DIFIQ
Niveau 2
Statistiques
Classification Ascendante Hiérarchique (CAH)

Jean-François Berger-Lefébure

21 Novembre 2024

Contents

1 Transition vers l'algorithme CAH

Passons maintenant à l'algorithme de la **Classification Ascendante Hiérarchique (CAH)**, l'un des algorithmes les plus utilisés dans le clustering.

L'idée de la CAH est simple mais efficace : on commence par considérer chaque individu comme un **cluster individuel**. Cette étape initiale peut sembler triviale, mais elle constitue la base sur laquelle l'algorithme va itérer. En fait, à l'origine, nous avons N clusters (un pour chaque individu). Bien sûr, si certains individus étaient identiques, le nombre de clusters au départ aurait été plus petit. Mais pour simplifier, imaginons que chaque individu soit dans un cluster distinct.

Ensuite, à chaque itération, l'algorithme va regrouper les **deux clusters les plus proches**, en fonction d'une **distance** que l'on choisira, et qui peut être une distance ultramétrique, par exemple. À chaque étape, le nombre de clusters diminue, jusqu'à ce qu'il en reste finalement **un seul**.

Ce processus de regroupement successif est simple, mais il soulève des questions intéressantes, notamment sur **comment mesurer la distance** entre des clusters de tailles différentes, comme entre un cluster de deux individus et un autre avec un seul individu. Cette mesure de la distance est un aspect clé de l'algorithme, car elle va guider l'agglomération des clusters à chaque itération.
Nous allons détailler cet algorithme et voir plus précisément comment les distances entre les clusters sont calculées à chaque étape.

2 Introduction

2.1 L'idée générale

L'algorithme commence par considérer chaque individu dans son propre cluster (étape initiale avec $K = n$). Puis, les clusters sont fusionnés progressivement en fonction de leur proximité (selon une distance ∇) jusqu'à obtenir un seul cluster final (étape finale avec $K = 1$).
À retenir : L'algorithme construit une hiérarchie, et l'agrégation est toujours basée sur les deux clusters les plus proches. Ce processus est au cœur de l'algorithme.

Les distances entre clusters jouent un rôle clé

- Le choix de la distance ∇ (comme Single Linkage, Complete Linkage, Ward, etc.) a un impact direct sur les regroupements successifs.
- Il est crucial de savoir que cette distance peut varier selon la stratégie choisie, mais le but reste d'identifier les clusters les plus similaires à chaque étape.

Le dendrogramme

- L'algorithme produit une hiérarchie de partitions que l'on visualise grâce à un dendrogramme.
- C'est une représentation graphique qui montre comment les clusters sont fusionnés à chaque étape. Savoir le lire et le comprendre est important pour identifier un bon découpage.

2.2 Illustration des formules de l'algorithme CAH

Pour illustrer les formules de l'algorithme de la Classification Ascendante Hiérarchique (CAH), examinons les étapes suivantes.

1. Formule Initiale ($K = n$)

Au départ, chaque individu est considéré comme un cluster distinct. La partition initiale est donnée par :

$$\mathcal{P}_n = \{\{1\}, \{2\}, \ldots, \{n\}\}$$

Chaque individu i forme son propre cluster $C_i = \{i\}$.

2. Après la Première Itération ($K = n - 1$)

Dans cette étape, les deux clusters les plus proches sont fusionnés. La proximité entre deux clusters est mesurée par une distance $\nabla(C_1, C_2)$. Par exemple, si on utilise la stratégie "Single Linkage" (distance minimale) :

$$\nabla(C_1, C_2) = \min_{i \in C_1, j \in C_2} d(i, j)$$

Supposons que $C_1 = \{1\}$ et $C_2 = \{2\}$ soient les clusters les plus proches, ils sont fusionnés pour former un nouveau cluster $C_{1,2}$:

$$\mathcal{P}_{n-1} = \{\{1, 2\}, \{3\}, \ldots, \{n\}\}$$

3. Après la Deuxième Itération ($K = n - 2$)

On répète le processus avec la partition $\mathcal{P}_{n-1}$. Les deux clusters les plus proches dans $\mathcal{P}_{n-1}$ sont fusionnés. Supposons que $C_{1,2} = \{1, 2\}$ et $C_3 = \{3\}$ soient les plus proches, ils sont regroupés pour former un nouveau cluster $C_{1,2,3}$:

$$\mathcal{P}_{n-2} = \{\{1, 2, 3\}, \{4\}, \ldots, \{n\}\}$$

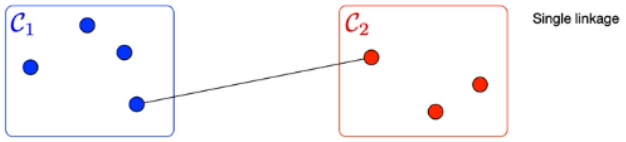
Synthèse des Partitions

- **Initiale** ($K = n$) : $\mathcal{P}_n = \{\{1\}, \{2\}, \ldots, \{n\}\}$
- **Après 1 itération** ($K = n - 1$) : $\mathcal{P}_{n-1} = \{\{1, 2\}, \{3\}, \ldots, \{n\}\}$
- **Après 2 itérations** ($K = n - 2$) : $\mathcal{P}_{n-2} = \{\{1, 2, 3\}, \{4\}, \ldots, \{n\}\}$

—

3 Stratégies d'agrégation classiques

3.1 Single Linkage (Saut minimal)

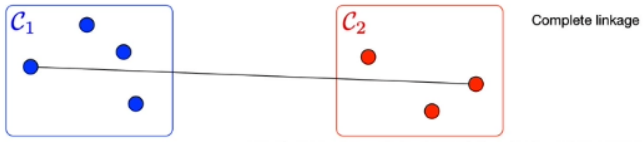


- **Définition** : La distance entre deux clusters C_1 et C_2 est la plus petite distance entre un élément de C_1 et un élément de C_2.

$$\nabla(C_1, C_2) = \min_{i \in C_1, j \in C_2} d(i, j)$$

- **Avantage** : Permet de connecter les clusters par les points les plus proches. Utile pour détecter les outliers !
- **Inconvénient** : Tendance à créer des clusters filiformes ou enchaînés.

3.2 Complete Linkage (Saut maximal)

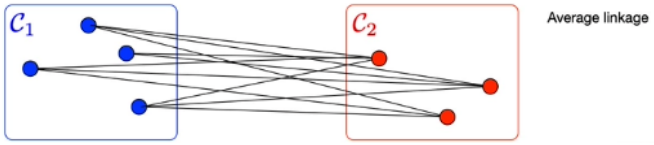


- **Définition** : La distance entre deux clusters C_1 et C_2 est la plus grande distance entre un élément de C_1 et un élément de C_2.

$$\nabla(C_1, C_2) = \max_{i \in C_1, j \in C_2} d(i, j)$$

- **Avantage** : Favorise la compacité des clusters.
- **Inconvénient** : Peut isoler certains points, créant des clusters très petits.

3.3 Average Linkage (Distance moyenne)

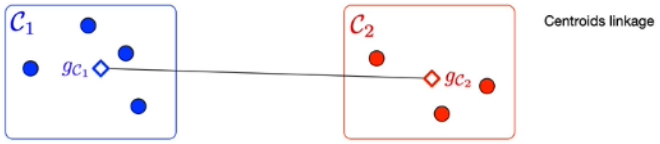


- **Définition** : La distance entre deux clusters C_1 et C_2 est la moyenne des distances entre tous les éléments de C_1 et C_2.

$$\nabla(C_1, C_2) = \frac{1}{|C_1| \cdot |C_2|} \sum_{i \in C_1, j \in C_2} d(i, j)$$

- **Avantage** : Équilibre entre Single et Complete Linkage.
- **Inconvénient** : Complexité calculatoire plus importante, car il faut calculer toutes les distances.

3.4 Centroids Linkage (Distance entre barycentres)



- **Définition** : La distance entre deux clusters C_1 et C_2 est la distance entre leurs barycentres (g_{C_1} et g_{C_2}).

$$\nabla(C_1, C_2) = d(g_{C_1}, g_{C_2})$$

- **Avantage** : Réduction du nombre de calculs, car seule la position des barycentres est utilisée.
- **Inconvénient** : Nécessite de recalculer les barycentres après chaque fusion, ce qui peut devenir coûteux si les clusters sont nombreux.

4 Stratégie d'agrégation de Ward

4.1 Définition

Ward est une stratégie d'agrégation, comme le sont Single Linkage, Complete Linkage ou Average Linkage. Ce qui distingue Ward des autres stratégies, c'est son objectif de minimiser l'augmentation de l'inertie intra-classe à chaque étape de l'agrégation.
La distance entre deux clusters C_1 et C_2 est définie comme suit :

$$\nabla(C_1, C_2) = \frac{\mu_{C_1}\mu_{C_2}}{\mu_{C_1} + \mu_{C_2}} d^2(g_{C_1}, g_{C_2})$$

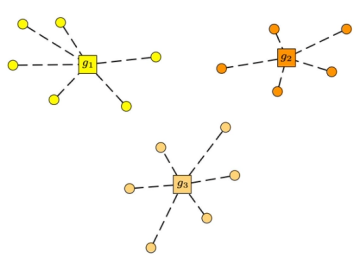
où :

- μ_{C_1} et μ_{C_2} sont respectivement les poids des clusters C_1 et C_2.
- g_{C_1} et g_{C_2} sont respectivement les centres de gravité des clusters C_1 et C_2.
- d est la distance **euclidienne**.

4.2 Principe

- La méthode de Ward minimise l'**inertie intra-classe** à chaque étape. Cela signifie que l'agrégation de deux clusters entraîne la plus faible augmentation possible de la somme des carrés des distances des individus au centre de gravité de leur cluster.
- L'algorithme garantit que chaque regroupement successif minimise cette inertie intra-classe.

Exemple : l'inertie intra-classes



4.3 Caractéristiques et contraintes

- **Distance utilisée** : La méthode de Ward nécessite l'utilisation de la **distance euclidienne**. Les autres distances (Manhattan, Chebyshev, etc.) ne sont pas compatibles avec cette stratégie.
- **Résultat mathématique** : Contrairement aux autres stratégies (Single, Complete, Average Linkage), la méthode de Ward a une base mathématique claire qui garantit la minimisation de l'inertie intra-classe.
- **Limitation** : Certains critiques de cette méthode estiment qu'elle réduit la flexibilité de la CAH, car elle contraint le choix de la distance.

4.4 Avantages et inconvénients

Avantages :

- Garantit la minimisation de l'inertie intra-classe, ce qui peut produire des clusters plus homogènes.
- Particulièrement adaptée lorsque les données sont représentées dans un espace euclidien.

Inconvénients :

- Contraint le choix de la distance (uniquement euclidienne).
- Peut être perçue comme similaire à des méthodes comme K-means, ce qui peut limiter son intérêt pour les puristes de la CAH.

Conclusion

La méthode de Ward est une stratégie puissante pour minimiser l'inertie intra-classe, mais elle est plus rigide que les autres stratégies d'agrégation. Le choix de cette méthode dépend des objectifs de l'analyse et des contraintes imposées par les données ou les préférences de l'analyste.

—

5 Dendrogramme : interprétation et lecture

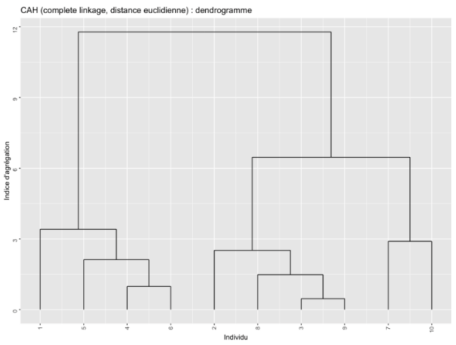
5.1 Utilité

Le dendrogramme est un outil essentiel dans la Classification Ascendante Hiérarchique (CAH) pour :

- Visualiser les **agrégations successives** des clusters lors de leur fusion.
- Comprendre **l'histoire des regroupements**, du niveau individuel (chaque individu est son propre cluster) jusqu'à un cluster unique contenant tous les individus.
- Aider à **déterminer le nombre optimal de clusters** :
  - Identifier des points où les fusions deviennent coûteuses en termes de dissimilarité.
  - Décider à quel niveau couper le dendrogramme pour obtenir une partition pertinente.
- Illustrer comment les regroupements sont influencés par la **distance** ou la stratégie d'agrégation choisie (ici, par exemple, Complete Linkage avec distance euclidienne).

5.2 Lecture et synthèse du dendrogramme

Voici les étapes principales pour lire et interpréter un dendrogramme :



1. Chaque **individu est un cluster unique au départ**, représenté par un point en bas du graphique.
2. Les **fusions successives** sont indiquées par des barres horizontales :
   - Une barre basse signifie que les clusters fusionnés étaient très proches.
   - Une barre haute indique que la fusion regroupe des clusters dissemblables.
3. **L'axe des X** : Représente les individus ou clusters.
4. **L'axe des Y** : Représente la distance ou la dissimilarité entre les clusters fusionnés (selon la stratégie d'agrégation utilisée).

5.3 Synthèse de l'interprétation

- Les fusions basses dans le dendrogramme montrent des regroupements entre individus ou clusters très similaires.
- Les fusions hautes reflètent des regroupements coûteux, entre clusters peu similaires.
- Une **coupure horizontale** permet de déterminer le nombre optimal de clusters :
  - Identifiez des sauts importants dans les hauteurs des barres pour repérer des regroupements significatifs.
  - Par exemple, une coupure à une hauteur spécifique peut indiquer K clusters pertinents.
- Dans le cas d'un grand jeu de données, la lisibilité du dendrogramme peut être difficile. Dans ces situations, analyser les **hauteurs d'agrégation** ou utiliser des méthodes quantitatives est recommandé.

Conclusion

Le dendrogramme est un outil puissant pour analyser les regroupements hiérarchiques et décider du nombre optimal de clusters. Il aide à interpréter les relations entre les données et à visualiser les impacts des stratégies d'agrégation choisies.

6 Différence entre Ward et K-means (au niveau de l'utilité)

6.1 Objectif principal

- **Ward (CAH - Classification Ascendante Hiérarchique)** :
  - **Utilité** : Construire une hiérarchie de partitions emboîtées pour identifier des regroupements naturels à différents niveaux.
  - **Objectif** : Minimiser l'inertie intra-classe à chaque étape, tout en construisant une hiérarchie visualisée sous forme de **dendrogramme**.
  - **Adapté pour** :
    - Explorer les données pour trouver une structure hiérarchique.
    - Identifier les meilleurs regroupements visuellement.
    - Petits jeux de données (complexité élevée).
- **K-means** :
  - **Utilité** : Identifier une partition optimale des données en K clusters prédéfinis.
  - **Objectif** : Minimiser directement l'inertie intra-classe pour un nombre fixé de clusters K.
  - **Adapté pour** :
    - Problèmes nécessitant une partition fixe avec un K connu à l'avance.
    - Jeux de données volumineux où une hiérarchie n'est pas nécessaire.

6.2 Approche méthodologique

Critère	**Ward (CAH)**	**K-means**
Structure des clusters	Hiérarchie emboîtée (dendrogramme)	Partition unique en K clusters
Choix du nombre K	Choix *a posteriori*	Fixé à l'avance
Représentation graphique	Dendrogramme	Aucune hiérarchie
Flexibilité	Exploration multi-niveaux	Partition unique

Table 1: Comparaison des types de résultats entre Ward et K-means.

6.3 Complexité algorithmique

- **Ward** : Complexité élevée ($O(n^2 \log(n))$), car il calcule les distances entre tous les clusters à chaque itération. Adapté aux petits jeux de données.
- **K-means** : Complexité plus faible ($O(n \cdot K \cdot t)$), où t est le nombre d'itérations. Convient aux grands jeux de données.

6.4 Cas d'utilisation

- **Ward** :
  - Explorer plusieurs niveaux de regroupement.
  - Étudier les clusters avec des visualisations comme les dendrogrammes.
- **K-means** :
  - Identifier rapidement une partition optimale pour un K fixé.
  - Travailler avec de grands ensembles de données.

Résumé

- **Ward** : Idéal pour explorer des données et identifier une hiérarchie naturelle, particulièrement quand le nombre de clusters optimal K n'est pas connu.
- **K-means** : Plus rapide et efficace pour des données volumineuses lorsque K est fixé à l'avance.