Introduction aux méthodes de clustering

Objectif des méthodes de clustering

Le **clustering**, ou classification non supervisée, vise à diviser un ensemble de données en plusieurs groupes (clusters, catégories, segments...) de sorte que les données similaires soient regroupées ensemble et les données dissemblables soient séparées. Ce processus est également appelé partitionnement ou segmentation.
Les données sont structurées sous forme de tableau avec n lignes et p colonnes :

- Chaque ligne représente une observation (ou un individu).
- Chaque colonne représente une variable correspondant à une caractéristique (attribut, descripteur...) mesurée sur les observations.

Exemples d'applications dans divers domaines :

- Fouille de données : catégorisation automatique de documents (emails, textes, photos, etc.).
- Réseaux sociaux : détection de communautés.
- Marketing : identification de types de profils clients.
- Bioinformatique : regroupement de gènes similaires.
- Segmentation d'images : identification de régions homogènes dans une image.

Classification supervisée vs. non supervisée :

- **Supervisée** : Les classes sont connues à l'avance, et l'ensemble de données inclut des exemples étiquetés (paires (donnée, étiquette)). L'objectif est d'apprendre un modèle (à partir des données étiquetées) qui prédira la classe de nouvelles données non vues.
- **Non supervisée (clustering)** : Les classes et leur nombre sont inconnus. L'objectif est de découvrir des groupes (clusters) directement à partir des caractéristiques des données.

Pourquoi cette distinction est-elle importante ? La classification supervisée nécessite des données étiquetées, ce qui peut être coûteux et chronophage. Le clustering, en revanche, permet de découvrir des structures cachées dans les données sans connaissance préalable, ce qui est particulièrement utile pour l'exploration de données ou la segmentation de marché où les catégories ne sont pas définies à l'avance [**?**].

Formulation d'un problème de clustering

Définition formelle :

- **Entrée** : Un ensemble de données $\mathcal{X} = \{\mathbf{x_1}, \cdots, \mathbf{x_i}, \cdots, \mathbf{x_n}\}$.
- Chaque $\mathbf{x_i}$ est un vecteur $\begin{bmatrix} x_i^1 & \cdots & x_i^j & \cdots & x_i^p \end{bmatrix}^T$ dans $\mathbb{R}^p$ représentant les p caractéristiques d'une observation i.
- **Sortie** : Une partition de $\mathcal{X}$ en K clusters : $\mathcal{P} = \{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_K\}$.
- **Objectif** : Trouver la meilleure partition $\mathcal{P}$ selon un critère donné.

Rappel : Une partition de $\mathcal{X}$ est une collection de sous-ensembles non vides, deux à deux disjoints, dont l'union est égale à $\mathcal{X}$.
Complexité combinatoire : Le nombre de partitions possibles d'un ensemble de n éléments en k clusters est donné par le nombre de Stirling de deuxième espèce : $S(n,k) = \frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\mathrm{C}_j^k j^n$. Par exemple, $S(10,4) = 34,105$. Le nombre total de partitions de n éléments est donné par le nombre de Bell : $\mathcal{B}(n) = \sum_{k=1}^{n} S(n,k)$. Par exemple, $\mathcal{B}(10)$ est de l'ordre de 100 000. Cela rend l'optimisation exacte impossible pour des ensembles de données de taille moyenne ou grande [**?**].

Principales approches de clustering

Conséquence : Il n'est pas possible d'optimiser le critère donné sur toutes les partitions possibles. On utilise donc des méthodes itératives qui explorent seulement un sous-ensemble de l'espace des solutions, espérant trouver une partition quasi-optimale.
Principes généraux :

- Commencer avec une partition initiale.
- Améliorer itérativement la partition par rapport au critère en déplaçant les points de données d'un cluster à un autre.

Trois grandes familles d'algorithmes de clustering :

- **Méthodes combinatoires** : Basées sur des considérations géométriques. L'objectif est de regrouper les points de données proches selon une mesure de similarité ou de dissimilarité (distance).
- **Méthodes basées sur la densité** : Basées sur l'estimation de la densité locale des points de données. L'objectif est de regrouper les points de données situés dans des régions de haute densité (zones avec de nombreux points proches).
- **Méthodes probabilistes** : Basées sur la modélisation statistique de la population à l'aide d'un mélange de distributions. L'objectif est de regrouper les points de données qui sont probablement tirés de la même distribution de probabilité.

Limites des méthodes combinatoires : Les méthodes combinatoires (comme K-means) supposent implicitement que les clusters sont de forme convexe (sphères, ellipsoïdes) et peuvent échouer à détecter des clusters de formes arbitraires ou imbriqués. Les méthodes basées sur la densité, comme DBSCAN, permettent de détecter des clusters de formes complexes et de gérer le bruit et les valeurs aberrantes [**?**, **?**].

Données

Observations et variables

En pratique, les données sont généralement organisées dans un tableau de n lignes et p colonnes, c'est-à-dire une matrice $\mathbf{X}$ de taille (n, p) :

- Chaque ligne correspond à une observation i, représentée par un vecteur $\mathbf{x_i} \in \mathbb{R}^p$.
- Chaque colonne correspond à une variable j, représentée par un vecteur $\mathbf{x^j} \in \mathbb{R}^n$.

Types de variables :

- **Variables qualitatives (catégorielles)** : Doivent être transformées en variables numériques.
  - **Label encoding** : Assigne une valeur entière spécifique à chaque catégorie (ex: 0 = Faible, 1 = Moyen, 2 = Élevé). Idéal pour les données ordinales où les catégories ont un ordre naturel.
  - **One hot encoding** : Génère une nouvelle caractéristique binaire pour chaque catégorie. Idéal pour les données nominales sans ordre entre les catégories (comme les couleurs, les pays...), mais le nombre de catégories doit être faible.
- **Variables quantitatives** : Sont standardisées pour garantir que toutes les variables aient le même poids dans les algorithmes de clustering.

Standardisation : La standardisation consiste à transformer les variables pour qu'elles aient une moyenne nulle et un écart-type unitaire. Cela est crucial pour les algorithmes basés sur la distance (comme K-means), car sinon, les variables avec une plus grande échelle domineront le calcul des distances [**?**].

Distance entre chaque paire d'observations

Pour partitionner un ensemble de points, il faut pouvoir mesurer la similarité ou la dissimilarité entre les points. Cela nécessite de définir une distance entre chaque paire d'observations x_i et x_j.

Propriétés d'une distance :

- **Symétrie** : $d(x_i, x_j) = d(x_j, x_i)$.
- **Séparation** : $d(x_i, x_j) \geq 0$ et $d(x_i, x_j) = 0$ si $i = j$.
- **Inégalité triangulaire** : $d(x_i, x_j) \leq d(x_i, x_k) + d(x_k, x_j)$.

Choix de la distance : Il dépend de la nature des données.

Cas des données quantitatives réelles :

- **Distance de Minkowski** (norme L_q) :

$$d\left(\mathbf{x}_i, \mathbf{x}_l\right) = \left\|\mathbf{x}_i - \mathbf{x}_l\right\|_q = \left(\sum_{j=1}^{p} \left|x_i^j - x_l^j\right|^q\right)^{\frac{1}{q}}$$

  - **Distance euclidienne** ($q = 2$) : La plus couramment utilisée.

$$d\left(\mathbf{x}_i, \mathbf{x}_l\right) = \left\|\mathbf{x}_i - \mathbf{x}_l\right\|_2 = \sqrt{\sum_{j=1}^{p} \left(x_i^j - x_l^j\right)^2} = \sqrt{\left(\mathbf{x}_i - \mathbf{x}_l\right)^T \left(\mathbf{x}_i - \mathbf{x}_l\right)}$$

  - **Distance de Manhattan** ($q = 1$) :

$$d(\mathbf{x}_i, \mathbf{x}_l) = \left\|\mathbf{x}_i - \mathbf{x}_l\right\|_1 = \sum_{j=1}^{p} \left|x_i^j - x_l^j\right|$$

  - **Distance de Chebyshev** ($q = \infty$) :

$$d(\mathbf{x}_i, \mathbf{x}_l) = \left\|\mathbf{x}_i - \mathbf{x}_l\right\|_\infty = \sup_{1 \leq j \leq p} \left|x_i^j - x_l^j\right|$$

Distance de Mahalanobis : La distance de Mahalanobis est définie à partir d'une matrice $\mathbf{M}$ symétrique définie positive :

$$d_M(\mathbf{x}_i, \mathbf{x}_l) = \left\|\mathbf{x}_i - \mathbf{x}_l\right\|_M = \sqrt{(\mathbf{x}_i - \mathbf{x}_l)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_l)}.$$

- Si $\mathbf{M} = \mathbf{V}^{-1}$, où $\mathbf{V}$ est la matrice de covariance des variables, on obtient la distance de Mahalanobis.
- Cette distance prend en compte la covariance entre les variables, ce qui la rend particulièrement utile lorsque les variables sont corrélées ou ont des échelles différentes [**?**, **?**].

Cas des données discrètes quantitatives :

- **Distance de Hamming** : Nombre d'éléments différents entre deux vecteurs.
- **Indices de similarité** : Basés sur les quantités a_{il} (nombre d'éléments communs), b_{il} (nombre d'éléments dans x_i mais pas dans x_l), c_{il} (nombre d'éléments dans x_l mais pas dans x_i), d_{il} (nombre d'éléments ni dans x_i ni dans x_l).
  - **Coefficient de simple matching** : $d(x_i, x_l) = \frac{a_{il} + d_{il}}{p}$
  - **Indice de Jaccard** : $d(x_i, x_l) = \frac{a_{il}}{a_{il} + b_{il} + c_{il}}$

Distance de Levenshtein : Nombre d'opérations élémentaires (insertion/suppression/remplacement) nécessaires pour transformer une chaîne de caractères source en une chaîne cible. Par exemple, $d(\text{"}a\text{"}, \text{"}ab\text{"}) = 1$ (insertion de "b") [**?**].

Caractéristiques géométriques du nuage d'observations

Le nuage d'observations est caractérisé par son centre de gravité, sa matrice de covariance et son inertie.
Centre de gravité : Le centre $\mathbf{g}$ du nuage d'observations est le vecteur moyen des observations :

$$\mathbf{g} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x_i} = \begin{bmatrix} \bar{x}^1 \\ \bar{x}^2 \\ \vdots \\ \bar{x}^p \end{bmatrix}$$

- $\mathbf{g}$ représente une observation moyenne de la population.
- Chaque composante est la valeur moyenne de la variable j sur toutes les n observations : $\bar{x}^j = \frac{1}{n}\sum_{i=1}^{n} x_i^j$.

Matrice de covariance : La matrice de covariance $\mathbf{V}$ est définie (à un facteur $\frac{1}{n}$ près) par :

$$\mathbf{V} = \sum_{i=1}^{n}(\mathbf{x_i} - \mathbf{g})(\mathbf{x_i} - \mathbf{g})^T$$

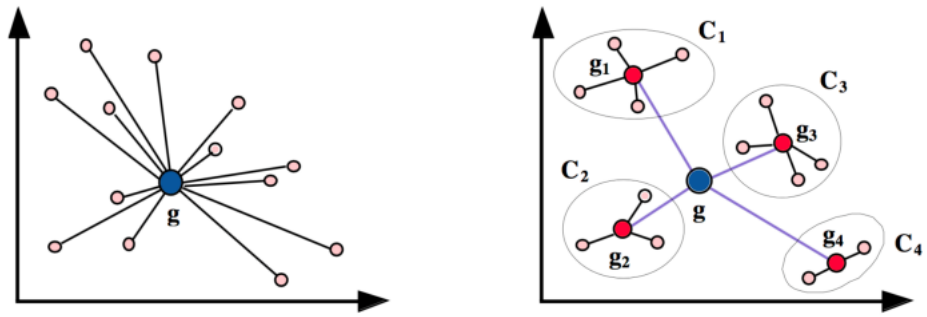
- $\mathbf{V}$ capture la dispersion de chaque variable et les relations entre les variables.
- Les éléments diagonaux V_{jj} représentent la variance de la variable j : $V_{jj} = \text{var}(\mathbf{x^j}) = \sum_{i=1}^{n}\left(x_i^j - \bar{x}^j\right)^2$.
- Les éléments hors diagonale V_{jk} représentent la covariance entre les variables j et $k \neq j$: $V_{jk} = \text{cov}(\mathbf{x^j}, \mathbf{x^k}) = \sum_{i=1}^{n}\left(x_i^j - \bar{x}^j\right)(x_i^k - \bar{x}^k)$.

Inertie totale du nuage d'observations : L'inertie totale du nuage d'observations est définie (à un facteur $\frac{1}{n}$ près) comme la somme des distances au carré entre les points et le centre de gravité :

$$I = \sum_{i=1}^{n} d^2(\mathbf{x_i}, \mathbf{g}) = \sum_{i=1}^{n}\|\mathbf{x_i} - \mathbf{g}\|^2$$

- L'inertie correspond à la généralisation de la variance au cas multivarié ; elle mesure la dispersion globale du nuage d'observations autour de son centre de gravité dans l'espace à p dimensions.
- Avec ces définitions, l'inertie est la trace de la matrice de covariance :

$$I = \sum_{i=1}^{n}\|\mathbf{x_i} - \mathbf{g}\|^2 = \sum_{j=1}^{p} V_{jj} = \text{trace}(\mathbf{V})$$



Inertie totale des points = Inertie Intra-cluster + Inertie Inter-cluster

Méthodes combinatoires

Principe des méthodes combinatoires

Les méthodes combinatoires cherchent à trouver la partition $\mathcal{P}$ de l'ensemble de données $\mathcal{X}$ en K clusters $\{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_K\}$ qui minimise une fonction de coût définie à partir d'un critère de similarité/dissimilarité sur $\mathcal{X}$.

Objectifs :

- Les observations les plus similaires sont assignées au même cluster.
- Les observations dissemblables sont placées dans des clusters distincts.
- Il n'y a pas de référence à un modèle probabiliste sous-jacent décrivant les données (contrairement aux méthodes probabilistes).

Caractéristiques géométriques d'un cluster

Comme pour l'ensemble du nuage d'observations, chaque cluster peut être caractérisé par son centre, sa matrice de covariance et son inertie.

Centre d'un cluster : Le centre $\mathbf{g}_k$ d'un cluster $\mathcal{C}_k$ de cardinalité $n_k = |\mathcal{C}_k|$ est défini comme le vecteur moyen des observations du cluster :

$$\mathbf{g}_k = \frac{1}{n_k} \sum_{\mathbf{x}_i \in \mathcal{C}_k} \mathbf{x}_i$$

Matrice de covariance d'un cluster : La matrice de covariance $\mathbf{V}_k$ d'un cluster $\mathcal{C}_k$ de centre $\mathbf{g}_k$ est définie par :

$$\mathbf{V}_k = \sum_{\mathbf{x}_i \in \mathcal{C}_k} (\mathbf{x}_i - \mathbf{g}_k)(\mathbf{x}_i - \mathbf{g}_k)^T$$

Inertie d'un cluster : L'inertie d'un cluster $\mathcal{C}_k$ est définie comme la somme des distances au carré entre les points du cluster et son centre $\mathbf{g}_k$:

$$I_k = \sum_{\mathbf{x}_i \in \mathcal{C}_k} d^2(\mathbf{x}_i, \mathbf{g}_k) = \sum_{\mathbf{x}_i \in \mathcal{C}_k} \|\mathbf{x}_i - \mathbf{g}_k\|^2$$

L'inertie peut aussi être obtenue à partir de la matrice de covariance :

$$I_k = \text{trace}(\mathbf{V}_k)$$

Critère à optimiser

Décomposition de l'inertie totale du nuage d'observations : Soit un nuage de centre $\mathbf{g}$ composé de n observations $\mathbf{x}_i$ réparties en K clusters $\mathcal{C}_1, \cdots, \mathcal{C}_K$. Chaque cluster $\mathcal{C}_k$ de centre $\mathbf{g}_k$ contient n_k observations. L'inertie totale peut être décomposée comme suit :

$$\begin{aligned}
I &= \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{g}\|^2 = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathcal{C}_k} \|\mathbf{x}_i - \mathbf{g}\|^2 \\
&= \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathcal{C}_k} \|\mathbf{x}_i - \mathbf{g}_k + \mathbf{g}_k - \mathbf{g}\|^2 \\
&= \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathcal{C}_k} \|\mathbf{x}_i - \mathbf{g}_k\|^2 + \|\mathbf{g}_k - \mathbf{g}\|^2 \\
&= \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathcal{C}_k} d^2(\mathbf{x}_i, \mathbf{g}_k) + \sum_{k=1}^{K} n_k d^2(\mathbf{g}_k, \mathbf{g})
\end{aligned}$$

Inertie totale = Inertie intra-cluster + Inertie inter-cluster :

$$I = I_W + I_B$$

- $I_W = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathcal{C}_k} d^2(\mathbf{x}_i, \mathbf{g}_k)$ est l'inertie intra-cluster. Elle représente la dispersion de tous les clusters. À minimiser pour obtenir des clusters aussi homogènes que possible.
- $I_B = \sum_{k=1}^{K} n_k d^2(\mathbf{g}_k, \mathbf{g})$ est l'inertie inter-cluster. Elle représente la séparation entre les clusters. À maximiser pour obtenir des clusters bien séparés.

Conséquence : Minimiser I_W est équivalent à maximiser I_B.
Formulation du problème d'optimisation : Pour un nombre de clusters K donné, l'objectif est de trouver la partition $\mathcal{P} = \{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_K\}$ qui minimise l'inertie intra-cluster :

$$I_W = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathcal{C}_k} d^2(\mathbf{x}_i, \mathbf{g}_k)$$

Algorithme K-means

Principe : K-means est un algorithme itératif de descente qui résout un problème d'optimisation élargi : il cherche à la fois les clusters $\{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_K\}$ et leurs centres $\{\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_K\}$ qui minimisent l'inertie intra-cluster I_W.
Étapes de l'algorithme :

1. Initialiser les centres $\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_K$ des K clusters.
2. Répéter :
   - Créer une nouvelle partition en assignant chaque point $\mathbf{x}_i$ au cluster dont le centre est le plus proche.
   - Mettre à jour le centre de chaque cluster $\mathcal{C}_k$ en calculant la moyenne des points qui lui sont assignés :
   $$\mathbf{g}_k = \frac{1}{n_k} \sum_{\mathbf{x}_i \in \mathcal{C}_k} \mathbf{x}_i$$
3. Jusqu'à convergence (plus de changement dans l'assignation des points ou inertie intra-cluster minimale).

Initialisation :

- Le nombre de clusters K est choisi a priori.
- Les K centres sont tirés aléatoirement ou de manière plus intelligente (par exemple, avec K-means++ pour éviter les clusters vides ou trop proches).

Critère d'arrêt :

- Nombre maximal d'itérations atteint.
- Convergence : soit la partition est stable (plus de changement dans l'assignation des observations aux clusters), soit l'inertie intra-cluster a atteint son minimum.

Propriétés de K-means :

- Convergence garantie en un nombre fini d'étapes, mais seulement vers un minimum local de l'inertie.
- Faible complexité computationnelle : $\mathcal{O}(nKN)$, où N est le nombre d'itérations. Cela permet de traiter de grands ensembles de données.
- Facile à interpréter.

- Le nombre de clusters K est fixé a priori.
- Sensible à l'initialisation, qui impacte fortement le résultat. En pratique, l'algorithme est exécuté plusieurs fois avec des initialisations différentes, et la meilleure partition (selon l'inertie intra-cluster) est conservée.
- Adapté aux clusters convexes (sphères, ellipsoïdes) de taille équilibrée.

Limites de K-means :

- **Sensibilité aux valeurs aberrantes** : Les points éloignés peuvent fausser la position des centroïdes.
- **Forme des clusters** : K-means suppose des clusters sphériques et peut mal performer sur des clusters de formes arbitraires ou de tailles très différentes.
- **Choix de K** : Le nombre de clusters doit être connu à l'avance, ce qui n'est pas toujours possible en pratique. Des méthodes comme l'elbow method ou le silhouette score sont souvent utilisées pour estimer K [**?**, **?**].

Clustering hiérarchique agglomératif

Principe : Les méthodes hiérarchiques produisent itérativement une hiérarchie de partitions imbriquées qui minimisent l'inertie intra-cluster. Il existe deux approches :

- **Agglomérative (bottom-up)** : Commence avec chaque observation dans son propre cluster. Fusionne itérativement les deux clusters les plus proches jusqu'à obtenir un seul cluster contenant toutes les données.
- **Divisive (top-down)** : Commence avec toutes les données dans un seul cluster. Divise itérativement les clusters jusqu'à ce que chaque observation soit dans son propre cluster. Moins utilisée, car le nombre de divisions possibles est plus grand que le nombre de fusions.

Algorithme du clustering hiérarchique agglomératif (AHC) :

1. Initialisation : Créer n clusters, chacun contenant une seule observation. Calculer la matrice des distances $\mathbf{M}$ entre toutes les paires de clusters.
2. Répéter :
   - Sélectionner dans $\mathbf{M}$ les 2 clusters $\mathcal{C}_k$ et $\mathcal{C}_m$ les plus proches selon une distance inter-clusters.
   - Fusionner $\mathcal{C}_k$ et $\mathcal{C}_m$ en un nouveau cluster $\mathcal{C}_a$.
   - Mettre à jour $\mathbf{M}$ avec les distances entre $\mathcal{C}_a$ et tous les autres clusters.
3. Jusqu'à ce que les 2 derniers clusters soient fusionnés.

Distances (ou critères de liaison) entre 2 clusters $\mathcal{C}_k$ et $\mathcal{C}_m$:

- **Single linkage** : Distance minimale entre deux points des clusters.

$$d_{\min}(\mathcal{C}_k, \mathcal{C}_m) = \min_{\mathbf{x}_i \in \mathcal{C}_k, \mathbf{x}_l \in \mathcal{C}_m} d(\mathbf{x}_i, \mathbf{x}_l)$$

  - Tendance à produire des clusters larges.
  - Sensible au bruit et aux valeurs aberrantes.

- **Complete linkage** : Distance maximale entre deux points des clusters.

$$d_{\max}(\mathcal{C}_k, \mathcal{C}_m) = \max_{\mathbf{x}_i \in \mathcal{C}_k, \mathbf{x}_l \in \mathcal{C}_m} d(\mathbf{x}_i, \mathbf{x}_l)$$

  - Tendance à produire des clusters compacts.

  - Sensible aux valeurs aberrantes.

- **Average linkage** : Distance moyenne entre tous les paires de points des clusters.

$$d_{\text{avg}}(\mathcal{C}_k, \mathcal{C}_m) = \frac{\sum_{\mathbf{x}_i \in \mathcal{C}_k} \sum_{\mathbf{x}_l \in \mathcal{C}_m} d(\mathbf{x}_i, \mathbf{x}_l)}{n_k n_m}$$

  - Tendance à produire des clusters plus homogènes.
  - Moins sensible aux valeurs aberrantes.

- **Centroid linkage** : Distance entre les centres des clusters.

$$d_{\text{cg}}(\mathcal{C}_k, \mathcal{C}_m) = d(\mathbf{g}_k, \mathbf{g}_m)$$

  - Moins sensible aux valeurs aberrantes.

- **Méthode de Ward** : Fusionne les deux clusters qui minimisent l'augmentation de l'inertie intra-cluster.

$$d_{\text{Ward}}(\mathcal{C}_k, \mathcal{C}_m) = \sqrt{\frac{n_k n_m}{n_k + n_m}} d(\mathbf{g}_k, \mathbf{g}_m)$$

  - Minimise la perte en inertie inter-cluster.
  - Tendance à produire des clusters de taille similaire.

Dendrogramme : Le dendrogramme est une représentation visuelle de la hiérarchie des clusters. Il permet de :

- Visualiser l'ordre de fusion des clusters.
- Choisir le nombre de clusters en "coupant" l'arbre à une certaine hauteur.
- Identifier les clusters naturels : les branches courtes indiquent des fusions de clusters similaires, tandis que les branches longues indiquent des fusions de clusters moins homogènes.

Propriétés du clustering hiérarchique agglomératif :

- Le nombre de clusters K n'est pas requis à l'avance.
- Résultat sous forme de dendrogramme visuel.
- Algorithme entièrement déterministe (pas de réévaluation des clusters fusionnés).
- Complexité computationnelle élevée : au moins n^2 (calcul des distances à la première itération).
- Sensible au bruit (selon le critère de liaison).

Comparaison K-means et AHC :

¡custom-element

data-json="

Critère	**K-means**	**AHC**
Complexité	Faible : $\mathcal{O}(nKN)$	Élevée : $\mathcal{O}(n^3)$
Interprétation	Facile	Visuelle (dendrogramme)
Nombre de clusters	Doit être fixé a priori	Pas besoin de fixer K a priori
Sensibilité à l'initialisation	Oui	Non
Forme des clusters	Convexe	Arbitraire (selon le critère de liaison)
Sensibilité au bruit	Oui	Dépend du critère de liaison

Choix du nombre de clusters K

Méthode du coude (Elbow method) :

- Tracer l'inertie intra-cluster I_W en fonction de K (courbe décroissante).
- Rechercher le "coude" où la courbe commence à s'aplatir, c'est-à-dire où l'ajout de clusters supplémentaires ne réduit pas significativement I_W.
- Ce point est considéré comme le nombre optimal de clusters.

Méthode de la silhouette :

- Le score de silhouette mesure la cohésion (similarité intra-cluster) et la séparation (dissimilarité inter-cluster).
- Tracer le score de silhouette en fonction de K.
- Rechercher la valeur maximale, qui indique le nombre optimal de clusters, reflétant le plus haut degré de cohésion et de séparation.

Méthode contextuelle :

- Choisir un nombre de clusters pour lequel les clusters peuvent être interprétés de manière significative dans le contexte de l'application.

Exemple visuel : Il est souvent utile de combiner les méthodes du coude et de la silhouette pour valider le choix de K. Par exemple, si la méthode du coude suggère $K = 3$ et que le score de silhouette est maximal pour $K = 3$, cela renforce la confiance dans ce choix [**?**, **?**].

Méthodes basées sur la densité

Limitations des méthodes précédentes

Les méthodes combinatoires (comme K-means et le clustering hiérarchique) ont des limites :

- Elles ne peuvent pas trouver des clusters de formes arbitraires et de tailles variées (par exemple, des cercles imbriqués).
- Elles supposent implicitement que les clusters sont de forme convexe (sphères, ellipsoïdes).

Exemple visuel : Les méthodes comme K-means et le clustering hiérarchique (avec la méthode de Ward) échouent souvent à détecter des clusters de formes non convexes, comme des cercles concentriques ou des spirales. DBSCAN, en revanche, est conçu pour gérer ces cas [**?**, **?**].

Méthodes de clustering basées sur la densité

Solution : Les méthodes de clustering basées sur la densité, comme DBSCAN (Density-Based Spatial Clustering of Applications with Noise), permettent de former des clusters de formes non convexes.
Approche :

- Hypothèse : Les clusters correspondent à des régions de haute densité (contenant beaucoup d'observations).
- Principe : Estimer la densité de la région autour de chaque observation et étendre les clusters à partir des observations situées dans des régions de haute densité.
- Avantages : Détection de clusters de formes arbitraires et identification des valeurs aberrantes comme du bruit.

Définitions

ϵ-voisinage : Soit ϵ un nombre réel positif. Le ϵ-voisinage d'un point $\mathbf{x}_i$ dans un ensemble de données $\mathcal{X}$ est le sous-ensemble $\mathcal{V}_\epsilon(\mathbf{x}_i)$ tel que :

$$\mathcal{V}_\epsilon(\mathbf{x}_i) = \{\mathbf{x}_l \in \mathcal{X}; d(\mathbf{x}_i, \mathbf{x}_l) < \epsilon\}$$

Types de points :

- **Point central (core point)** : Un point $\mathbf{x}_i$ situé dans une région de haute densité. Son ϵ-voisinage contient au moins un nombre seuil de points $n_{\min}$:

$$|\mathcal{V}_\epsilon(\mathbf{x}_i)| \geq n_{\min}$$

- **Point frontière (border point)** : Un point $\mathbf{x}_i$ situé à la frontière d'une région de haute densité. Il appartient au ϵ-voisinage d'un point central, mais n'est pas lui-même un point central.
- **Point de bruit (noise/outlier point)** : Un point isolé, ni central ni frontière.

Accessibilité par la densité : Un point $\mathbf{x}_i$ est accessible par la densité depuis un point $\mathbf{x}_l$ s'il existe une séquence de ϵ-voisinages, chacun contenant au moins $n_{\min}$ points, les reliant.

Algorithme DBSCAN

Principe : DBSCAN (Density-Based Spatial Clustering of Applications with Noise) est un algorithme de clustering basé sur la densité. Il regroupe les points proches et marque les points isolés comme du bruit.
Étapes :

1. Pour chaque point $\mathbf{x}_i$, calculer son ϵ-voisinage $\mathcal{V}_\epsilon(\mathbf{x}_i)$.
2. Si $|\mathcal{V}_\epsilon(\mathbf{x}_i)| \geq n_{\min}$, $\mathbf{x}_i$ est un point central.
3. Étendre le cluster en ajoutant tous les points accessibles par la densité depuis $\mathbf{x}_i$.
4. Répéter jusqu'à ce que tous les points soient visités.

Paramètres :

- ϵ : Rayon du voisinage.
- $n_{\min}$: Nombre minimal de points pour former un cluster.

Avantages :

- Pas besoin de spécifier le nombre de clusters a priori.
- Capable de détecter des clusters de formes arbitraires.
- Robuste aux valeurs aberrantes.

Limites :

- Sensible au choix des paramètres ϵ et $n_{\min}$.
- Peut mal performer si les densités des clusters varient fortement.

Exemple visuel : DBSCAN est particulièrement efficace pour des données avec des clusters de formes complexes et du bruit. Par exemple, il peut détecter des clusters en forme de cercle ou de spirale, là où K-means échoue [**?**, **?**].

Métriques de performance en clustering

Introduction

Évaluer la qualité d'un clustering est essentiel pour choisir le bon algorithme et les bons paramètres. Plusieurs métriques existent, selon que l'on dispose ou non d'étiquettes de référence.

Métriques internes

Inertie intra-cluster :

$$I_W = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathcal{C}_k} d^2(\mathbf{x}_i, \mathbf{g}_k)$$

- Mesure la compacité des clusters.
- À minimiser.

Score de silhouette : Pour un point $\mathbf{x}_i$, le score de silhouette $s(\mathbf{x}_i)$ est défini comme :

$$s(\mathbf{x}_i) = \frac{b(\mathbf{x}_i) - a(\mathbf{x}_i)}{\max(a(\mathbf{x}_i), b(\mathbf{x}_i))}$$

où :

- $a(\mathbf{x}_i)$: Distance moyenne entre $\mathbf{x}_i$ et les autres points de son cluster.
- $b(\mathbf{x}_i)$: Distance moyenne entre $\mathbf{x}_i$ et les points du cluster le plus proche.
- Le score varie entre -1 et 1.
- Un score proche de 1 indique que $\mathbf{x}_i$ est bien clusterisé.
- Un score proche de -1 indique que $\mathbf{x}_i$ est mal clusterisé.

Métriques externes

Précision, rappel, et F1-score : Si des étiquettes de référence sont disponibles, on peut calculer :

- La précision : proportion de paires de points du même cluster qui sont aussi dans la même classe de référence.
- Le rappel : proportion de paires de points de la même classe de référence qui sont aussi dans le même cluster.
- Le F1-score : moyenne harmonique de la précision et du rappel.

Indice de Rand ajusté : Mesure la similarité entre la partition obtenue et la partition de référence, en tenant compte du hasard.

Choix du nombre de clusters

Méthode du coude :

- Tracer I_W en fonction de K.
- Choisir K au "coude" de la courbe.

Méthode de la silhouette :

- Tracer le score de silhouette moyen en fonction de K.

- Choisir K qui maximise le score.

Exemple : Pour un jeu de données donné, si la méthode du coude suggère $K = 3$ et que le score de silhouette est maximal pour $K = 3$, cela renforce la confiance dans ce choix. En pratique, il est souvent utile de combiner plusieurs méthodes pour valider le nombre optimal de clusters [**?**, **?**].