables ont une moyenne de 0 et un écart-type de 1. Elles contribuent de manière égale au calcul de la distance.



Gérer les données mixtes avec la distance de Gower

Une approche unifiée pour les variables quantitatives, qualitatives et binaires

Problème : Nos données sont souvent un mélange de types (numériques, catégorielles, binaires).

La distance de Gower : une approche hybride

Calcule une similarité $s(i,j)$ entre deux observations i et j comme une moyenne pondérée des similarités sur chaque variable k.

$$s(i,j) = \frac{\Sigma\, w_k \times\, s_k(i,j)}{\Sigma\, w_k}$$

Type de Variable	Principe de Similarité	Formule de Calcul (s_k)	Exemples Actuariels
Quantitative	**Distance de Manhattan normalisée**		
Qualitative	La similarité est de 1 si les catégories sont identiques, et de 0 sinon.	1 si CatA == CatB ; 0 sinon	Profession (CSP), Type de véhicule, Zone géographique, Formule de garanties
Binaire	Traitée comme une variable qualitative à deux modalités.	1 si ValA == ValB ; 0 sinon	Propriétaire (Oui/Non), Sexe (M/F), Garantie X souscrite (Oui/Non)

Similarité Globale $s(A,B)$: C'est la moyenne (généralement non pondérée) de toutes les similarités s_k calculées pour chaque variable

Distance de Gower $d(A,B)$: $1 - s(A,B)$

1. Fondements et clustering par partitionnement

Introduction au non supervisé

Préparation des données pour le clustering

Algorithme k-means et son évaluation

ISFA
Institut de Science Financière et d'Assurances

Algorithme k-means

Une danse itérative entre assignation et mise à jour

Exemple illustratif

Objectif : Partitionner un ensemble de n observations x_i en K clusters $C = \{C_1, \dots, C_K\}$ de manière à minimiser l'**inertie intra-cluster** (Within-Cluster Sum of Squares).

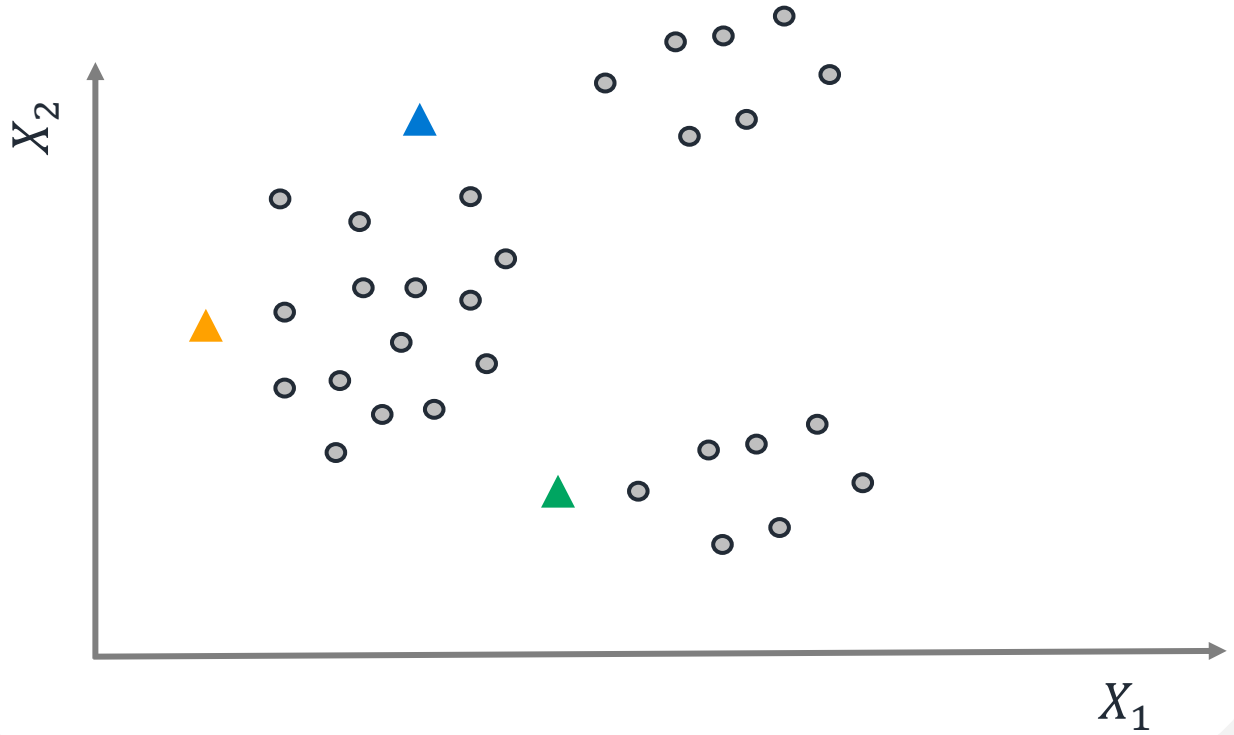
Fonction objectif : Minimiser,

$$WCSS = \sum_{k=1\ to\ K} \sum_{x_i \in C_k} ||x_i - \mu_k||^2$$

où μ_k est le centroïde (la moyenne) du cluster k.

Algorithme :

1. **Initialisation :** Choisir K centroïdes initiaux (ex: au hasard).
2. **Assignation :** Assigner chaque point x_i au cluster C_k dont le centroïde μ_k est le plus proche.
3. **Mise à jour :** Recalculer le centroïde μ_k de chaque cluster comme étant la moyenne de tous les points x_i assignés à C_k.
4. Répéter les étapes 2 et 3 jusqu'à ce que les assignations de clusters ne changent plus.



Algorithme k-means

Une danse itérative entre assignation et mise à jour

Exemple illustratif

Objectif : Partitionner un ensemble de n observations x_i en K clusters $C = \{C_1, \dots, C_K\}$ de manière à minimiser l'**inertie intra-cluster** (Within-Cluster Sum of Squares).

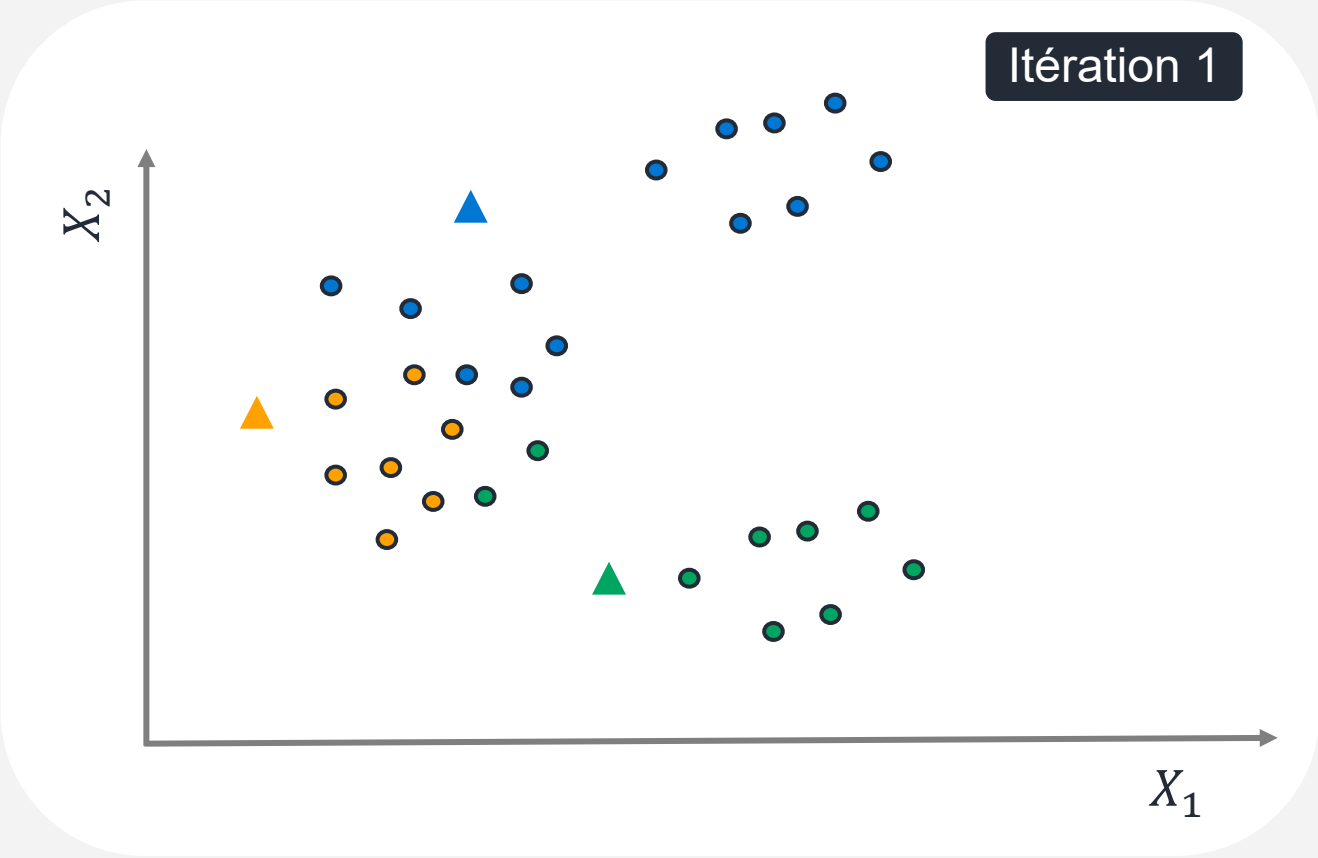
Fonction objectif : Minimiser,

$$WCSS = \sum_{k=1\ to\ K} \sum_{x_i \in C_k} ||x_i - \mu_k||^2$$

où μ_k est le centroïde (la moyenne) du cluster k.

Algorithme :

1. **Initialisation :** Choisir K centroïdes initiaux (ex: au hasard).
2. **Assignation :** Assigner chaque point x_i au cluster C_k dont le centroïde μ_k est le plus proche.
3. **Mise à jour :** Recalculer le centroïde μ_k de chaque cluster comme étant la moyenne de tous les points x_i assignés à C_k.
4. Répéter les étapes 2 et 3 jusqu'à ce que les assignations de clusters ne changent plus.



Algorithme k-means

Une danse itérative entre assignation et mise à jour

Exemple illustratif

Objectif : Partitionner un ensemble de n observations x_i en K clusters $C = \{C_1, \dots, C_K\}$ de manière à minimiser l'**inertie intra-cluster** (Within-Cluster Sum of Squares).

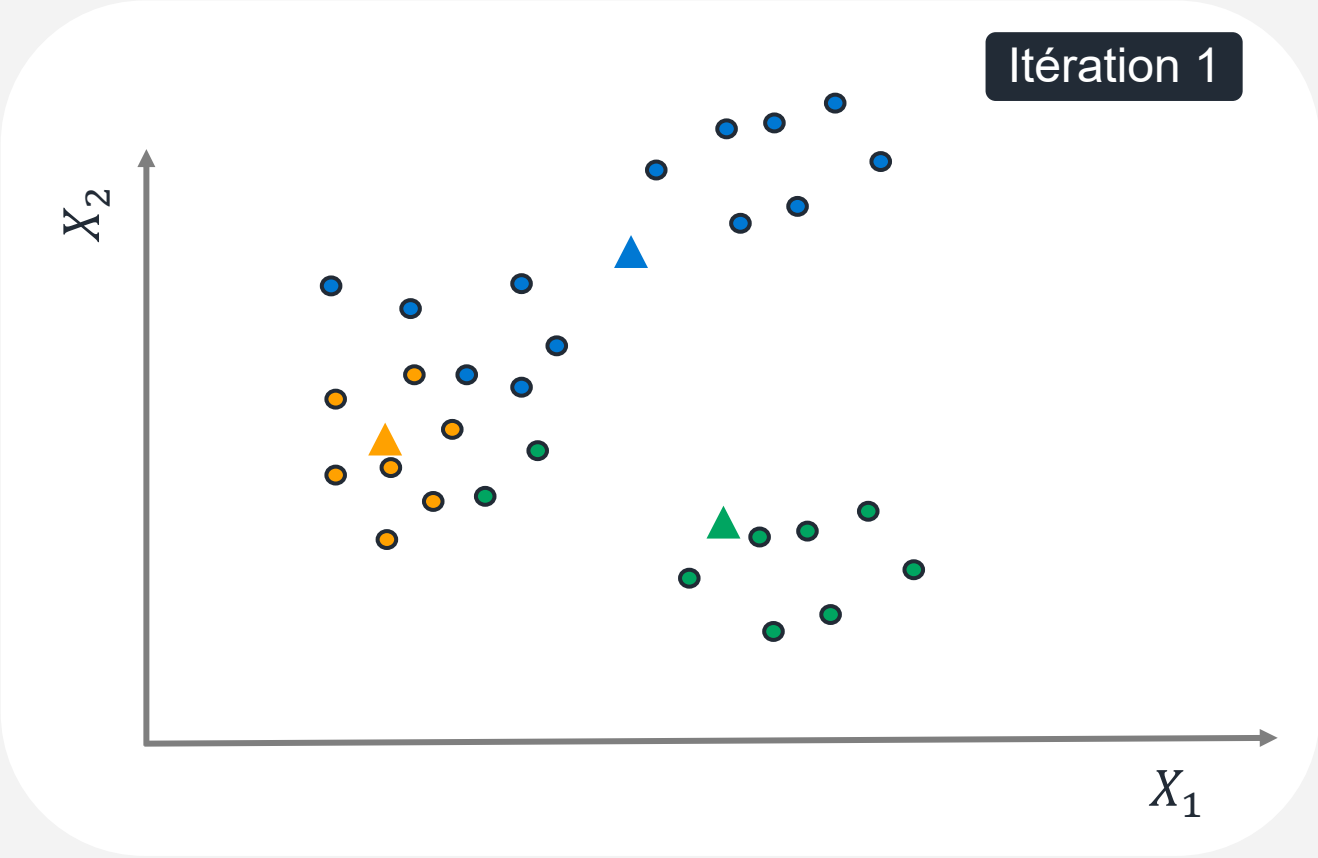
Fonction objectif : Minimiser,

$$WCSS = \sum_{k=1\ to\ K} \sum_{x_i \in C_k} ||x_i - \mu_k||^2$$

où μ_k est le centroïde (la moyenne) du cluster k.

Algorithme :

1. **Initialisation :** Choisir K centroïdes initiaux (ex: au hasard).
2. **Assignation :** Assigner chaque point x_i au cluster C_k dont le centroïde μ_k est le plus proche.
3. **Mise à jour :** Recalculer le centroïde μ_k de chaque cluster comme étant la moyenne de tous les points x_i assignés à C_k.
4. Répéter les étapes 2 et 3 jusqu'à ce que les assignations de clusters ne changent plus.



Algorithme k-means

Une danse itérative entre assignation et mise à jour

Objectif : Partitionner un ensemble de n observations x_i en K clusters $C = \{C_1, \dots, C_K\}$ de manière à minimiser l'**inertie intra-cluster** (Within-Cluster Sum of Squares).

Fonction objectif : Minimiser,

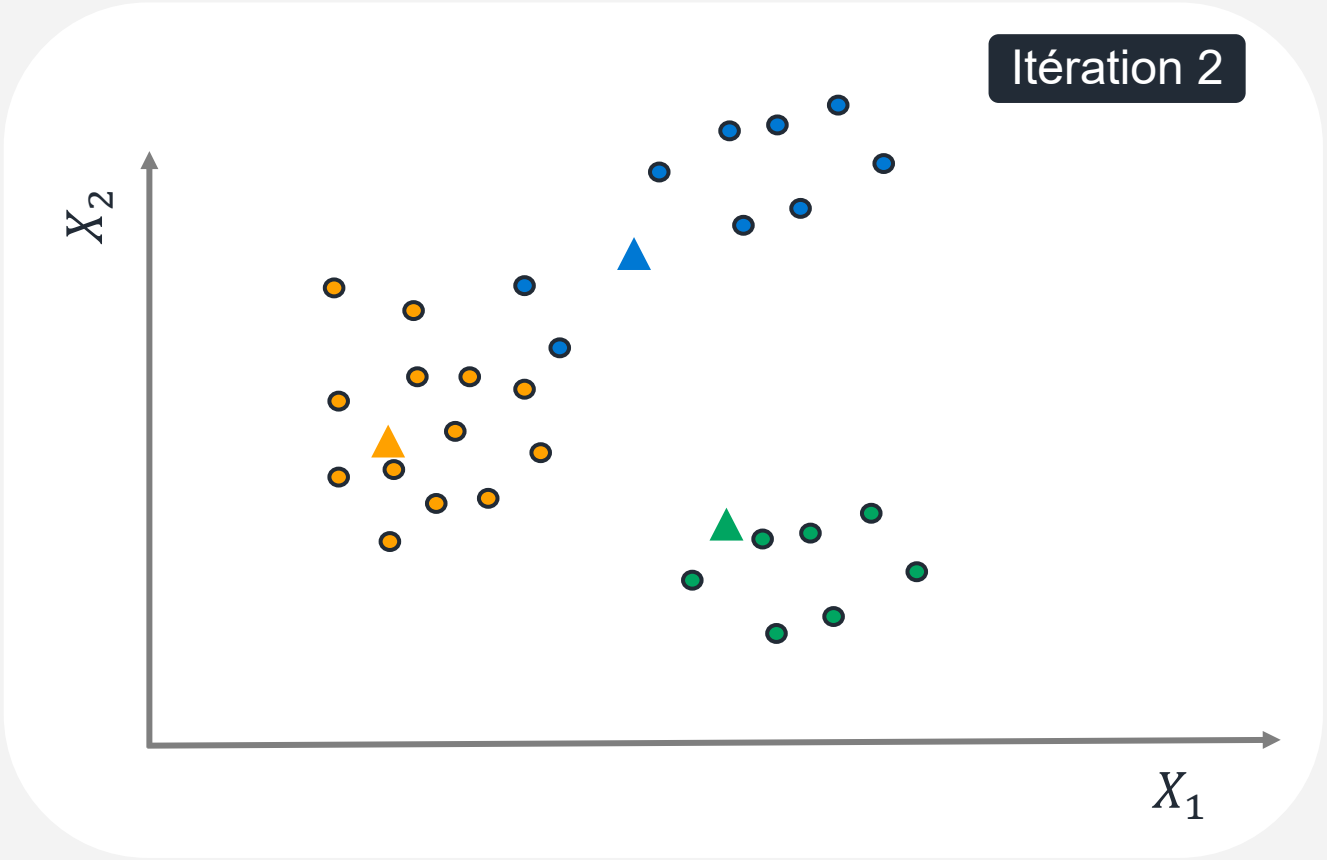
$$WCSS = \sum_{k=1\ to\ K} \sum_{x_i \in C_k} ||x_i - \mu_k||^2$$

où μ_k est le centroïde (la moyenne) du cluster k.

Algorithme :

1. **Initialisation :** Choisir K centroïdes initiaux (ex: au hasard).
2. **Assignation :** Assigner chaque point x_i au cluster C_k dont le centroïde μ_k est le plus proche.
3. **Mise à jour :** Recalculer le centroïde μ_k de chaque cluster comme étant la moyenne de tous les points x_i assignés à C_k.
4. Répéter les étapes 2 et 3 jusqu'à ce que les assignations de clusters ne changent plus.

Exemple illustratif



Algorithme k-means

Une danse itérative entre assignation et mise à jour

Objectif : Partitionner un ensemble de n observations x_i en K clusters $C = \{C_1, \dots, C_K\}$ de manière à minimiser l'**inertie intra-cluster** (Within-Cluster Sum of Squares).

Fonction objectif : Minimiser,

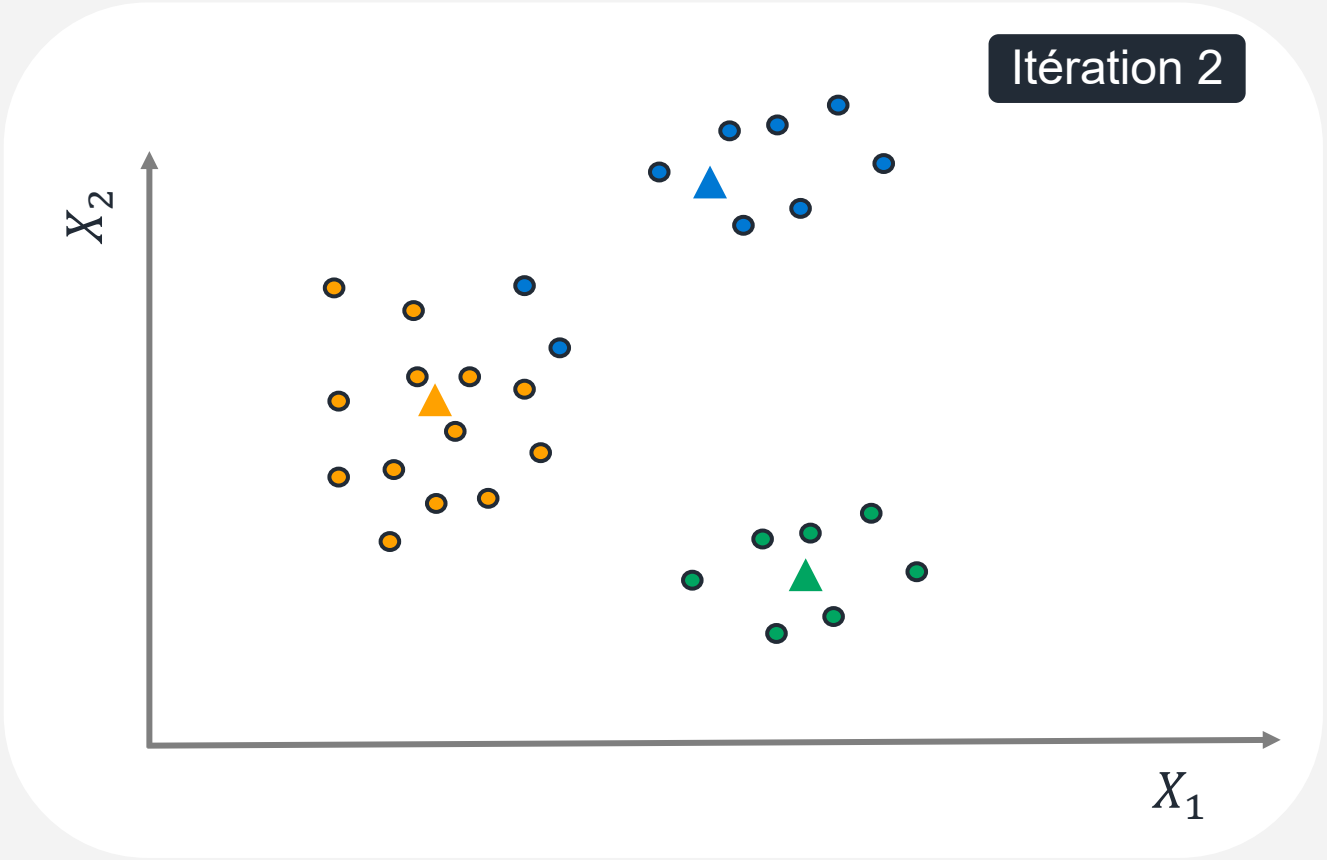
$$WCSS = \sum_{k=1\ to\ K} \sum_{x_i \in C_k} ||x_i - \mu_k||^2$$

où μ_k est le centroïde (la moyenne) du cluster k.

Algorithme :

1. **Initialisation :** Choisir K centroïdes initiaux (ex: au hasard).
2. **Assignation :** Assigner chaque point x_i au cluster C_k dont le centroïde μ_k est le plus proche.
3. **Mise à jour :** Recalculer le centroïde μ_k de chaque cluster comme étant la moyenne de tous les points x_i assignés à C_k.
4. Répéter les étapes 2 et 3 jusqu'à ce que les assignations de clusters ne changent plus.

Exemple illustratif



Algorithme k-means

Une danse itérative entre assignation et mise à jour

Exemple illustratif

Objectif : Partitionner un ensemble de n observations x_i en K clusters $C = \{C_1, \dots, C_K\}$ de manière à minimiser l'**inertie intra-cluster** (Within-Cluster Sum of Squares).

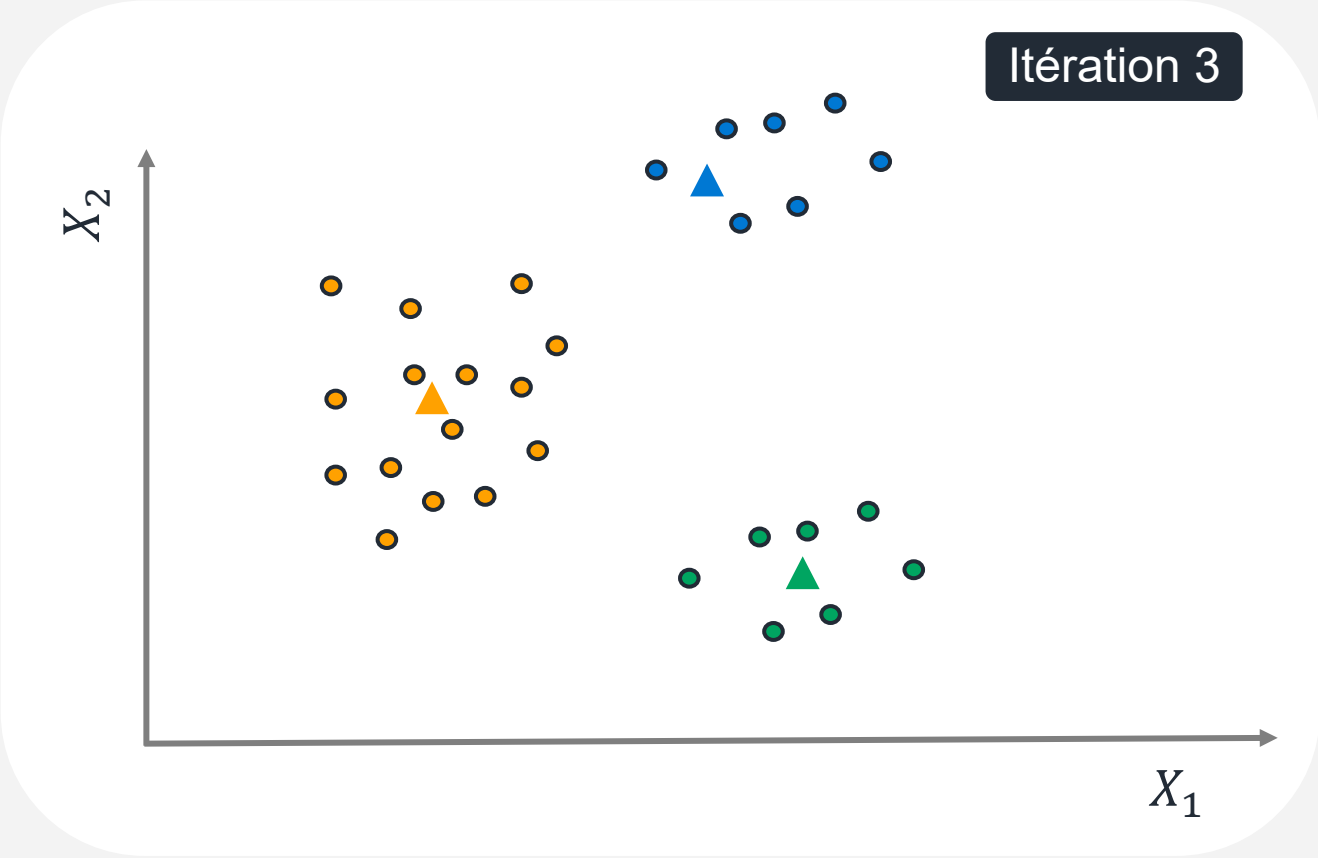
Fonction objectif : Minimiser,

$$WCSS = \sum_{k=1\ to\ K} \sum_{x_i \in C_k} ||x_i - \mu_k||^2$$

où μ_k est le centroïde (la moyenne) du cluster k.

Algorithme :

1. **Initialisation :** Choisir K centroïdes initiaux (ex: au hasard).
2. **Assignation :** Assigner chaque point x_i au cluster C_k dont le centroïde μ_k est le plus proche.
3. **Mise à jour :** Recalculer le centroïde μ_k de chaque cluster comme étant la moyenne de tous les points x_i assignés à C_k.
4. Répéter les étapes 2 et 3 jusqu'à ce que les assignations de clusters ne changent plus.



Algorithme k-means

Une danse itérative entre assignation et mise à jour

Objectif : Partitionner un ensemble de n observations x_i en K clusters $C = \{C_1, \dots, C_K\}$ de manière à minimiser l'**inertie intra-cluster** (Within-Cluster Sum of Squares).

Fonction objectif : Minimiser,

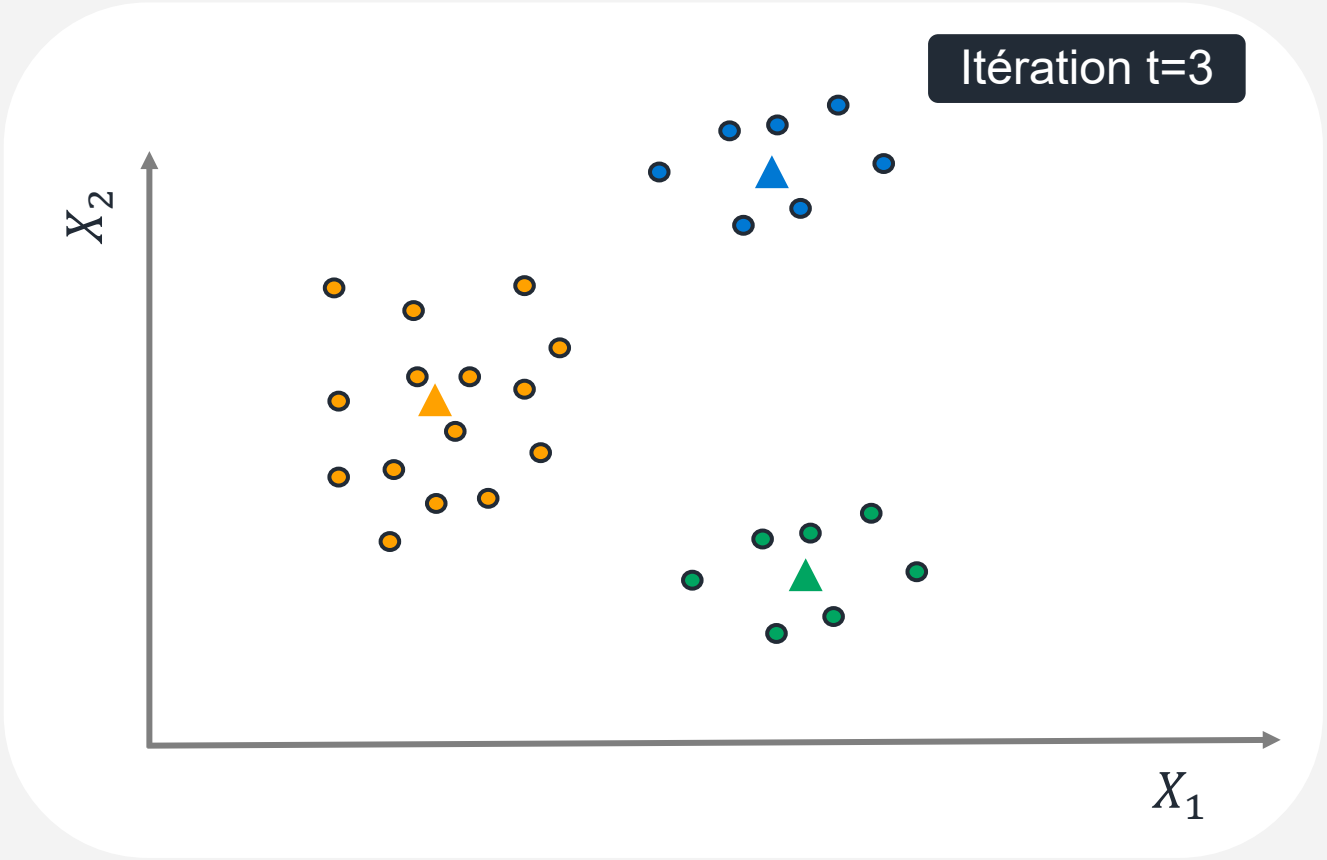
$$WCSS = \sum_{k=1\ to\ K} \sum_{x_i \in C_k} ||x_i - \mu_k||^2$$

où μ_k est le centroïde (la moyenne) du cluster k.

Algorithme :

1. **Initialisation :** Choisir K centroïdes initiaux (ex: au hasard).
2. **Assignation :** Assigner chaque point x_i au cluster C_k dont le centroïde μ_k est le plus proche.
3. **Mise à jour :** Recalculer le centroïde μ_k de chaque cluster comme étant la moyenne de tous les points x_i assignés à C_k.
4. Répéter les étapes 2 et 3 jusqu'à ce que les assignations de clusters ne changent plus.

Exemple illustratif



Algorithme k-means

Une danse itérative entre assignation et mise à jour

Exemple illustratif

Objectif : Partitionner un ensemble de n observations x_i en K clusters $C = \{C_1, \dots, C_K\}$ de manière à minimiser l'**inertie intra-cluster** (Within-Cluster Sum of Squares).

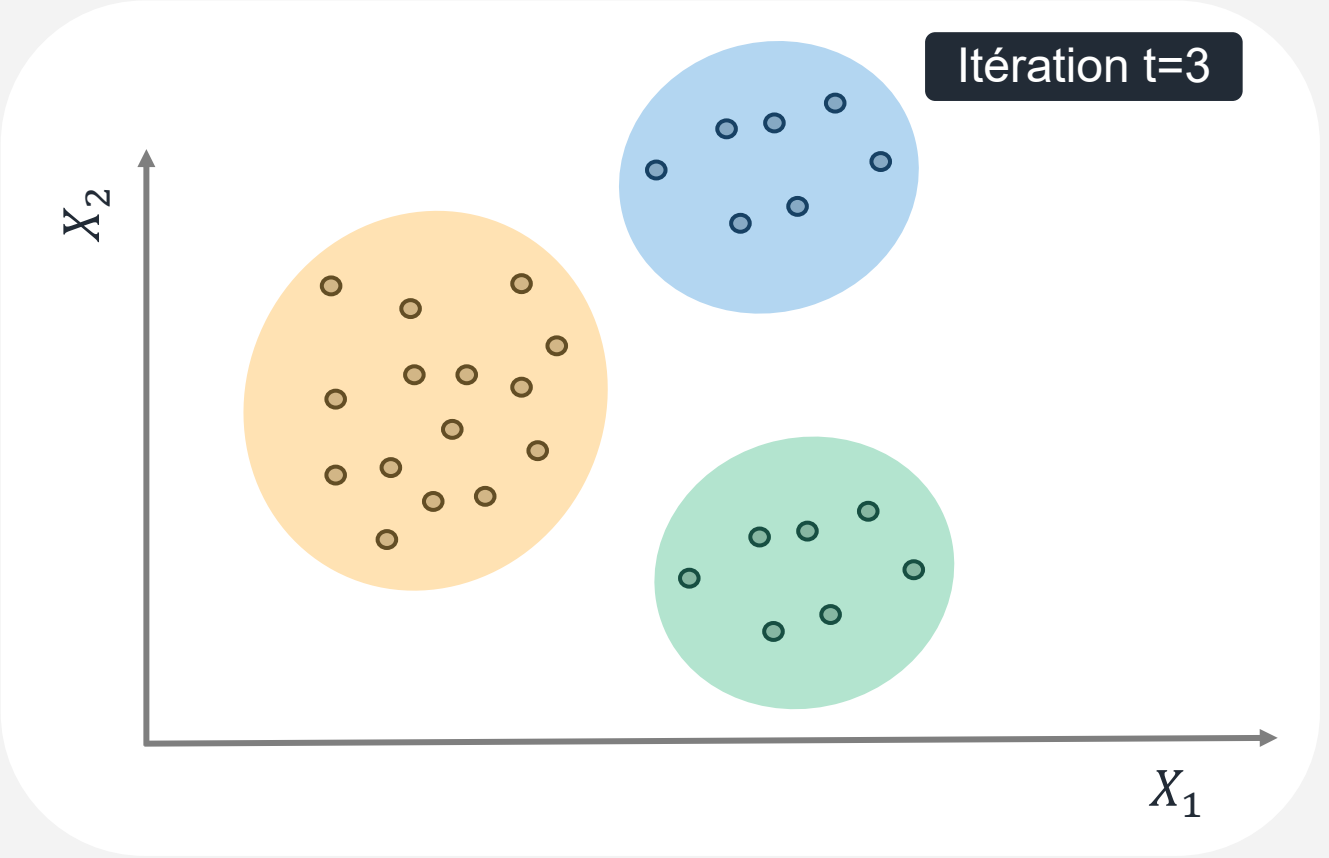
Fonction objectif : Minimiser,

$$WCSS = \sum_{k=1\ to\ K} \sum_{x_i \in C_k} ||x_i - \mu_k||^2$$

où μ_k est le centroïde (la moyenne) du cluster k.

Algorithme :

1. **Initialisation :** Choisir K centroïdes initiaux (ex: au hasard).
2. **Assignation :** Assigner chaque point x_i au cluster C_k dont le centroïde μ_k est le plus proche.
3. **Mise à jour :** Recalculer le centroïde μ_k de chaque cluster comme étant la moyenne de tous les points x_i assignés à C_k.
4. Répéter les étapes 2 et 3 jusqu'à ce que les assignations de clusters ne changent plus.



Analyse critique de k-means

Forces, faiblesses, et comment l'améliorer

Caractéristique	Description et Analyse	Solution ou Alternative
Vitesse / Scalabilité	La complexité est quasi-linéaire ($O(n * K * t * d)$). K-means est capables de traiter des millions d'observations efficacement. C'est une excellente **baseline**.	
Simplicité / Interprétabilité	L'algorithme est facile à comprendre. Les clusters sont simplement définis par leur centre de gravité (le centroïde μ_k), qui est facile à interpréter comme le "profil moyen" du cluster.	
Sensibilité à l'initialisation	Une mauvaise initialisation des centroïdes peut mener l'algorithme à converger vers un **minimum local** de la fonction objectif (WCSS), donnant une partition de mauvaise qualité.	L'initialisation **k-means++** est la solution standard. Elle choisit des centroïdes initiaux bien espacés, ce qui améliore considérablement la qualité et la stabilité des résultats.
Sensibilité aux Outliers	Le centroïde est une **moyenne**, qui est une mesure très sensible aux valeurs extrêmes. Un seul outlier peut significativement "tirer" le centre d'un cluster vers lui et fausser la partition.	Utiliser **K-Medoids (PAM)**, qui utilise des observations réelles (médoïdes) comme centres, une approche basée sur une mesure plus robuste (similaire à la médiane).
Hypothèses Géométriques	K-means suppose implicitement que les clusters sont **sphériques** (isotropes), de **taille similaire** et de **densité égale**. Il échoue sur des formes plus complexes.	Utiliser des algorithmes plus flexibles : - **DBSCAN** pour les formes arbitraires - **GMM** (Mélanges Gaussiens) pour les formes elliptiques.
Nécessité de fixer K	L'algorithme ne découvre pas le nombre de clusters. L'utilisateur doit le **spécifier à l'avance**, ce qui peut être difficile.	Utiliser des heuristiques comme la **méthode du Coude** ou le **Coefficient de Silhouette** pour guider le choix de K.

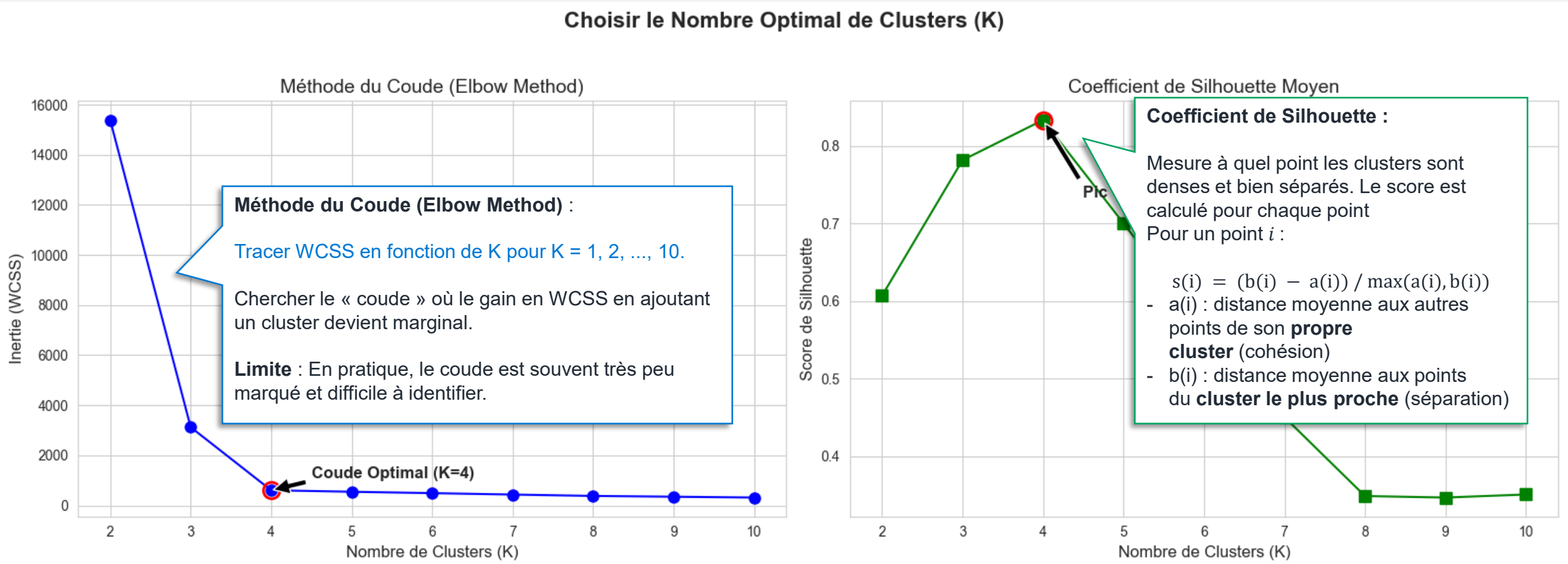
Focus de cette session

ISFA
Institut de Science Financière et d'Assurances

Le choix du nombre de clusters k

Utiliser des heuristiques pour guider une décision métier

Le choix de K est le principal hyperparamètre. Il n'y a pas de « bonne » réponse unique, cela dépend du niveau de granularité souhaité.



Approche finale : Utiliser ces méthodes pour proposer 2 ou 3 valeurs de K plausibles (ex: $K = 3$ et $K = 5$), puis analyser les profils des clusters pour chaque K et choisir la segmentation la plus **interprétable et actionnable** pour le métier.

2. Clustering avancé : hiérarchie et densité

- **Le clustering hiérarchique**
- Le clustering par densité (DBSCAN)
- Synthèse des méthodes de clustering

ISFA
Institut de Science Financière et d'Assurances

Le clustering hiérarchique

Construire un arbre de relations plutôt qu'une partition fixe

Exemple illustratif

Motivation : K-Means nous force à choisir K à l'avance. Le clustering hiérarchique propose une approche plus exploratoire.

Philosophie : On ne crée pas une seule partition, mais une **hiérarchie de partitions imbriquées**.

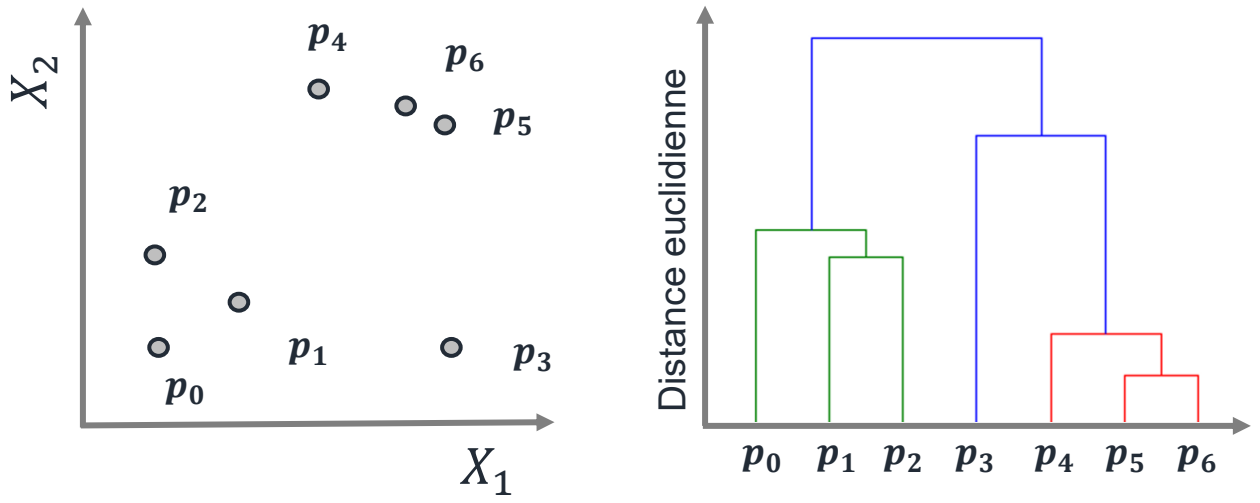
Approche la plus courante :

Agglomérative (Bottom-Up)

1. **Initialisation :** Chaque observation commence dans son propre cluster (on a n clusters).
2. **Itération :** À chaque étape, on trouve les deux clusters les plus "proches" et on les **fusionne**.
3. **Fin :** On répète ce processus n-1 fois, jusqu'à ce qu'il ne reste qu'un seul grand cluster contenant toutes les données.

Le résultat : Cet historique de fusions est stocké dans une structure en arbre appelée **dendrogramme**.

initialisation



Le clustering hiérarchique

Construire un arbre de relations plutôt qu'une partition fixe

Exemple illustratif

Motivation : K-Means nous force à choisir K à l'avance. Le clustering hiérarchique propose une approche plus exploratoire.

Philosophie : On ne crée pas une seule partition, mais une **hiérarchie de partitions imbriquées**.

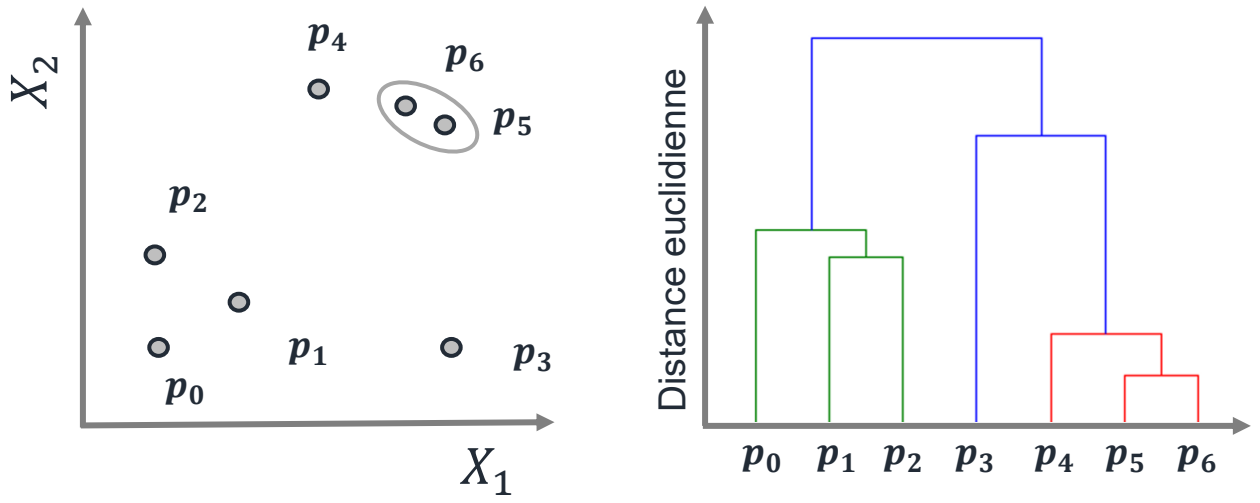
Approche la plus courante :

Agglomérative (Bottom-Up)

1. **Initialisation :** Chaque observation commence dans son propre cluster (on a n clusters).
2. **Itération :** À chaque étape, on trouve les deux clusters les plus "proches" et on les **fusionne**.
3. **Fin :** On répète ce processus n-1 fois, jusqu'à ce qu'il ne reste qu'un seul grand cluster contenant toutes les données.

Le résultat : Cet historique de fusions est stocké dans une structure en arbre appelée **dendrogramme**.

Itération t=1



Le clustering hiérarchique

Construire un arbre de relations plutôt qu'une partition fixe

Exemple illustratif

Motivation : K-Means nous force à choisir K à l'avance. Le clustering hiérarchique propose une approche plus exploratoire.

Philosophie : On ne crée pas une seule partition, mais une **hiérarchie de partitions imbriquées**.

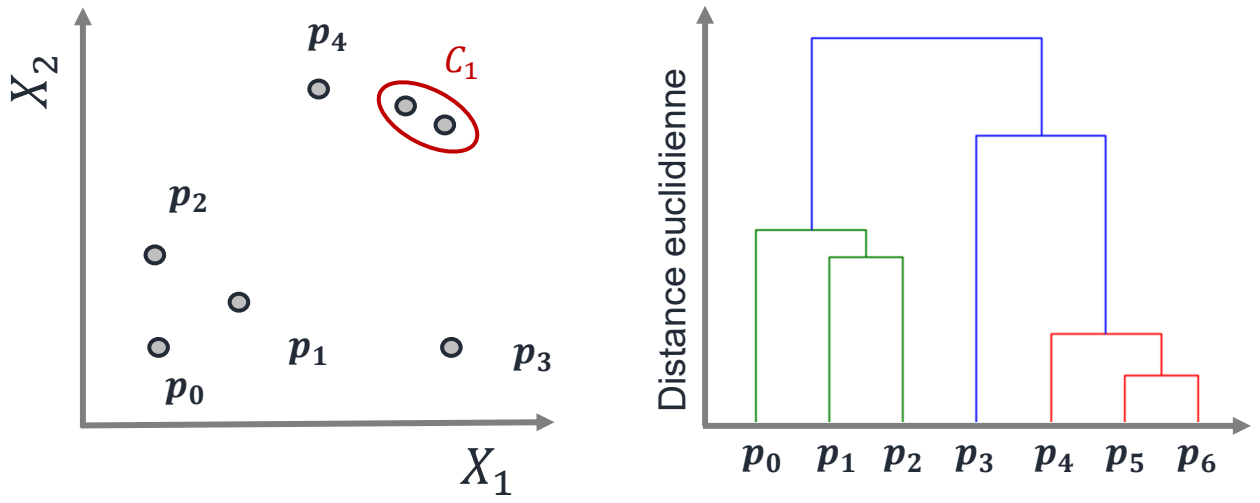
Approche la plus courante :

Agglomérative (Bottom-Up)

1. **Initialisation :** Chaque observation commence dans son propre cluster (on a n clusters).
2. **Itération :** À chaque étape, on trouve les deux clusters les plus "proches" et on les **fusionne**.
3. **Fin :** On répète ce processus n-1 fois, jusqu'à ce qu'il ne reste qu'un seul grand cluster contenant toutes les données.

Le résultat : Cet historique de fusions est stocké dans une structure en arbre appelée **dendrogramme**.

Itération t=1



Le clustering hiérarchique

Construire un arbre de relations plutôt qu'une partition fixe

Exemple illustratif

Motivation : K-Means nous force à choisir K à l'avance. Le clustering hiérarchique propose une approche plus exploratoire.

Philosophie : On ne crée pas une seule partition, mais une **hiérarchie de partitions imbriquées**.

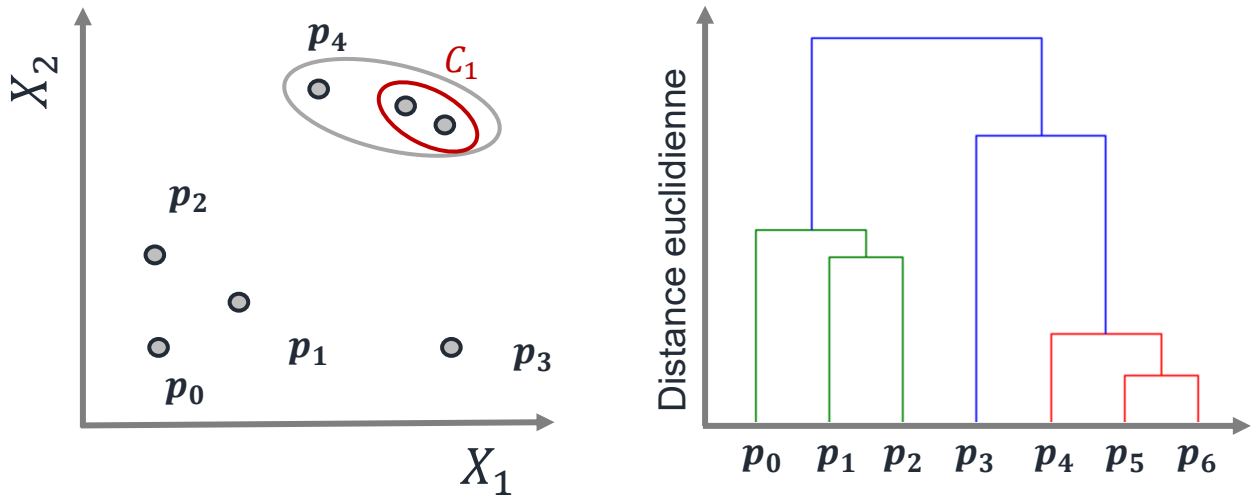
Approche la plus courante :

Agglomérative (Bottom-Up)

1. **Initialisation :** Chaque observation commence dans son propre cluster (on a n clusters).
2. **Itération :** À chaque étape, on trouve les deux clusters les plus "proches" et on les **fusionne**.
3. **Fin :** On répète ce processus n-1 fois, jusqu'à ce qu'il ne reste qu'un seul grand cluster contenant toutes les données.

Le résultat : Cet historique de fusions est stocké dans une structure en arbre appelée **dendrogramme**.

Itération t=2



Le clustering hiérarchique

Construire un arbre de relations plutôt qu'une partition fixe

Exemple illustratif

Motivation : K-Means nous force à choisir K à l'avance. Le clustering hiérarchique propose une approche plus exploratoire.

Philosophie : On ne crée pas une seule partition, mais une **hiérarchie de partitions imbriquées**.

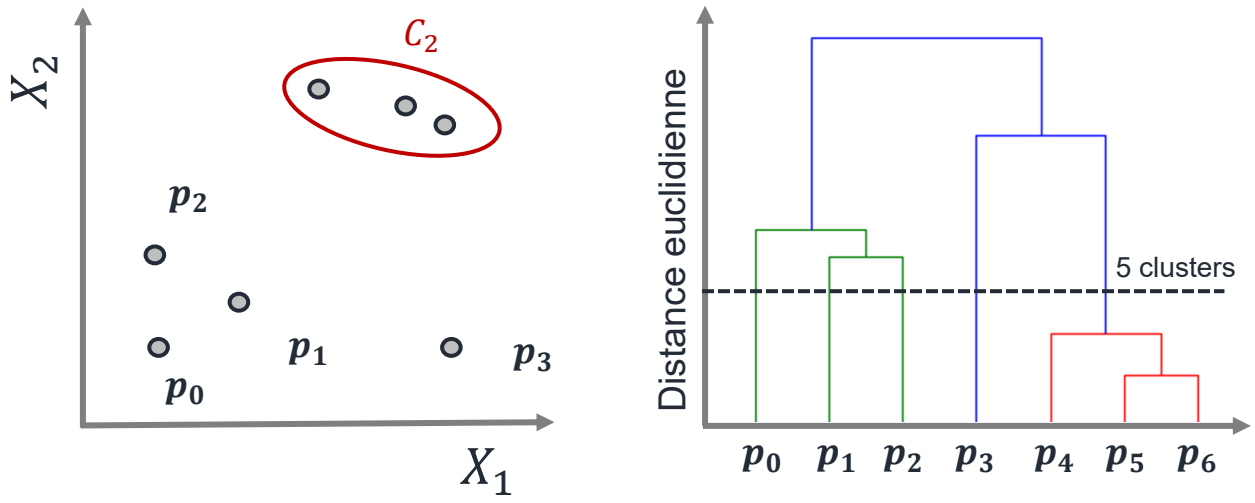
Approche la plus courante :

Agglomérative (Bottom-Up)

1. **Initialisation :** Chaque observation commence dans son propre cluster (on a n clusters).
2. **Itération :** À chaque étape, on trouve les deux clusters les plus "proches" et on les **fusionne**.
3. **Fin :** On répète ce processus n-1 fois, jusqu'à ce qu'il ne reste qu'un seul grand cluster contenant toutes les données.

Le résultat : Cet historique de fusions est stocké dans une structure en arbre appelée **dendrogramme**.

Itération t=2



Le clustering hiérarchique

Construire un arbre de relations plutôt qu'une partition fixe

Exemple illustratif

Motivation : K-Means nous force à choisir K à l'avance. Le clustering hiérarchique propose une approche plus exploratoire.

Philosophie : On ne crée pas une seule partition, mais une **hiérarchie de partitions imbriquées**.

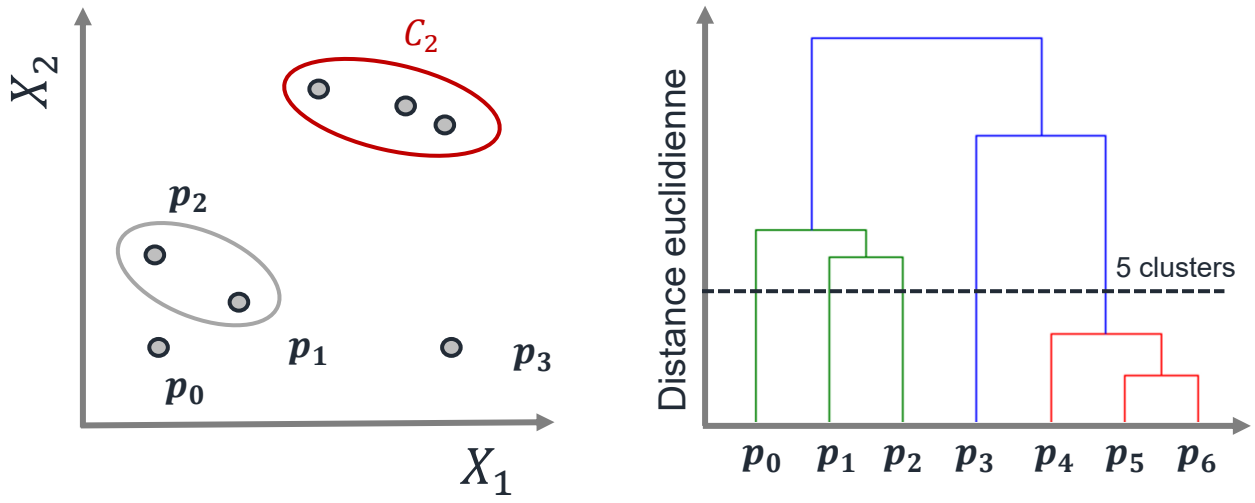
Approche la plus courante :

Agglomérative (Bottom-Up)

1. **Initialisation :** Chaque observation commence dans son propre cluster (on a n clusters).
2. **Itération :** À chaque étape, on trouve les deux clusters les plus "proches" et on les **fusionne**.
3. **Fin :** On répète ce processus n-1 fois, jusqu'à ce qu'il ne reste qu'un seul grand cluster contenant toutes les données.

Le résultat : Cet historique de fusions est stocké dans une structure en arbre appelée **dendrogramme**.

Itération t=3



Le clustering hiérarchique

Construire un arbre de relations plutôt qu'une partition fixe

Exemple illustratif

Motivation : K-Means nous force à choisir K à l'avance. Le clustering hiérarchique propose une approche plus exploratoire.

Philosophie : On ne crée pas une seule partition, mais une **hiérarchie de partitions imbriquées**.

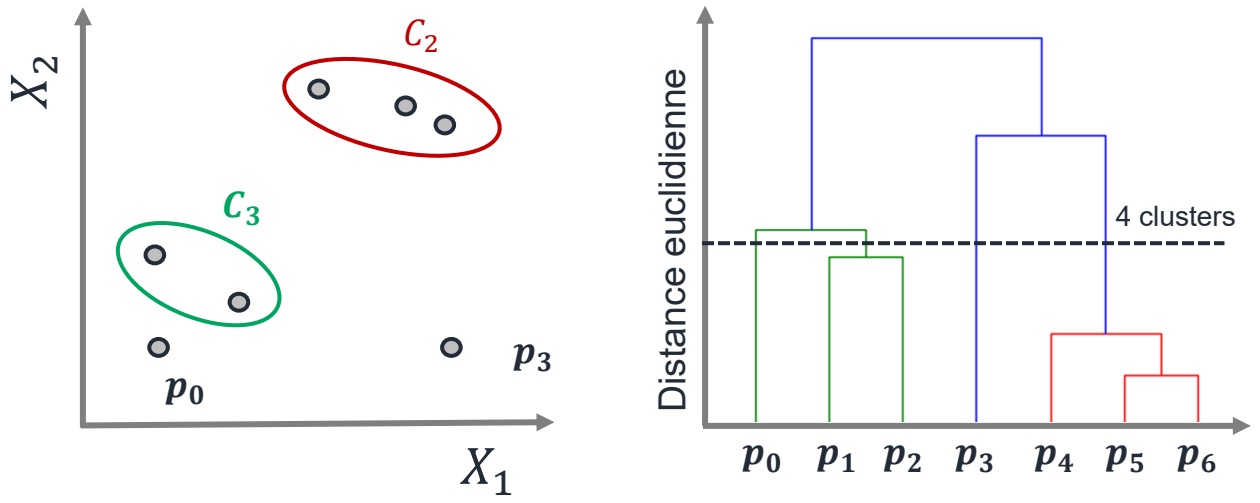
Approche la plus courante :

Agglomérative (Bottom-Up)

1. **Initialisation :** Chaque observation commence dans son propre cluster (on a n clusters).
2. **Itération :** À chaque étape, on trouve les deux clusters les plus "proches" et on les **fusionne**.
3. **Fin :** On répète ce processus n-1 fois, jusqu'à ce qu'il ne reste qu'un seul grand cluster contenant toutes les données.

Le résultat : Cet historique de fusions est stocké dans une structure en arbre appelée **dendrogramme**.

Itération t=3



Le clustering hiérarchique

Construire un arbre de relations plutôt qu'une partition fixe

Exemple illustratif

Motivation : K-Means nous force à choisir K à l'avance. Le clustering hiérarchique propose une approche plus exploratoire.

Philosophie : On ne crée pas une seule partition, mais une **hiérarchie de partitions imbriquées**.

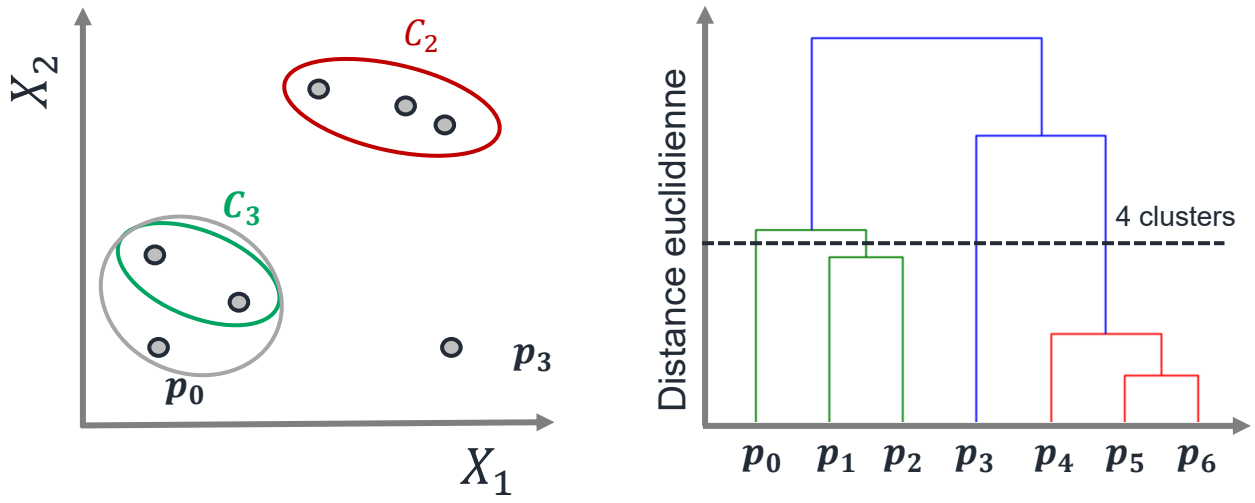
Approche la plus courante :

Agglomérative (Bottom-Up)

1. **Initialisation :** Chaque observation commence dans son propre cluster (on a n clusters).
2. **Itération :** À chaque étape, on trouve les deux clusters les plus "proches" et on les **fusionne**.
3. **Fin :** On répète ce processus n-1 fois, jusqu'à ce qu'il ne reste qu'un seul grand cluster contenant toutes les données.

Le résultat : Cet historique de fusions est stocké dans une structure en arbre appelée **dendrogramme**.

Itération t=3



Le clustering hiérarchique

Construire un arbre de relations plutôt qu'une partition fixe

Exemple illustratif

Motivation : K-Means nous force à choisir K à l'avance. Le clustering hiérarchique propose une approche plus exploratoire.

Philosophie : On ne crée pas une seule partition, mais une **hiérarchie de partitions imbriquées**.

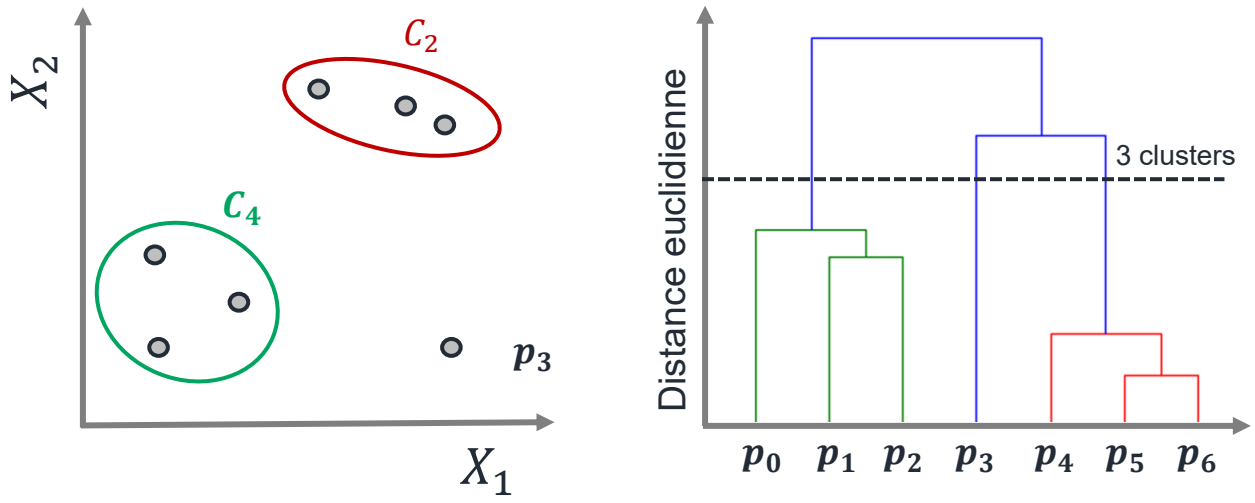
Approche la plus courante :

Agglomérative (Bottom-Up)

1. **Initialisation :** Chaque observation commence dans son propre cluster (on a n clusters).
2. **Itération :** À chaque étape, on trouve les deux clusters les plus "proches" et on les **fusionne**.
3. **Fin :** On répète ce processus n-1 fois, jusqu'à ce qu'il ne reste qu'un seul grand cluster contenant toutes les données.

Le résultat : Cet historique de fusions est stocké dans une structure en arbre appelée **dendrogramme**.

Itération t=3



Le clustering hiérarchique

Construire un arbre de relations plutôt qu'une partition fixe

Exemple illustratif

Motivation : K-Means nous force à choisir K à l'avance. Le clustering hiérarchique propose une approche plus exploratoire.

Philosophie : On ne crée pas une seule partition, mais une **hiérarchie de partitions imbriquées**.

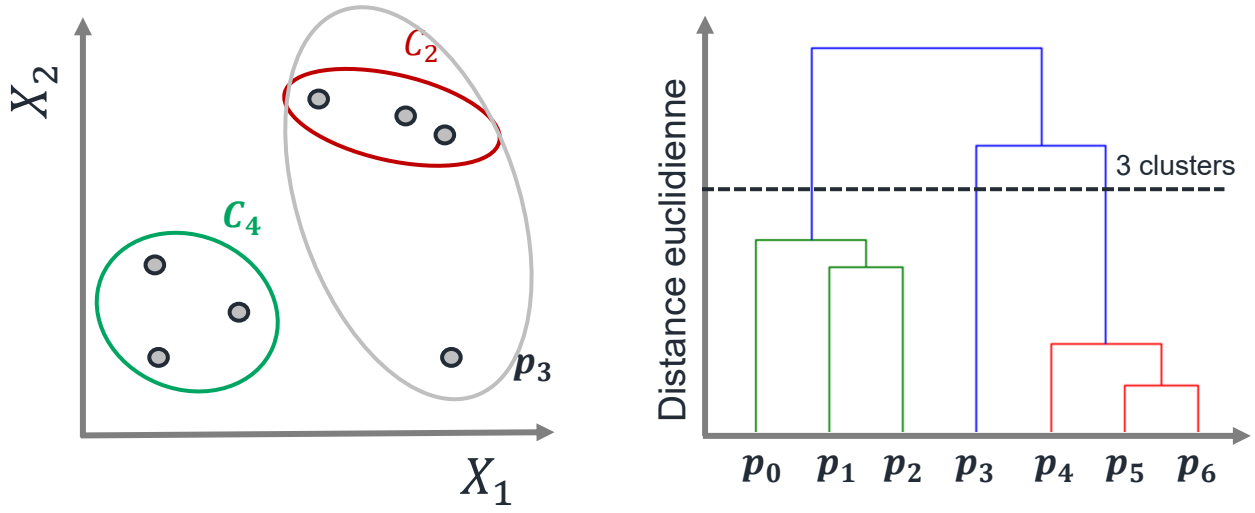
Approche la plus courante :

Agglomérative (Bottom-Up)

1. **Initialisation :** Chaque observation commence dans son propre cluster (on a n clusters).
2. **Itération :** À chaque étape, on trouve les deux clusters les plus "proches" et on les **fusionne**.
3. **Fin :** On répète ce processus n-1 fois, jusqu'à ce qu'il ne reste qu'un seul grand cluster contenant toutes les données.

Le résultat : Cet historique de fusions est stocké dans une structure en arbre appelée **dendrogramme**.

Itération t=4



Le clustering hiérarchique

Construire un arbre de relations plutôt qu'une partition fixe

Exemple illustratif

Motivation : K-Means nous force à choisir K à l'avance. Le clustering hiérarchique propose une approche plus exploratoire.

Philosophie : On ne crée pas une seule partition, mais une **hiérarchie de partitions imbriquées**.

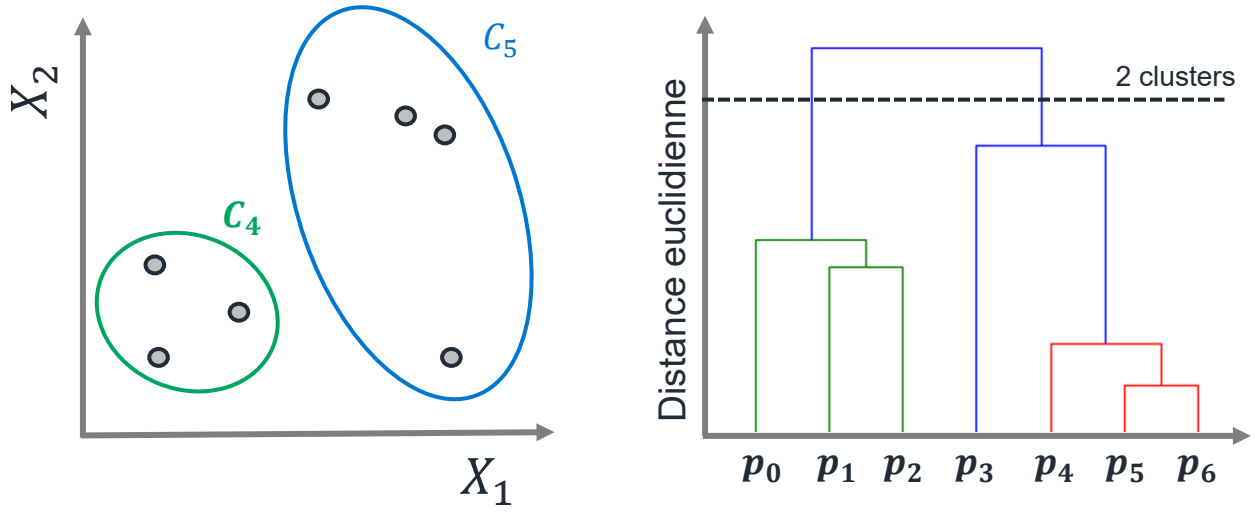
Approche la plus courante :

Agglomérative (Bottom-Up)

1. **Initialisation :** Chaque observation commence dans son propre cluster (on a n clusters).
2. **Itération :** À chaque étape, on trouve les deux clusters les plus "proches" et on les **fusionne**.
3. **Fin :** On répète ce processus n-1 fois, jusqu'à ce qu'il ne reste qu'un seul grand cluster contenant toutes les données.

Le résultat : Cet historique de fusions est stocké dans une structure en arbre appelée **dendrogramme**.

Itération t=4



Le clustering hiérarchique

Construire un arbre de relations plutôt qu'une partition fixe

Exemple illustratif

Motivation : K-Means nous force à choisir K à l'avance. Le clustering hiérarchique propose une approche plus exploratoire.

Philosophie : On ne crée pas une seule partition, mais une **hiérarchie de partitions imbriquées**.

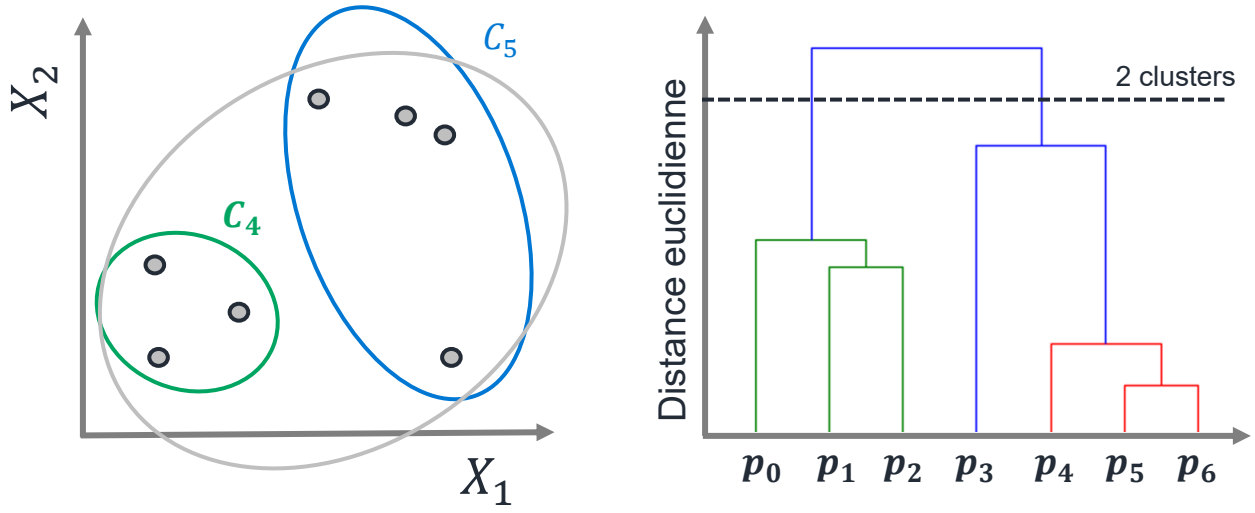
Approche la plus courante :

Agglomérative (Bottom-Up)

1. **Initialisation :** Chaque observation commence dans son propre cluster (on a n clusters).
2. **Itération :** À chaque étape, on trouve les deux clusters les plus "proches" et on les **fusionne**.
3. **Fin :** On répète ce processus n-1 fois, jusqu'à ce qu'il ne reste qu'un seul grand cluster contenant toutes les données.

Le résultat : Cet historique de fusions est stocké dans une structure en arbre appelée **dendrogramme**.

Itération t=5



Le clustering hiérarchique

Construire un arbre de relations plutôt qu'une partition fixe

Exemple illustratif

Motivation : K-Means nous force à choisir K à l'avance. Le clustering hiérarchique propose une approche plus exploratoire.

Philosophie : On ne crée pas une seule partition, mais une **hiérarchie de partitions imbriquées**.

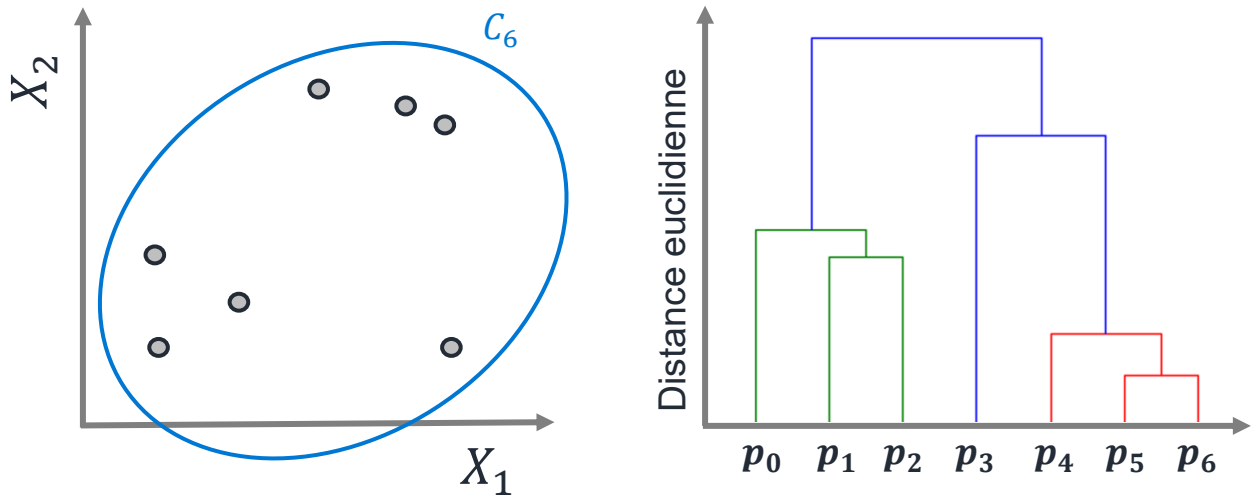
Approche la plus courante :

Agglomérative (Bottom-Up)

1. **Initialisation :** Chaque observation commence dans son propre cluster (on a n clusters).
2. **Itération :** À chaque étape, on trouve les deux clusters les plus "proches" et on les **fusionne**.
3. **Fin :** On répète ce processus n-1 fois, jusqu'à ce qu'il ne reste qu'un seul grand cluster contenant toutes les données.

Le résultat : Cet historique de fusions est stocké dans une structure en arbre appelée **dendrogramme**.

Itération t=5



Les méthodes de liage (linkage)

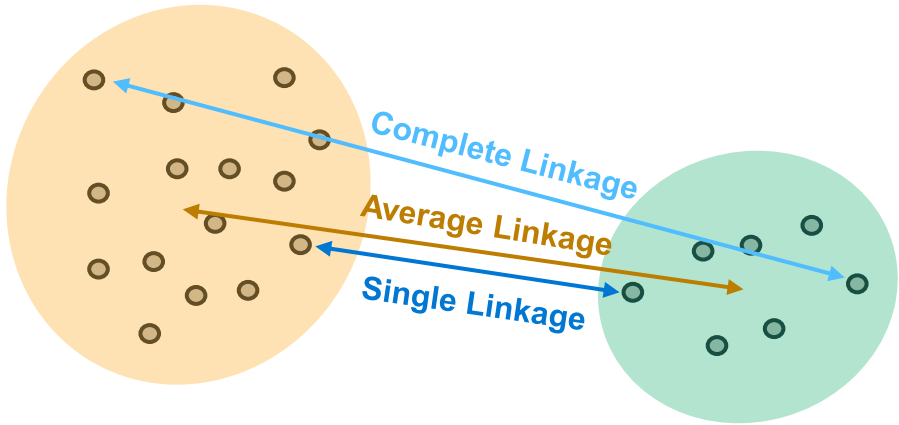
Comment mesurer la distance entre deux **clusters** d'observations ?

Exemple illustratif

Le choix du **critère de liage** a un impact majeur sur le résultat.

Quatre méthodes principales :

1. **Single Linkage (Lien simple – méthode « optimiste ») :** Distance entre les deux points les plus **proches** des deux clusters. Tendance à créer de longues chaînes ("chaining effect").
2. **Complete Linkage (Lien complet – méthode « pessimiste ») :** Distance entre les deux points les plus **éloignés**. Favorise des clusters très compacts et de même diamètre.
3. **Average Linkage (Lien moyen) :** Distance **moyenne** entre toutes les paires de points (un de chaque cluster). Un bon compromis.
4. **Ward's Linkage (Lien de Ward) :** Fusionne les deux clusters qui entraînent la **plus faible augmentation de l'inertie intra-cluster totale (WCSS)**. C'est une approche basée sur la variance (similaire à K-Means) et souvent la plus performante en pratique pour trouver des clusters bien séparés.



2. Clustering avancé : hiérarchie et densité

Le clustering hiérarchique

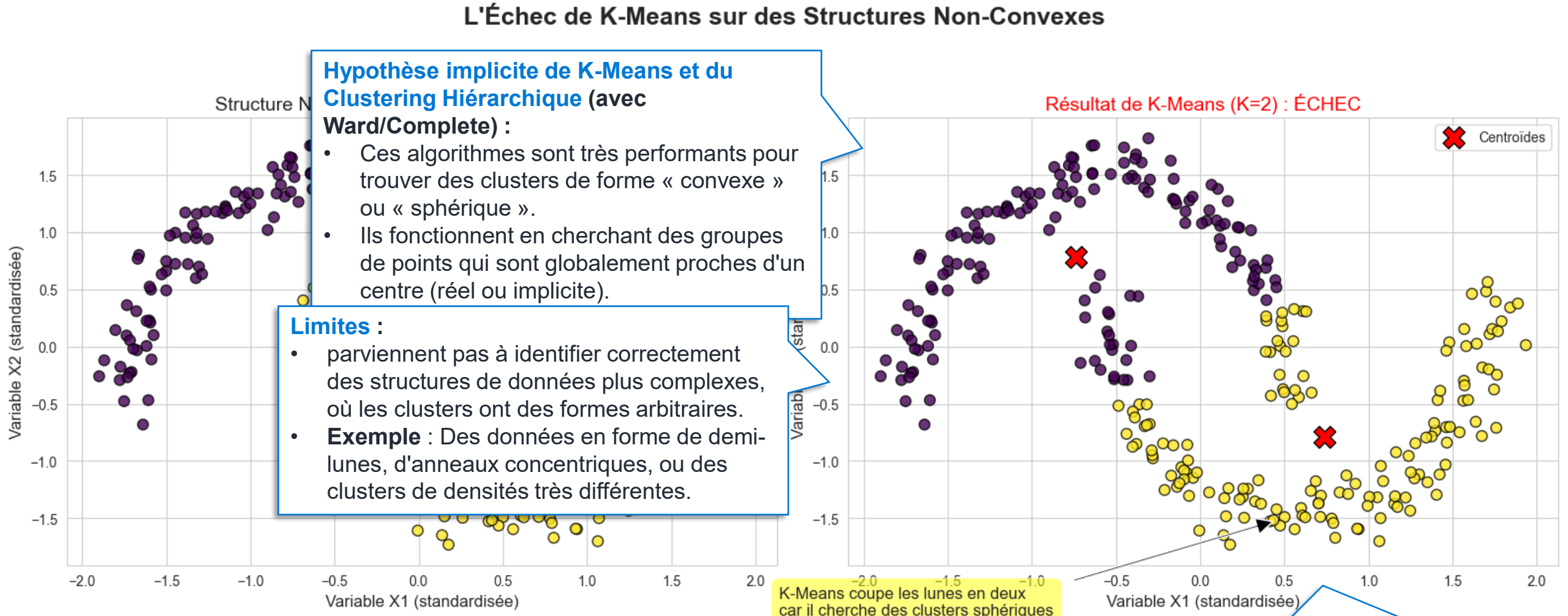
Le clustering par densité (DBSCAN)

Synthèse des méthodes de clustering

ISFA
Institut de Science Financière et d'Assurances

Les limites des approches basées sur la distance et les centroïdes

Que faire quand les clusters ne sont pas de forme "convexe" ?



Hypothèse implicite de K-Means et du Clustering Hiérarchique (avec Ward/Complete) :

- Ces algorithmes sont très performants pour trouver des clusters de forme « convexe » ou « sphérique ».
- Ils fonctionnent en cherchant des groupes de points qui sont globalement proches d'un centre (réel ou implicite).

Limites :

- parviennent pas à identifier correctement des structures de données plus complexes, où les clusters ont des formes arbitraires.
- **Exemple** : Des données en forme de demi-lunes, d'anneaux concentriques, ou des clusters de densités très différentes.

Besoin : Un algorithme qui ne se base pas sur la notion de « centre », mais sur la notion de « densité » et de « connectivité ».

ISFA
Institut de Science Financière et d'Assurances

Le clustering par densité : DBSCAN

Un cluster est une région dense de l'espace, séparée par du vide

Exemple illustratif

DBSCAN

(Density-Based Spatial Clustering of Applications with Noise) **:**

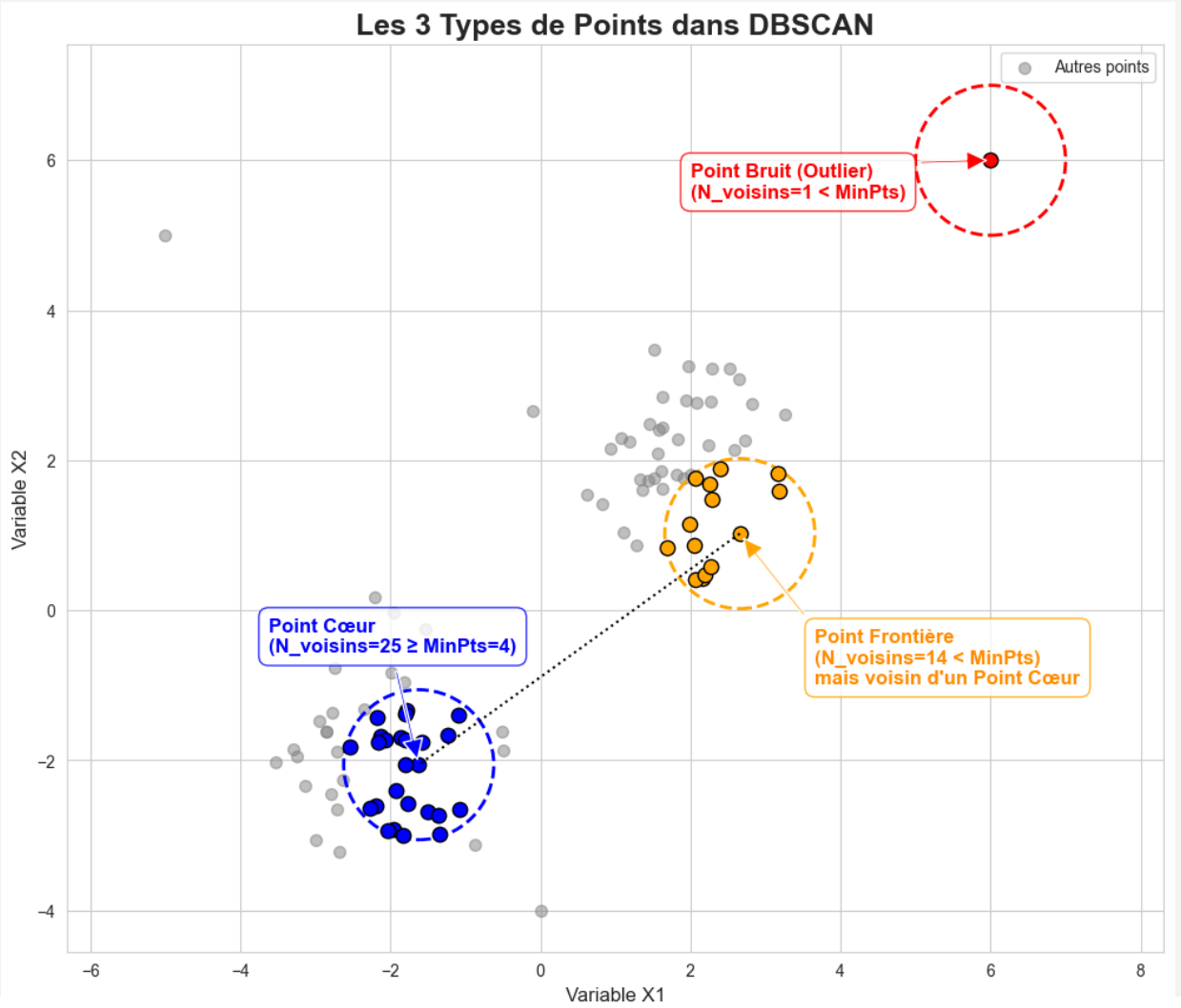
- Cette approche ne partitionne pas toutes les données.
- Elle identifie des « régions denses » et les considère comme des clusters. Le reste est considéré comme « bruit » (outliers).

Deux hyperparamètres clés :

- ε **(epsilon) :** Le rayon pour définir le « voisinage » d'un point. C'est une distance.
- **MinPts :** Le nombre minimum de points requis dans un voisinage ε pour considérer une région comme « dense ».

Classification des points :

- **Point Cœur :** Un point qui a au moins MinPts voisins (lui-même inclus) dans son rayon ε. Ce sont les « cœurs » des clusters.
- **Point Frontière :** Un point qui n'est pas un point cœur, mais qui est dans le voisinage d'un point cœur. Ce sont les « bords » des clusters.
- **Point Bruit (Outlier) :** Un point qui n'est ni cœur, ni frontière. Il est isolé.



ISFA Institut de Science Financière et d'Assurances

L'algorithme DBSCAN et ses avantages

Une approche robuste qui trouve des formes complexes

Exemple illustratif

Algorithme :

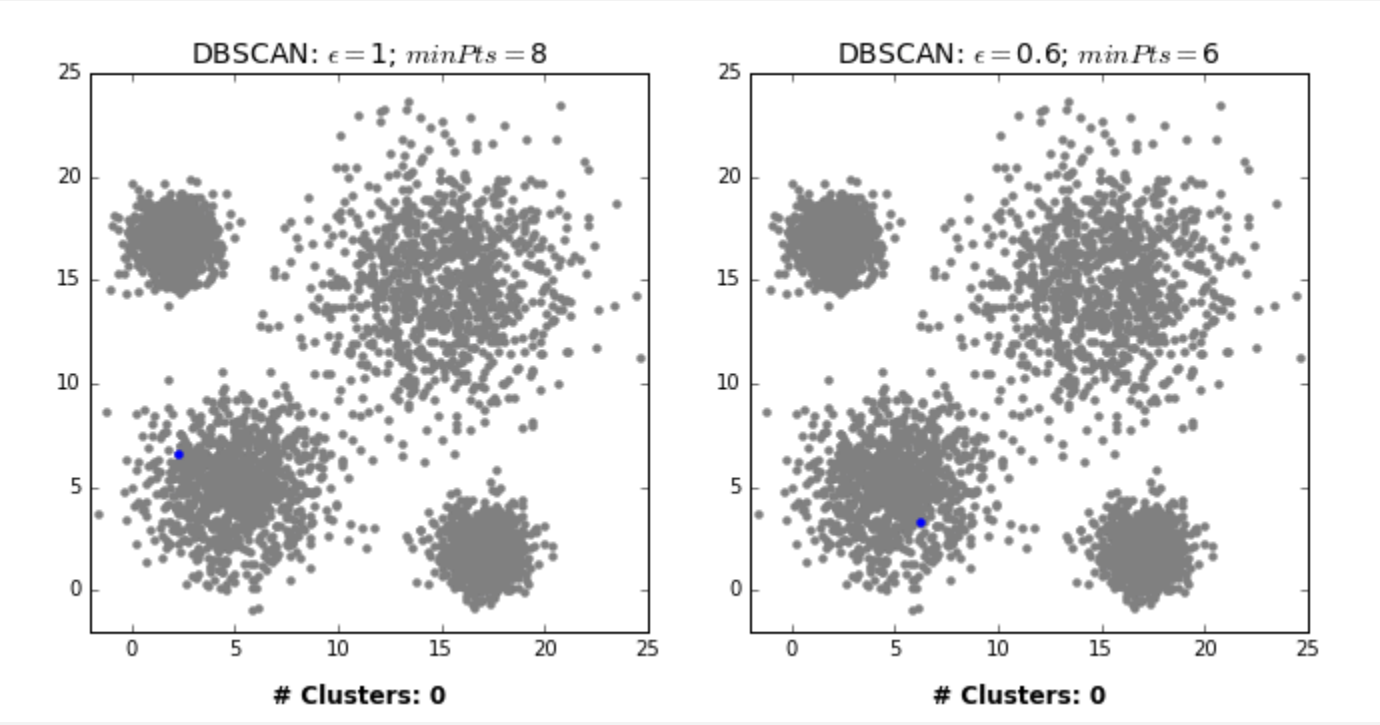
1. Pour chaque point, déterminer s'il est « cœur », « frontière » ou « bruit ».
2. Éliminer tous les points « bruit ».
3. Créer un graphe où l'on connecte tous les points cœurs qui sont voisins.
4. **Chaque composante connexe de ce graphe forme un cluster.**
5. Assigner chaque point frontière au cluster d'un de ses points cœurs voisins.

Avantages :

- **Pas besoin de spécifier le nombre de clusters K**.
- Peut trouver des **clusters de forme arbitraire** (non-convexe).
- **Robuste aux outliers**, qu'il identifie explicitement comme des « points bruit ».

Inconvénients :

- Le choix de ε et MinPts peut être délicat.
- Ne fonctionne pas bien si les clusters ont des densités très différentes.



Credit : David Sheehan

2. Clustering avancé : hiérarchie et densité

Le clustering hiérarchique

Le clustering par densité (DBSCAN)

Synthèse des méthodes de clustering

ISFA
Institut de Science Financière et d'Assurances

Synthèse : quel algorithme pour quel problème ?

Un guide pour choisir votre outil de segmentation

Caractéristique	K-Means	Clustering Hiérarchique	DBSCAN
Principe de base	Centroïde (Moyenne)	Distance (Fusion)	Densité (Connectivité)
Choix de K	**Requis** (hyperparamètre)	Non requis (déduit du dendrogramme)	Non requis (découvert)
Scalabilité	**Très bonne** $O(n)$	Faible $O(n^2 \log n)$	Moyenne $O(n \log n)$
Forme des clusters	Sphérique / Convexe	Dépend du liage	**Arbitraire**
Gestion des outliers	Sensible	Sensible	**Robuste** (les identifie)
Hyperparamètres	K	Méthode de liage, seuil de coupe	ε, MinPts

ISFA Institut de Science Financière et d'Assurances

3. Approche probabiliste et les modèles à variables latentes

- **Introduction au cadre probabiliste**
- Les modèles de mélange gaussien (GMM)

ISFA
Institut de Science Financière et d'Assurances

Les limites de l'assignation « dure »

Et si un point pouvait appartenir à plusieurs clusters ?

Les méthodes précédentes (K-Means, Hiérarchique, DBSCAN) :

- *hard assignment :* ces algorithmes effectuent une **assignation « dure »** une observation x_i appartient au cluster C_k ou n'y appartient pas. $x_i \in \{C_1, \dots, C_K\}$.
- Chaque observation est assignée à **100%** à un et un seul cluster.

Le problème :

- Cette vision binaire (appartient / n'appartient pas) est trop rigide.
- Que faire des observations qui se trouvent à la frontière entre deux clusters ? Leur assignation peut être arbitraire et instable.

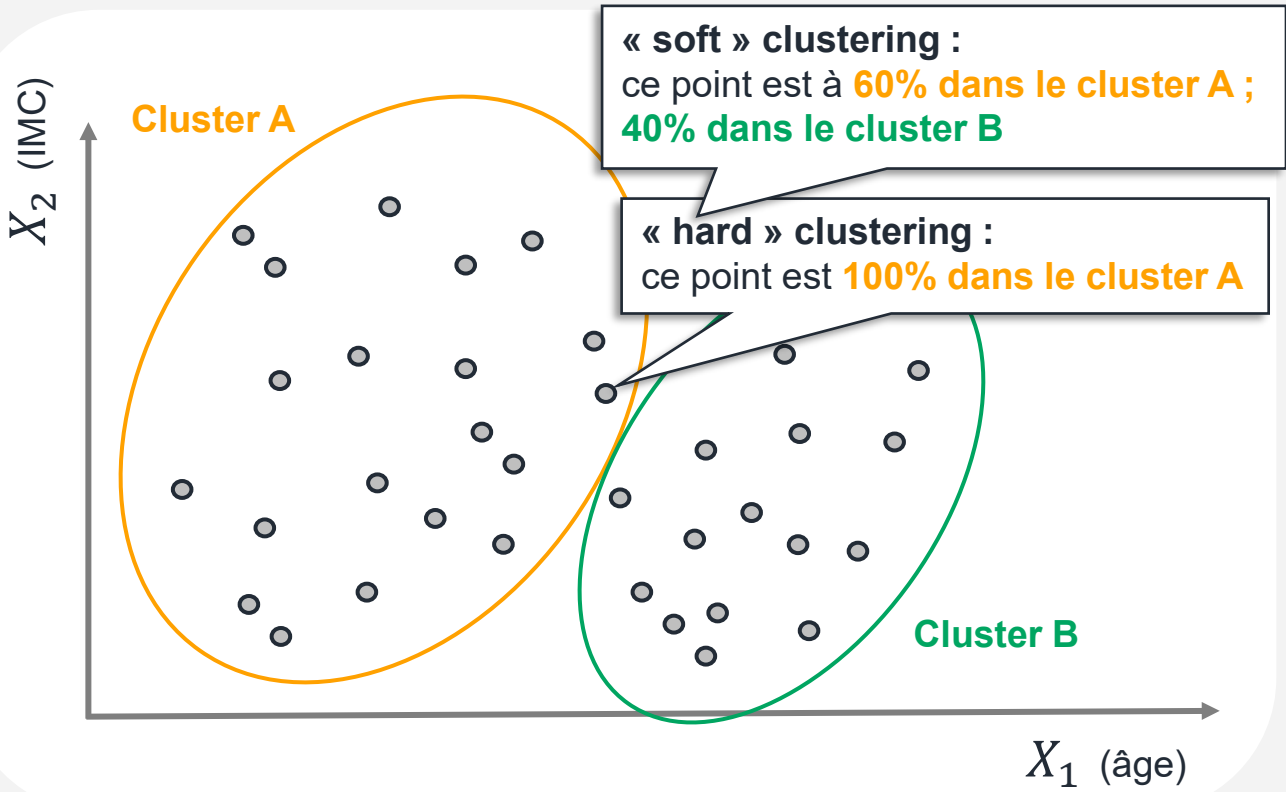
L'approche probabiliste : l'assignation « douce » (*soft assignment*)

Au lieu d'une décision binaire, on estime pour chaque observation x_i sa **probabilité d'appartenance** à chaque cluster k.

Ex : Pour un point x_i à la frontière, le modèle pourrait retourner :

$$P(x_i \in C_1) = 0.6, P(x_i \in C_2) = 0.4$$

Cette approche est plus nuancée et **capture mieux l'incertitude de l'assignation.**



Le concept de variable latente Z

Modéliser la structure cachée comme une variable non observée

Idée fondamentale : On formalise le clustering probabiliste en postulant un modèle génératif. On suppose que nos données observées X sont générées par un processus qui dépend d'une **variable latente** (cachée) Z.

Dans le cas du clustering :

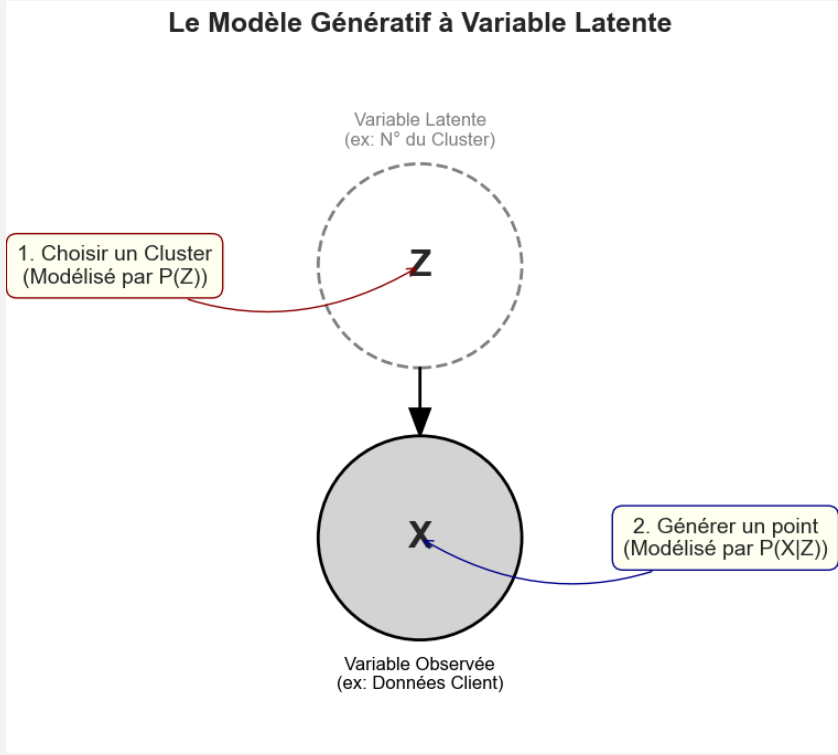
- La variable latente Z est une **variable catégorielle discrète**.
- Si on a K clusters, Z peut prendre les valeurs {1, 2, ..., K}.
- $Z_i = k$ signifie que l'observation x_i a été générée par le k-ième cluster.

Le modèle génératif :

- **Les poids des clusters :** On spécifie une distribution de probabilité pour la variable latente, $P(Z)$
- **La forme de chaque cluster** : On spécifie une distribution conditionnelle pour les données observées, $P(X|Z)$
- La distribution jointe est $P(X, Z) = P(X|Z) \times P(Z)$.

Objectif de l'apprentissage :

- **Estimer les paramètres** du modèle (ceux de $P(Z)$ et $P(X|Z)$).
- **Calculer la distribution a posteriori** $P(Z|X)$, qui correspond à notre « soft assignment ».



3. Approche probabiliste et les modèles à variables latentes

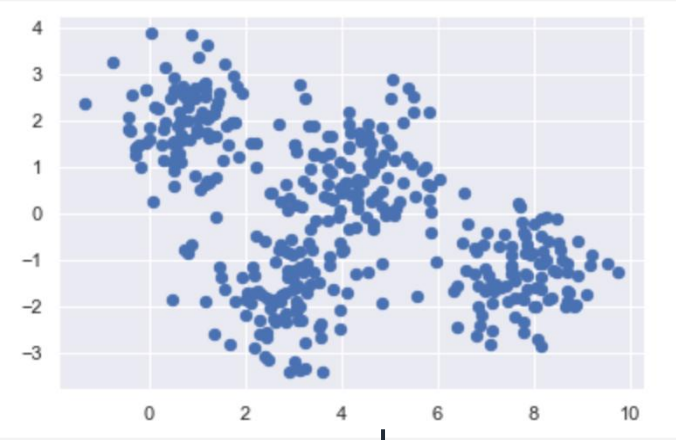
Introduction au cadre probabiliste

- **Les modèles de mélange gaussien (GMM)**

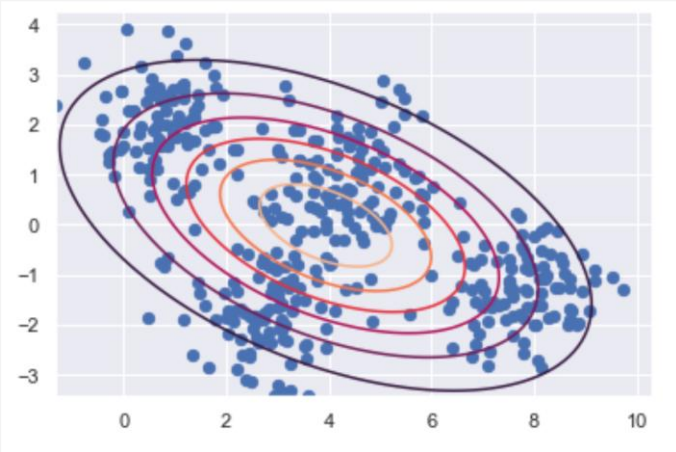
ISFA
Institut de Science Financière et d'Assurances

Le modèle de mélange gaussien (GMM)

L'hypothèse que les données sont un mélange de distributions normales

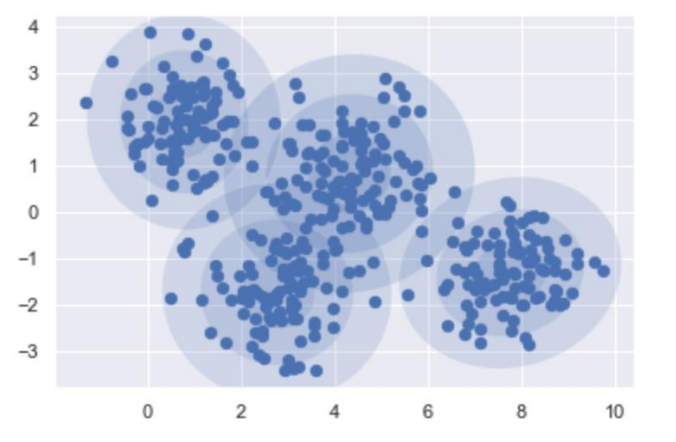


Entraîner une gaussienne sur les données …

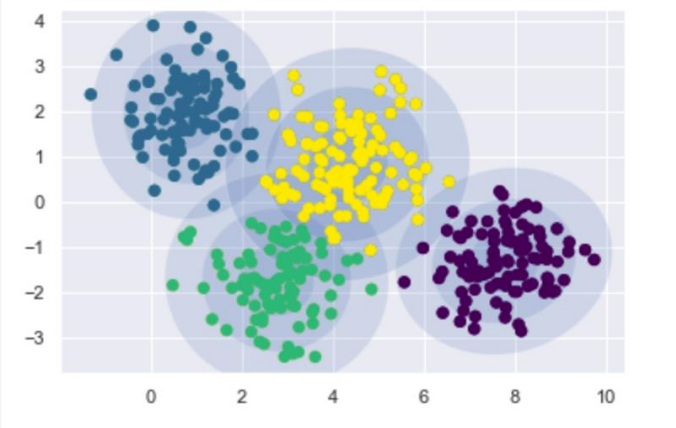


$N(\mu, \sigma)$

Entraîner un mélange gaussien !



Qui peut être utilisé pour faire un clustering



$$\sum_{k=1,\dots 4} \pi_k \times N(\mu_k, \sigma_k)$$

Les paramètres de notre GMM$\theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1,\dots,4}$

Comment apprendre ce modèle ?

Le modèle de mélange gaussien (GMM)

L'hypothèse que les données sont un mélange de distributions normales

Le **GMM** (Gaussian Mixture Model) est le modèle à variable latente le plus courant pour le clustering.

$$\pi_1 \mathcal{N}(\mu_1, \Sigma_1) + \pi_2 \mathcal{N}(\mu_2, \Sigma_2) + \pi_3 \mathcal{N}(\mu_3, \Sigma_3) + \pi_4 \mathcal{N}(\mu_4, \Sigma_4)$$

$$\text{parameters} : \theta = \{\pi_1, \mu_1, \Sigma_1, \pi_2, \mu_2, \Sigma_2, \pi_3, \mu_3, \Sigma_3, \pi_4, \mu_4, \Sigma_4\}$$

Hypothèse générative : Chaque observation x_i est générée par le processus suivant :

1. On choisit un cluster k avec une probabilité π_k (le poids du cluster).
2. On tire l'observation x_i de la distribution Normale multivariée associée à ce cluster : $x_i \sim N(\mu_k, \Sigma_k)$.

Les paramètres à estimer :

π_k : Poids du mélange pour chaque cluster k (avec $\Sigma\, \pi_k = 1$).

μ_k : Vecteur des moyennes du cluster k (son centre).

Σ_k : Matrice de covariance du cluster k (sa forme, sa taille et son orientation).

Flexibilité : Grâce à la matrice de covariance Σ_k, les GMM peuvent modéliser des clusters de forme **elliptique**, contrairement à k-means qui suppose des clusters sphériques (covariance $\sigma^2 I$).

Algorithme Espérance-Maximisation (EM)

Une approche itérative pour l'estimation par maximum de vraisemblance

Estimer les paramètres (π_k, μ_k, Σ_k) par maximum de vraisemblance est difficile à cause de la variable latente Z.
L'algorithme EM est l'outil standard pour résoudre ce problème. C'est une méthode itérative qui alterne deux étapes jusqu'à convergence :

1. **Étape E (Espérance) :** *« Deviner les clusters »*
   Étant donné les paramètres actuels du modèle, on calcule pour chaque point x_i la probabilité qu'il appartienne à chaque cluster k. C'est le « soft assignment », souvent noté r_{ik}.

$$p(t=2 \mid x, \theta) = \frac{p(x \mid t=2, \theta) \times p(t=2 \mid \theta)}{\text{Const}}$$

2. **Étape M (Maximisation) :** *« Mettre à jour le modèle »*
   Étant donné ces probabilités (considérées comme des poids), on met à jour les paramètres de chaque cluster k par des formules de maximum de vraisemblance pondérées.

$$p(x \mid t=2, \theta) = \mathcal{N}(x \mid \mu_{soft}^{MLE}, \Sigma_{soft}^{MLE})$$

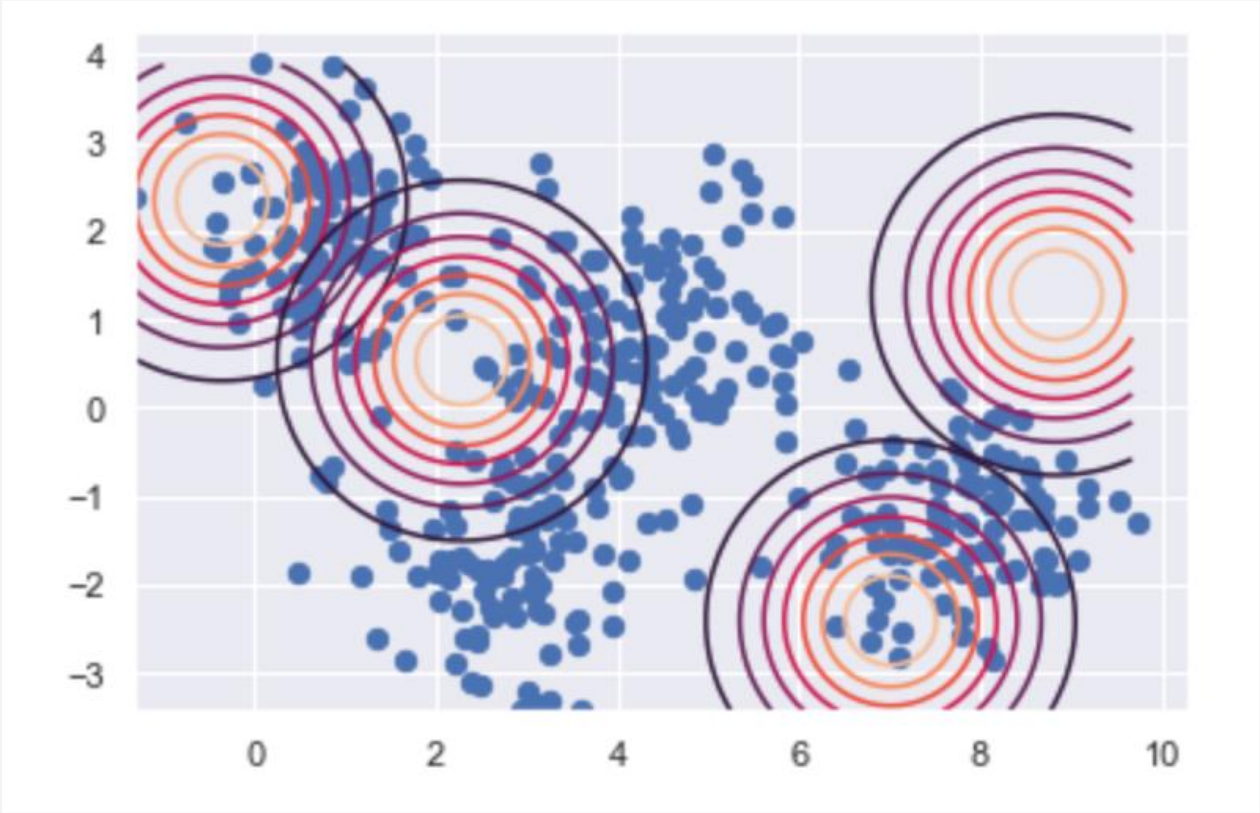
$$\mu_{soft}^{MLE} = \frac{\sum_i p(t=2 \mid x, \theta)\, x_i}{\sum_i p(t=2 \mid x, \theta)}$$

$$\Sigma_{soft}^{MLE} = \frac{\sum_i p(t=2 \mid x_i, \theta)\,(x_i - \mu_{soft}^{MLE}) \times (x_i - \mu_{soft}^{MLE})^T}{\sum_i p(t=2 \mid x_i, \theta)}$$

Algorithme Espérance-Maximisation (EM)

Une approche itérative pour l'estimation par maximum de vraisemblance

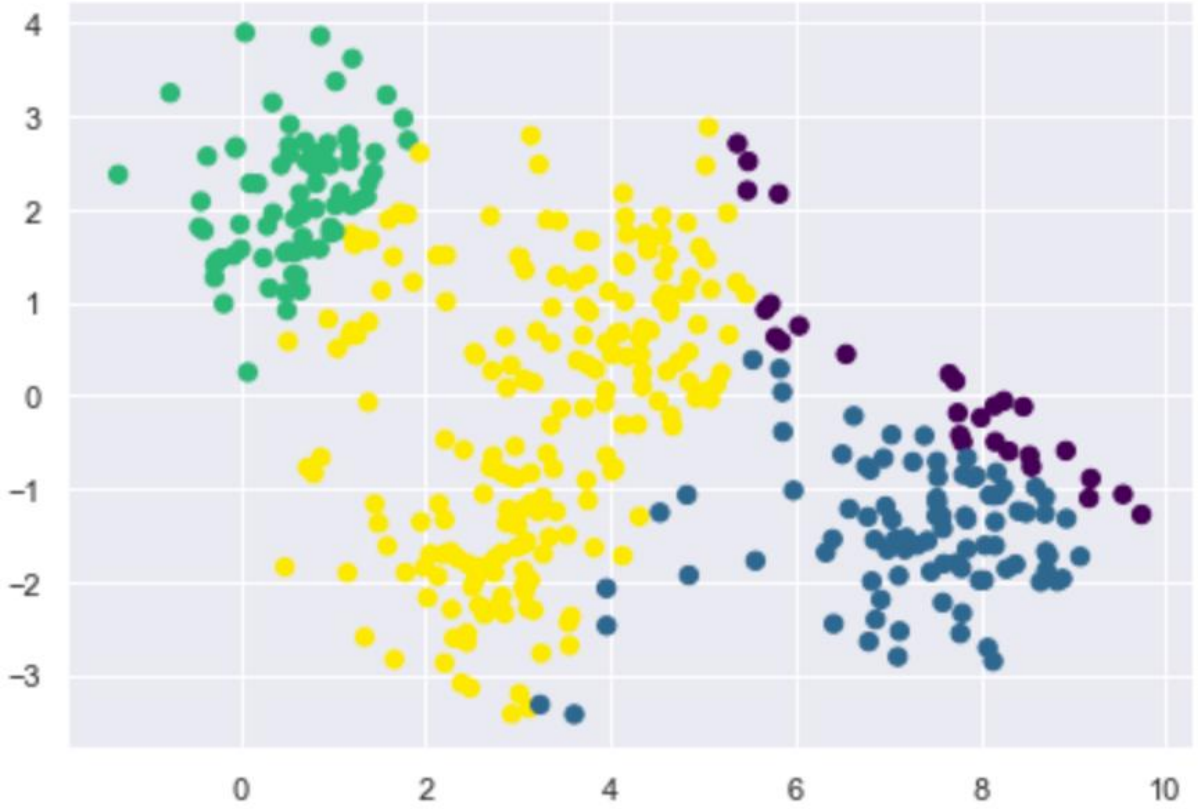
Initialisation

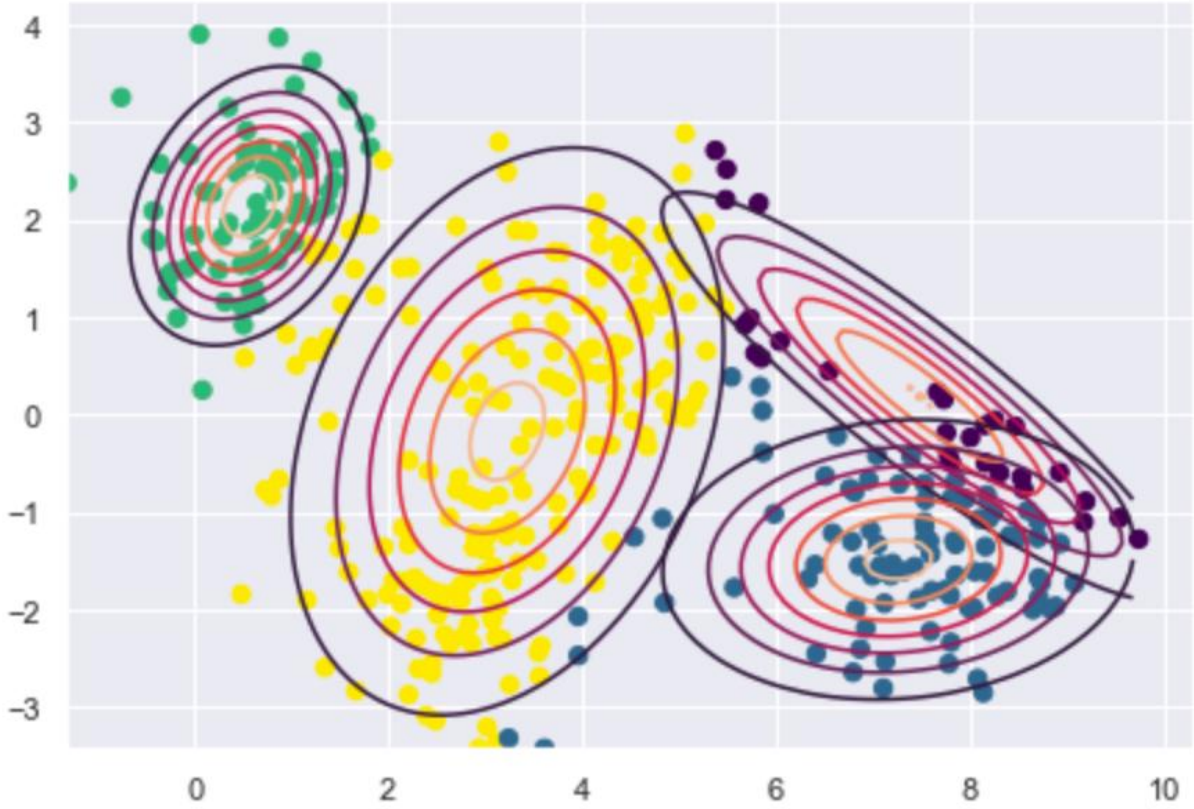


Algorithme Espérance-Maximisation (EM)

Une approche itérative pour l'estimation par maximum de vraisemblance

Etape 1

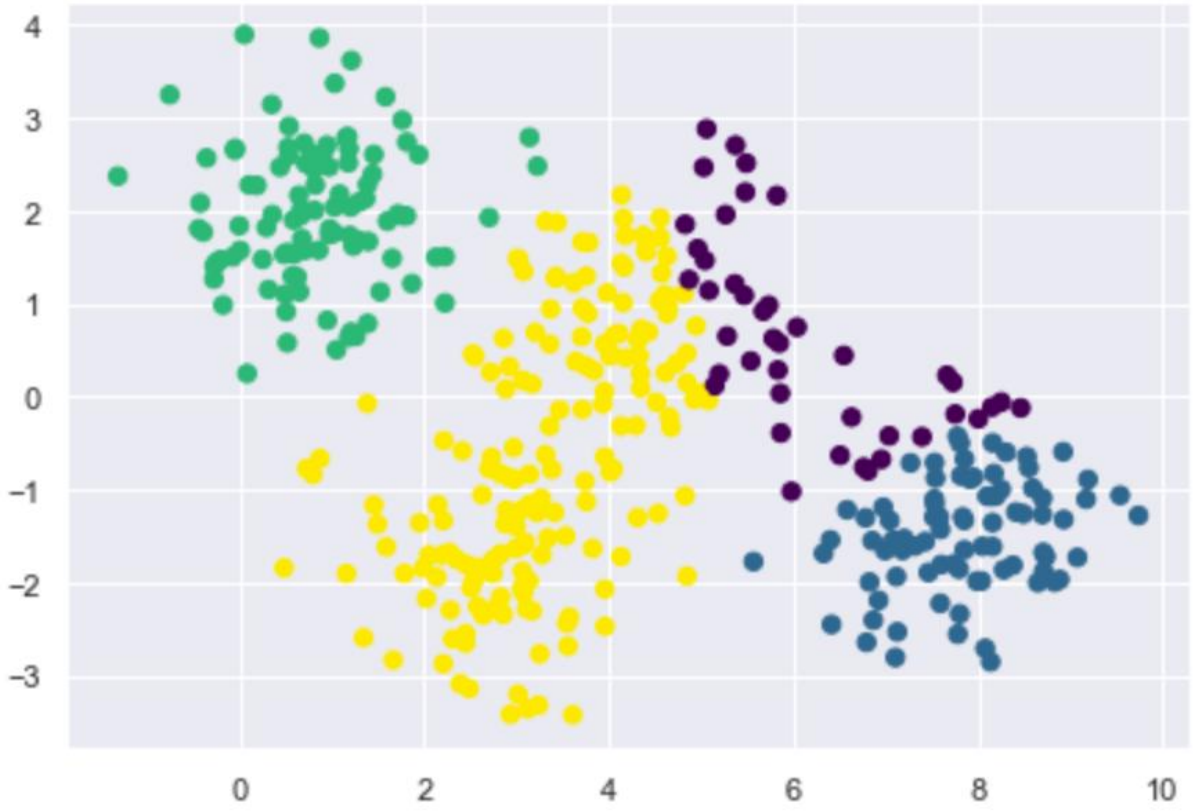


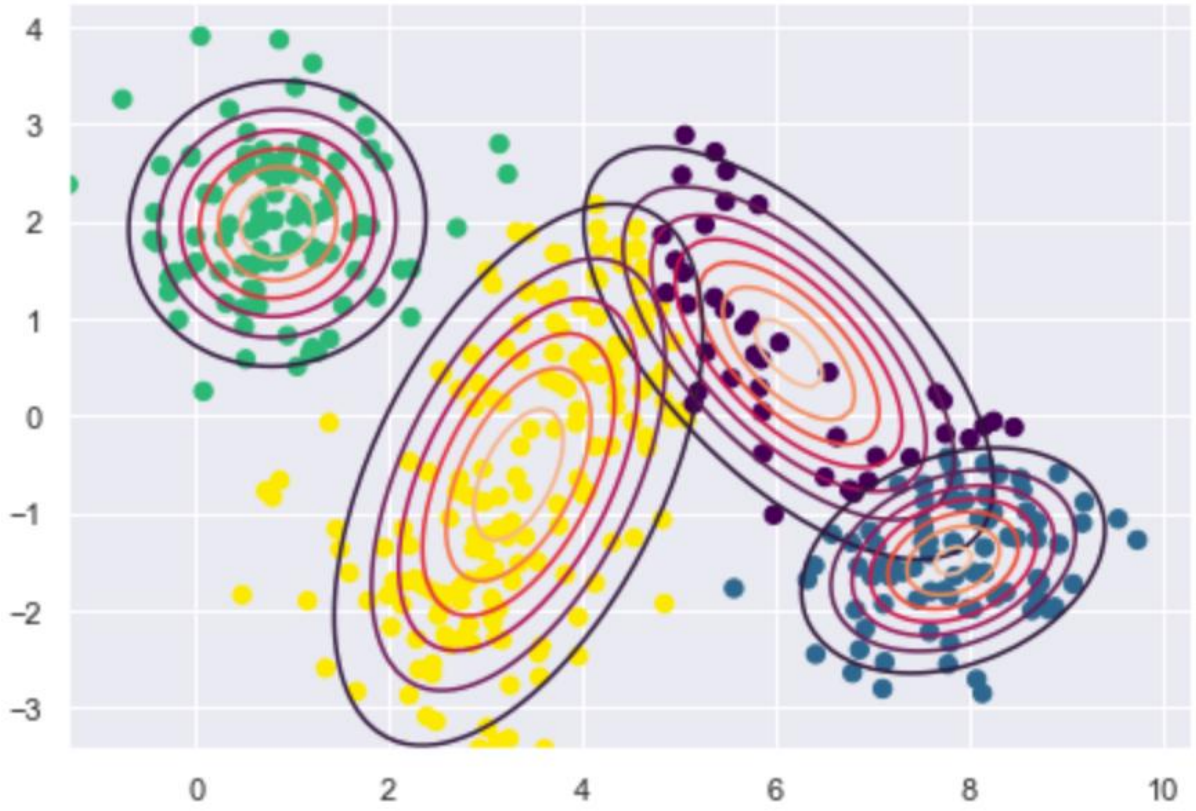


Algorithme Espérance-Maximisation (EM)

Une approche itérative pour l'estimation par maximum de vraisemblance

Etape 2

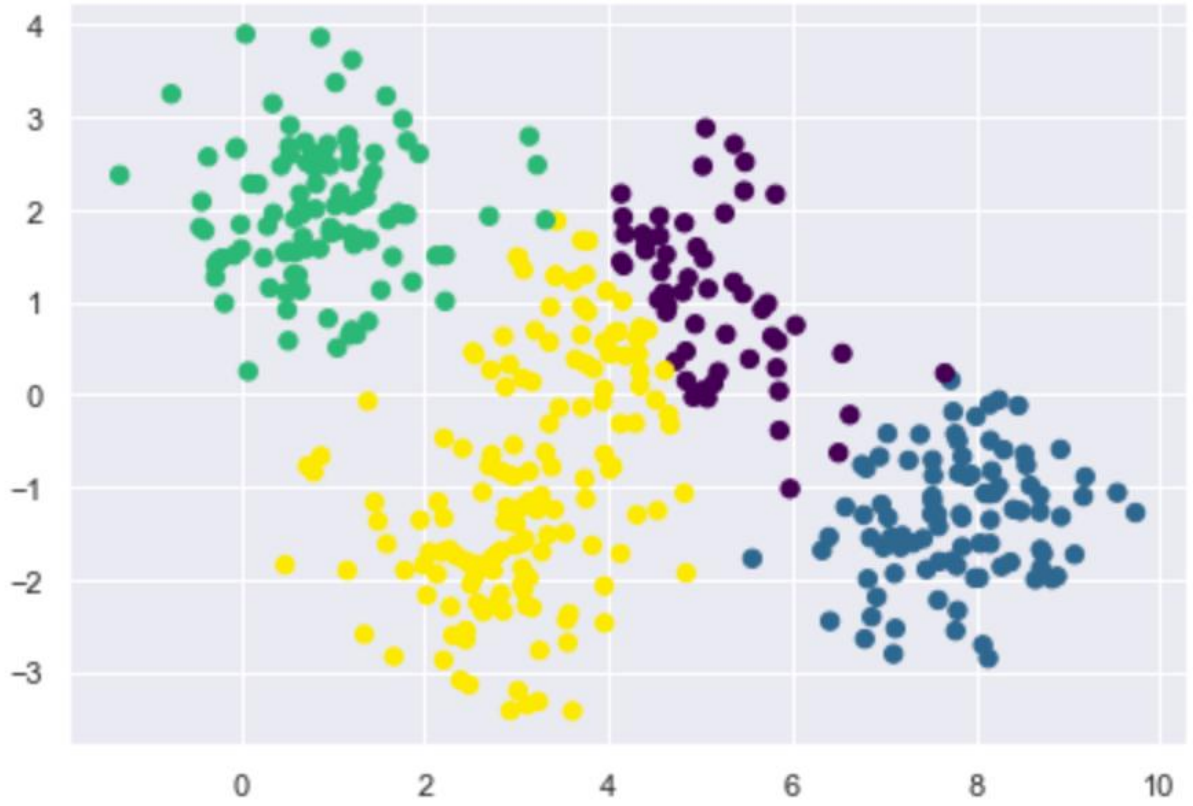


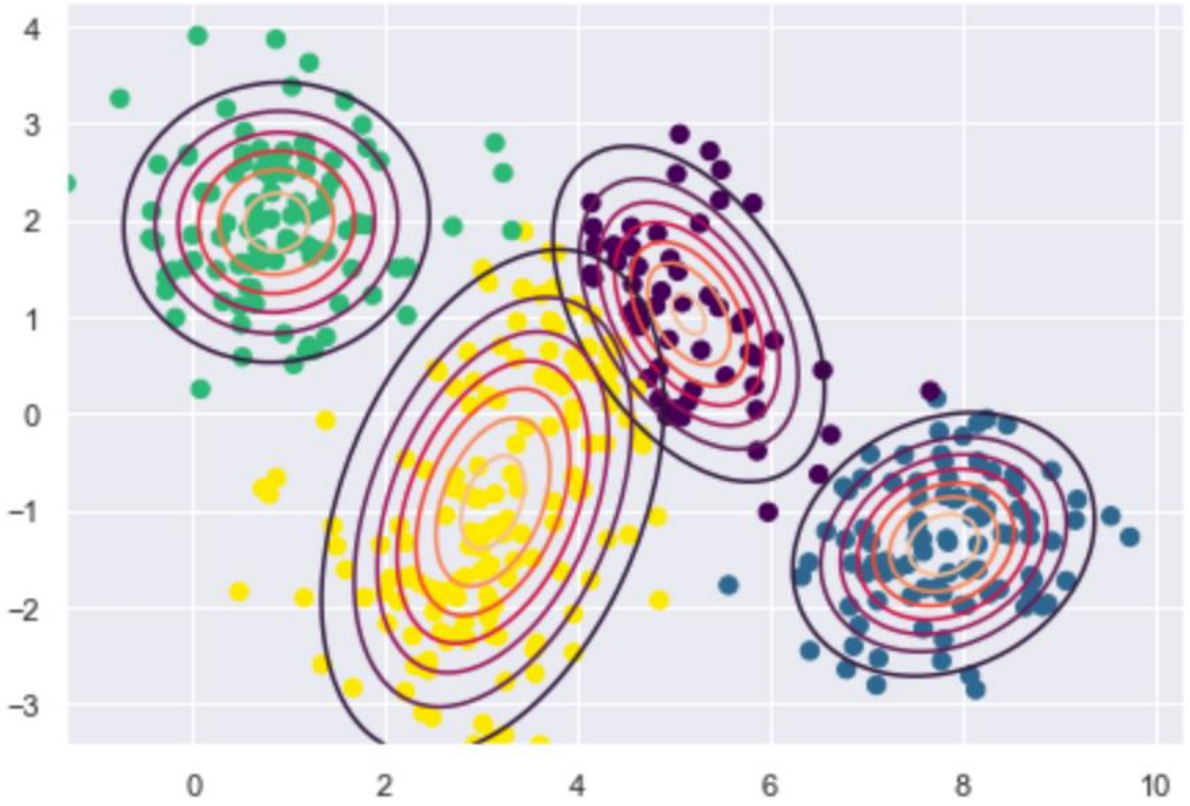


Algorithme Espérance-Maximisation (EM)

Une approche itérative pour l'estimation par maximum de vraisemblance

Etape 3

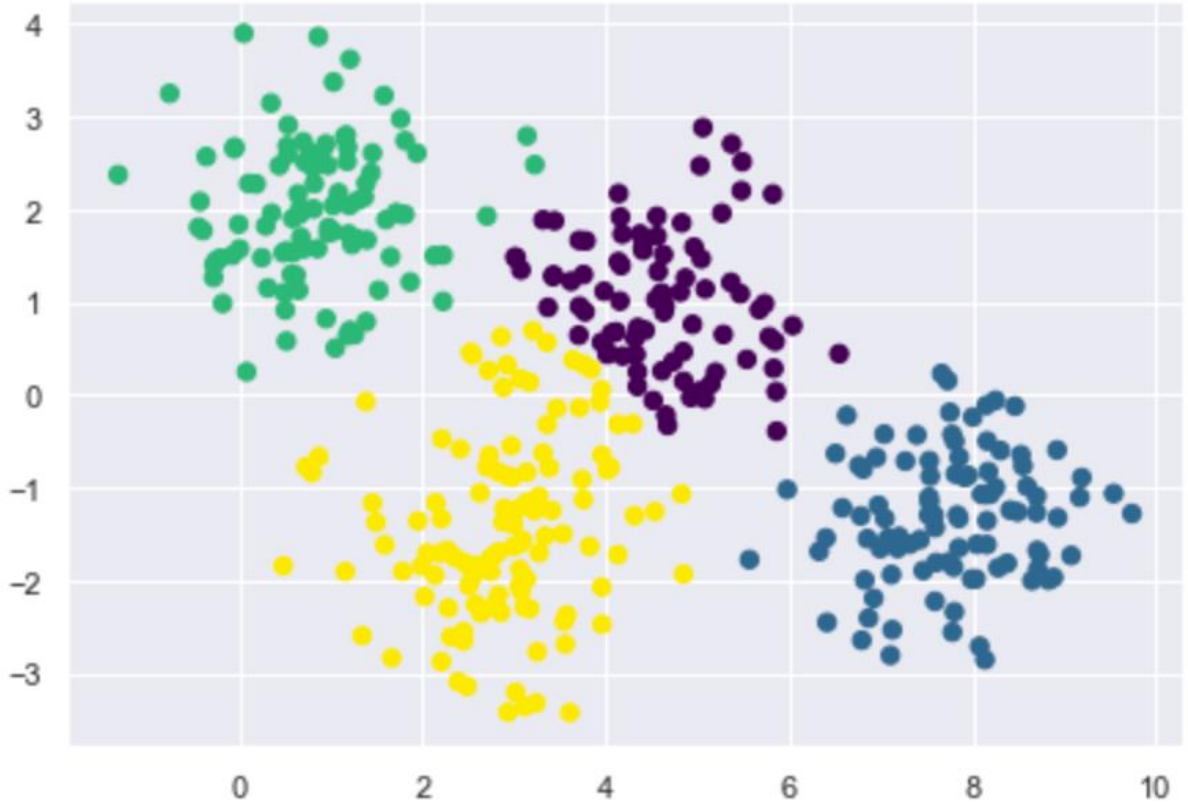


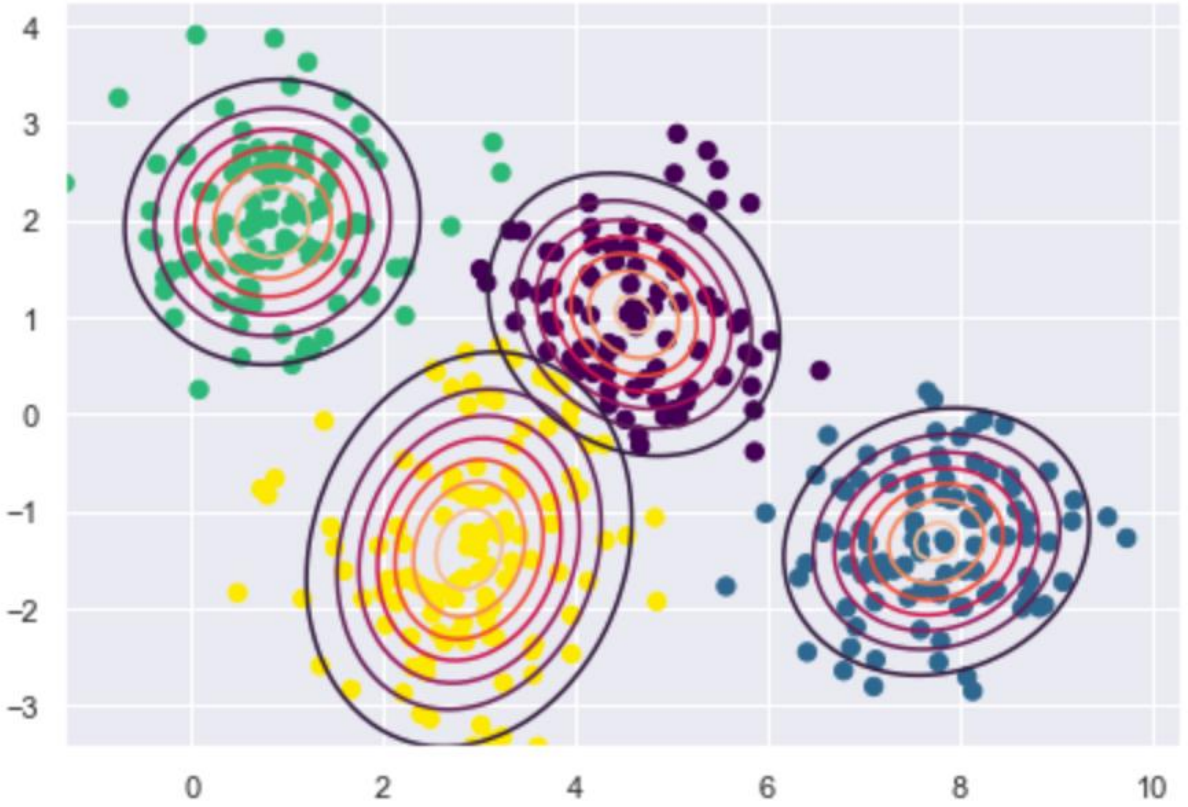


Algorithme Espérance-Maximisation (EM)

Une approche itérative pour l'estimation par maximum de vraisemblance

Etape 4

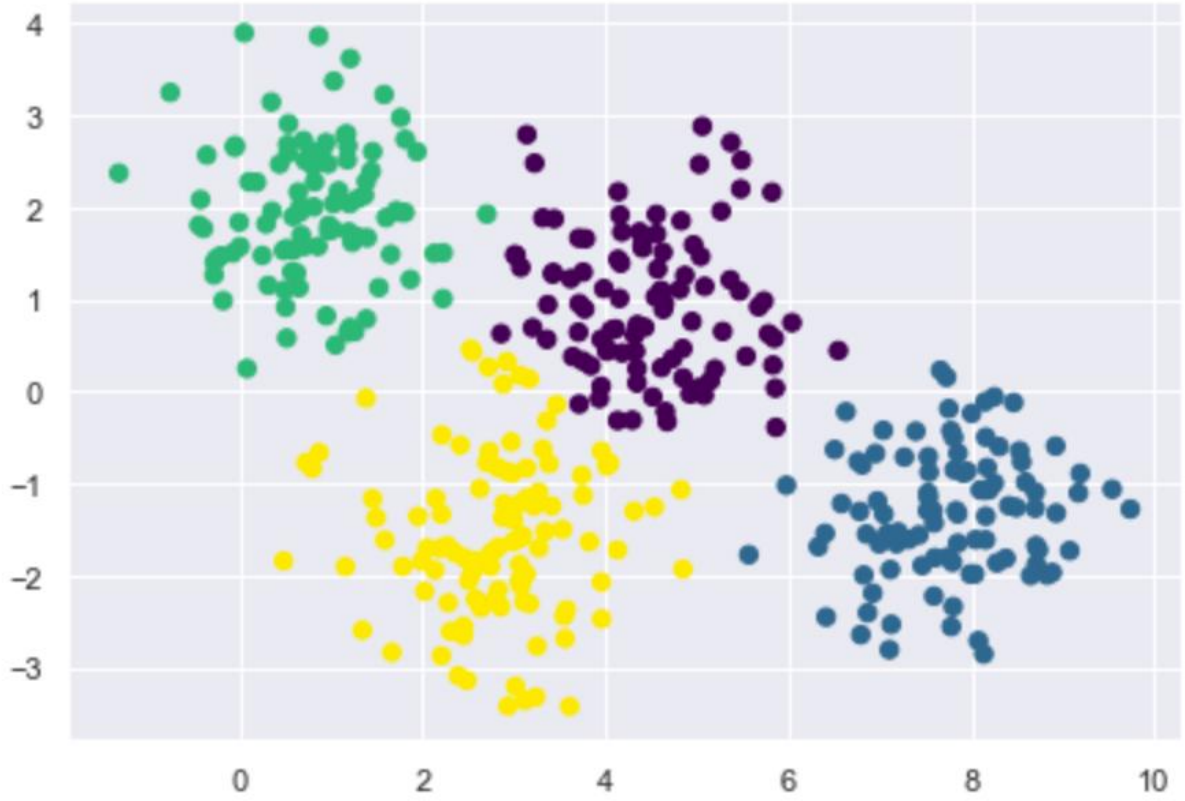


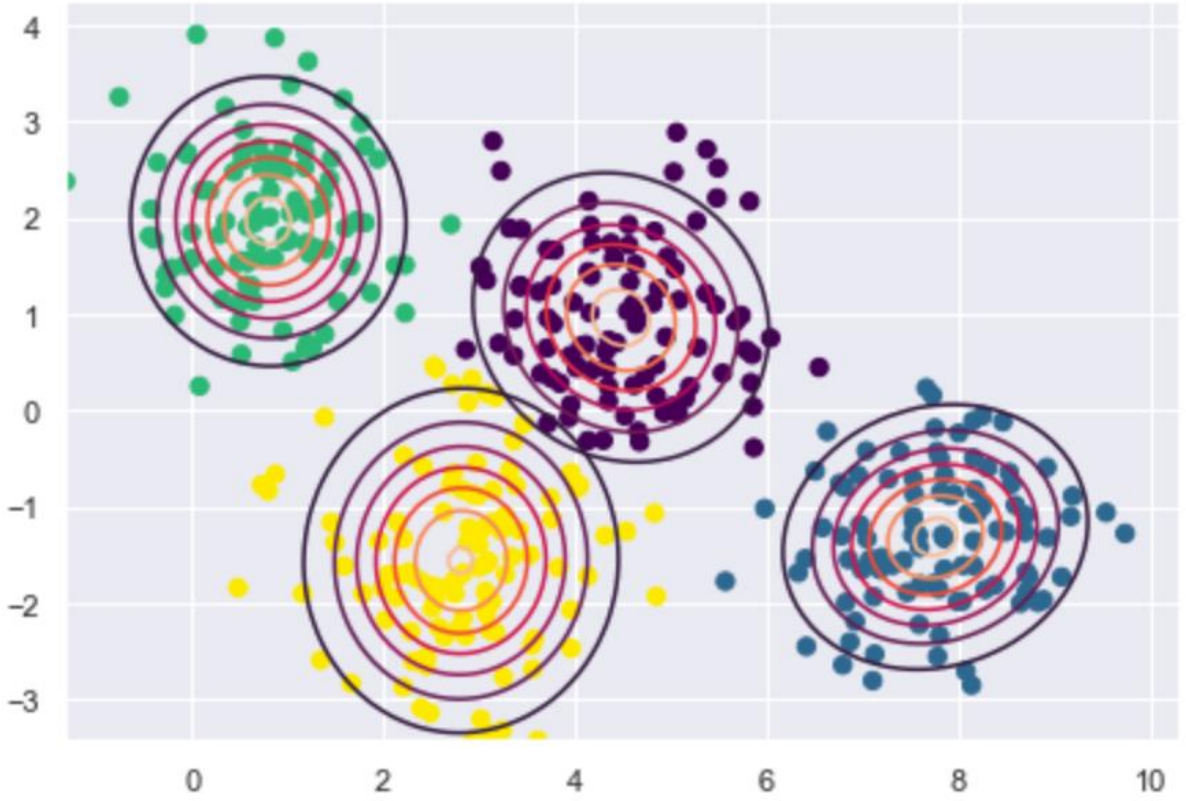


Algorithme Espérance-Maximisation (EM)

Une approche itérative pour l'estimation par maximum de vraisemblance

Etape 5

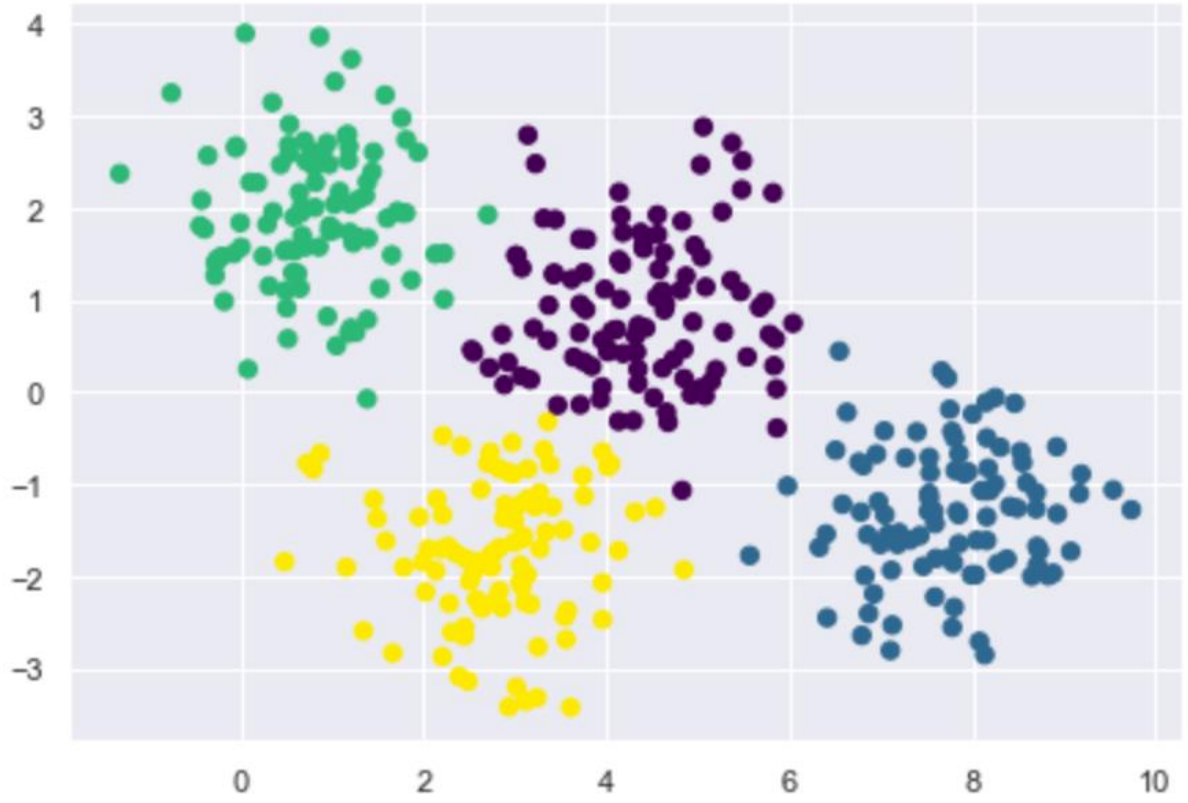


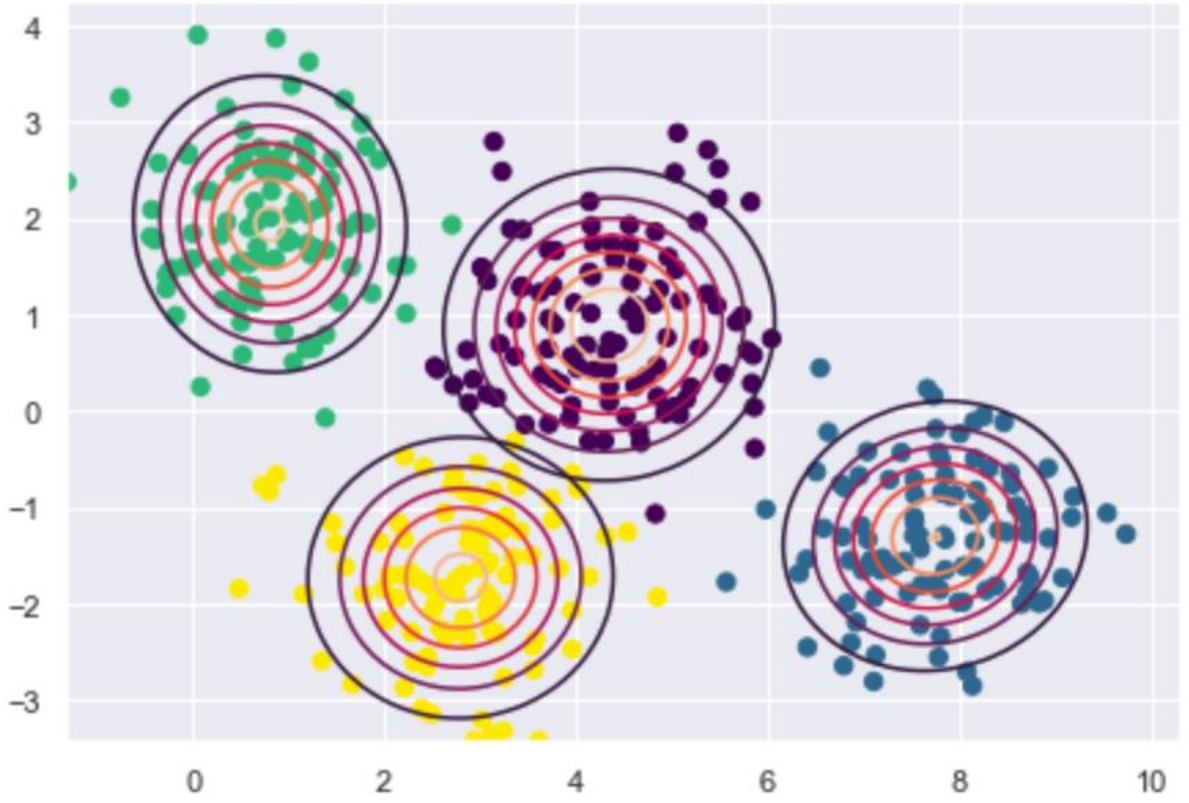


Algorithme Espérance-Maximisation (EM)

Une approche itérative pour l'estimation par maximum de vraisemblance

Etape 6





Analyse des Mélanges Gaussiens (GMM) en Perspective

Une approche statistique flexible pour le clustering

Caractéristique	K-Means	Clustering Hiérarchique	DBSCAN	Mélanges Gaussiens (GMM)
Principe de base	**Centroïde** (géométrique)	**Distance** (agglomératif)	**Densité** (connectivité)	**Probabiliste** (génératif)
Assignation	**Hard**	**Hard**	**Hard**	**Soft** (probabiliste)
Forme des Clusters	Sphérique	Dépend du liage	**Arbitraire**	**Elliptique**
Scalabilité	**Très bonne** $O(n)$			Faible $O(n * K * d^2)$
Choix de K	**Requis** (hyperparamètre)			Requis (**AIC, BIC**)
Gestion des Outliers	Sensible			Implicite (faible probabilité d'appartenance)

Comment choisir K pour un GMM ?

On utilise des critères qui balancent la qualité d'ajustement (vraisemblance) et la complexité du modèle (nombre de paramètres).

- **Critère d'Information Bayésien (BIC) :**
  - BIC = log(n) * (nombre de paramètres) - 2 * log(Vraisemblance)
  - Pénalise plus fortement la complexité. Souvent préféré pour le clustering car il favorise des modèles plus simples.
- **En pratique :** On entraîne des GMM pour K = 1, 2, ... et on choisit le K qui **minimise** le score BIC (ou AIC).

ISFA Institut de Science Financière et d'Assurances

Conclusion

Synthèse : l'apport de l'approche probabiliste au clustering

Ce qu'il faut retenir sur les modèles à variables latentes et les GMM

De l'Algorithme au modèle

- Nous sommes passés d'approches géométriques (K-Means, DBSCAN) à **un modèle statistique génératif.**
- L'idée clé est de supposer que la structure des clusters est gouvernée par une variable latente non observée.

Le « Soft clustering »

- La principale valeur ajoutée des GMM est l'**assignation douce** : chaque point a une **probabilité d'appartenance** à chaque cluster.
- Cela permet de quantifier l'incertitude et d'identifier les observations ambiguës, ce qui est une information précieuse en soi.

Des outils d'inférence

- L'algorithme **Espérance-Maximisation (EM)** est la méthode standard pour estimer les paramètres de ces modèles par **Maximum de Vraisemblance**.
- Le cadre probabiliste permet d'utiliser des **critères d'information (AIC/BIC)** pour sélectionner le nombre de clusters K de manière plus objective que les heuristiques.

Une Flexibilité accrue sur la forme des clusters

- En modélisant la matrice de covariance, les GMM peuvent capturer des clusters de forme **elliptique**, ce qui est souvent plus réaliste que les clusters sphériques de K-Means.

Merci

François HU

Francois.hu@milliman.com