

Analyse de regroupements

Analyse multidimensionnelle appliquée

Léo Belzile

HEC Montréal

automne 2022

L'analyse de regroupements cherche à créer une division de n observations de p variables en regroupements.

1. méthodes basées sur la connectivité (regroupements hiérarchiques, AGNES et DIANA)
2. méthodes basées sur les centroïdes et les médoïdes (k -moyennes, k -médoïdes PAM, CLARA)
3. mélanges de modèles (mélanges Gaussiens, etc.)
4. méthodes basées sur la densité (DBScan)
5. méthodes spectrales

1. Initialisation: sélectionner K des n observations comme médoïdes initiaux.
2. Assigner chaque observation au médoïde le plus près.
3. Calculer la dissimilarité totale entre chaque médoïde et les observations de son groupe.
4. Pour chaque médoïde ($k = 1, \dots, K$):
 - considérer tous les $n - K$ observations à tour de rôle et permuter le médoïde avec l'observation.
 - calculer la distance totale et sélectionner l'observation qui diminue le plus la distance totale.
5. Répéter les étapes 2 à 4 jusqu'à ce que les médoïdes ne changent plus.

L'algorithme CLARA, décrit dans Kaufman & Rousseeuw (1990), réduit le coût de calcul et de stockage.

- Diviser l'échantillon en S sous-échantillons de taille approximativement égale de taille n_S (typiquement $K \ll n_S < 1000$)
- Utiliser l'algorithme PAM sur chaque sous-échantillon.

Une fois les médoïdes obtenus, le reste de toutes les observations de l'échantillon sont assignées au regroupement du médoïde le plus près.

La qualité de la segmentation pour chacune des S segmentations est calculée en obtenant la distance moyenne entre les médoïdes et les observations.

On retourne la meilleure segmentation parmi les S (celle qui a la plus petite distance moyenne).

Disponible depuis le paquet cluster.

```
1 set.seed(60602)
2 kmedoide5 <- cluster::clara(
3   x = donsmult_std,
4   k = 5L, # nombre de groupes
5   sampsize = 500, #taille échantillon pour PAM
6   metric = "euclidean", # distance l2
7   #cluster.only = TRUE, # ne conserver que étiquettes
8   rngR = TRUE, # germe aléatoire depuis R
9   pamLike = TRUE, # même algorithme que PAM
10  samples = 10) #nombre de répétitions aléatoires
```

Valeurs initiales et paramètres

Même hyperparamètres que K -moyennes (dissemblance, nombre de regroupements, initialisation et séparation).

Comme les K -moyennes, on fera plusieurs essais pour trouver de bonnes valeurs de départ. On peut tracer le profil des silhouettes.

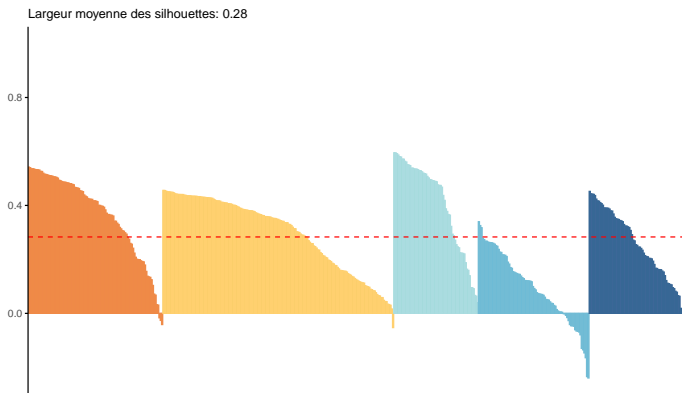


Figure 1: Silhouettes pour les données de dons multiples avec l'algorithme CLARA pour $K = 5$ regroupements.

Puisque les prototypes (médoides) sont des observations, on peut simplement extraire leur identifiant.

```
1 medoides_orig <- donsmult[kmedoide5$i.med,]  
2 # Taille des regroupements  
3 kmedoide5$clusinfo
```


- (—) solution approximative pour grand échantillons
- (+) les prototypes sont des observations de l'échantillon.
- (+) la fonction objective est moins impactée par les extrêmes.
- (—) le coût de calcul est prohibitif avec des mégadonnées (problème combinatoire) avec complexité $O(n^2)$. PAM fonctionne avec maximum 1000 observations.

On suppose qu'on a K groupes, chacun caractérisé par une densité de dimension p , soit $f_k(X_i; \theta_k)$ si X_i provient du groupe $k = 1, \dots, K$.

Généralement, on choisit une loi normale multidimensionnelle pour le k e groupe G ,

$$X \mid G = k \sim \text{No}_p(\mu_k, \Sigma_k)$$

La probabilité qu'une observation soit tirée du groupe $G = k$ est π_k .

La vraisemblance est une fonction des paramètres μ_k, Σ_k et de la probabilité π_k qu'une observation \mathbf{x}_i tombe dans le groupe k ,

$$L_i(\{\mu_k, \Sigma_k, \pi_k\}_{k=1}^K; \mathbf{x}_i) = \sum_{k=1}^K \pi_k f_k(X_i \mid \mu_k, \Sigma_k).$$

Le maximum de vraisemblance est obtenu à l'aide de l'algorithme d'espérance-maximisation en augmentant les observations avec un indicateur de groupe.

- Étape *E*: assignation aux groupes (multinomiale).
- Étape *M*: estimation des probabilités, des moyennes et variances.

Le mélange de modèle nous donne accès à la probabilité π_k qu'une observation appartienne au groupe G_k (**assignation probabiliste**).

Chacune des K matrice de covariance contient $p(p + 1)/2$ paramètres!

En paramétrisant ces dernière, on peut réduire le nombre de paramètres à estimer.

- compromis entre simplicité (d'estimation) et nombre de paramètres

La matrice de covariance dans `mclust` est paramétrisée en fonction de

- λ , qui contrôle le volume,
- une matrice diagonale **A** qui contrôle les variances de chaque observation et
- **D** une matrice orthogonale qui permet de créer de la corrélation entre observations.

Un index k spécifie que cette composante varie d'un regroupement à l'autre.

Paramétrisation des variables

Voir `mclust.options("emModelNames")` et la documentation dans le Tableau 3 de l'article sur `mclust`.

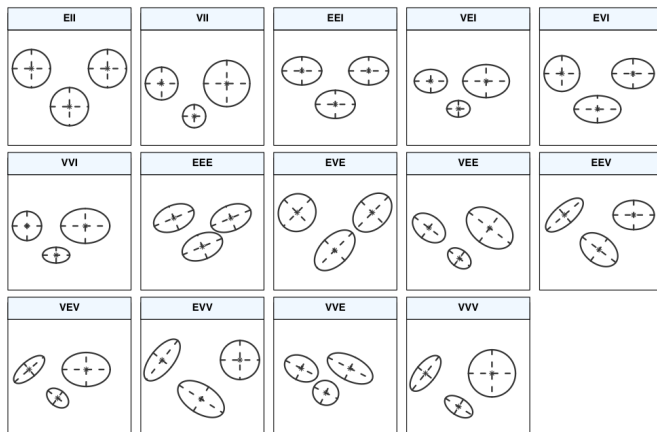


Figure 2: Forme des ellipsoïdes pour le mélange de modèle selon la forme de la structure de covariance. Tirée de `mclust5`, licence CC BY 4.0.

```
1  ## Mélanges de modèles gaussiens
2  set.seed(60602)
3  library(mclust)
4  mmg <- Mclust(data = donsmult_std,
5               G = 1:10,
6               # Ajouter composante uniforme
7               # pour bruit (aberrances)
8               initialization = list(noise = TRUE))
9  # Résumé de la segmentation
10 summary(mmg)
```

On peut obtenir les étiquettes (avec 0 pour le bruit) avec `mmg$classification`.

- le nombre de regroupements K
- la forme des ellipsoïdes (structure de covariance)
- les valeurs pour l'initialisation.

Les mêmes considérations pratiques qu'avec les K -moyennes s'appliquent.

En pratique, on ajuste le modèle avec différent nombre de regroupements et différentes structures de covariance et on prend le modèle avec le plus petit BIC.

Sélection des hyperparamètres

```
1 plot(mmg, what = "BIC")
```

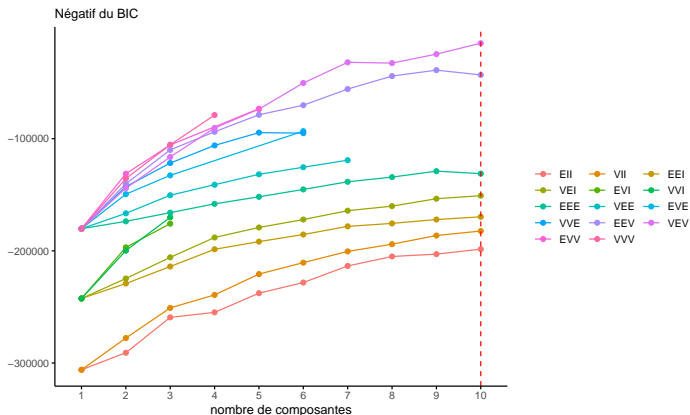


Figure 3: Valeur du négatif du BIC pour les mélanges de modèles gaussiens selon le nombre de regroupements et la structure de covariance.

Le logiciel n'arrive pas à estimer certains modèles complexes quand le nombre de

Représentation graphique des regroupements

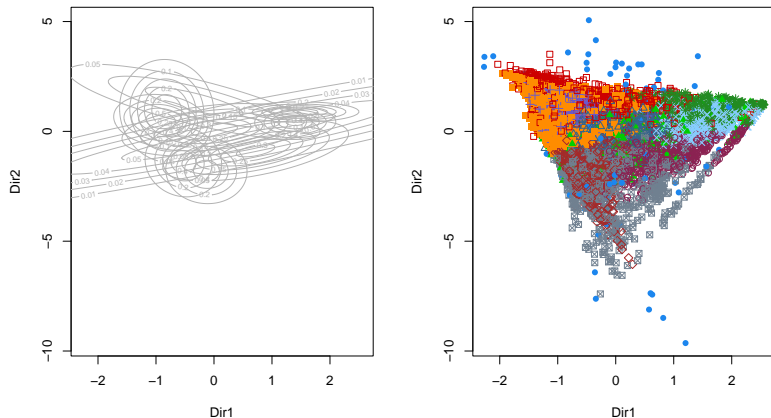


Figure 4: Projection des observations, colorées par regroupement (gauche) et structure des regroupements avec ellipsoïdes de confiance (droite).

- (+) approche est plus flexible que les K -moyennes.
- (+) l'ajout d'une composante uniforme permet de gérer les aberrances (supporté par `mclust`).
- (+) l'algorithme EM garantit la convergence à un minimum local (comme pour les K -moyennes)
- (+) on obtient une assignation probabiliste plutôt que rigide
- (−) le coût de calcul est plus élevé que les K -moyennes
- (−) le nombre de paramètre des matrices de covariance augmente rapidement avec la dimension p .

L'algorithme DBSCAN (*density-based spatial clustering of applications with noise*) est une méthode de partitionnement basée sur la densité des points.

L'idée de base de l'algorithme est de tracer une boule de rayon ϵ autour de chaque observation et de voir si elle inclut d'autres observations.

L'algorithme classe les observations en trois catégories:

- Un point central est une observation qui possède $M - 1$ voisins à distance ϵ .
- Un point frontière est un point qui est distant de moins de ϵ d'un point central, sans en être un.
- Un point isolé est une observation qui n'est pas rattachée à aucun regroupement.

L'algorithme répète les étapes suivantes jusqu'à ce que chaque observation ait été visitée.

1. Choisir un point aléatoirement parmi ceux qui n'ont pas été visités.
2. Si le point n'est pas étiqueté, calculer le nombre de points voisins qui se trouvent dans un rayon ϵ : s'il y a moins de M observations, provisoirement étiqueter l'observation comme point isolé, sinon comme point central.
3. Si l'observation est un point central avec $M - 1$ voisins ou plus, créer un regroupement.
4. Étiqueter chaque point à distance ϵ créé et l'ajouter au regroupement, ainsi que tout point à distance ϵ de ces voisins.

Ce site web offre une visualisation interactive des différentes étapes de l'algorithme et de comparer la performance de DBSCAN selon le type de regroupements.

- le rayon ϵ et
- le nombre minimal de points pour former un regroupement, M

Hyperparamètres M et ϵ corrélés: si on augmente le nombre minimal de point M par regroupement, il faudra également augmenter le rayon ϵ pour éviter d'avoir un nombre trop élevé de points isolés et d'aberrances.

Classification avec DBSCAN

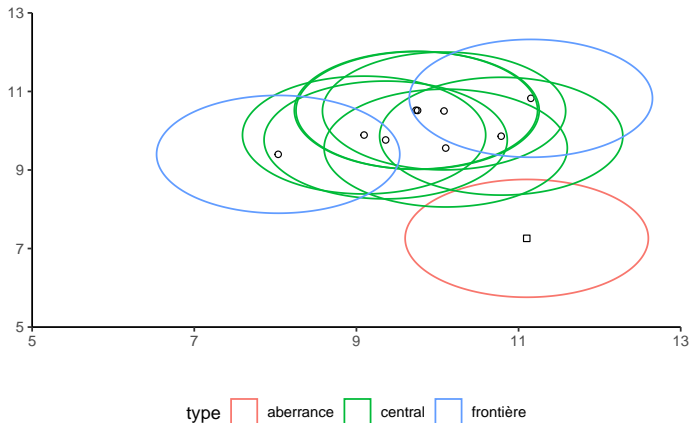


Figure 5: Illustration de la classification des points avec DBSCAN: toutes les observations sont assignées à un regroupement, moins une aberrance.

Avec p variables explicatives, on recommande $M > p + 1$.

Pour ε , considérer les M plus proches voisins.

La fonction `kNNdistplot` du paquet `dbscan` permet de tracer un graphique de la distance moyenne des k plus proches voisins pour chaque observation:

- en prenant $k = M - 1$, calculer la distance entre le k plus proche voisin de chaque observation.
- ordonner ces distances.
- choisir ϵ selon coude

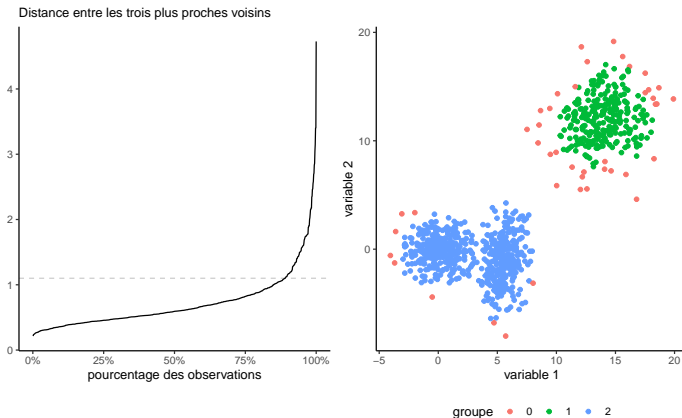


Figure 6: Graphique des distances entre chaque observation et son troisième plus proche voisin (gauche), en fonction du pourcentage d'observations à moins de cette distance et regroupements obtenus avec DBSCAN avec $M = 10$ et $\epsilon = 1.1$ (droite).

- (—) le traitement des aberrances est automatique et l'algorithme est robuste.
- (+) le nombre de regroupements n'a pas à être spécifié apriori.
- (+) la forme des regroupements est arbitraire, peut être non convexe et de taille différente.
- (+/-) la complexité de l'algorithme est d'au mieux $O(n \ln n)$ (calcul) et $O(n)$ pour le stockage puisque chaque point est visité à tour de rôle et comparé aux autres pour trouver les plus proches voisins.

- (—) les hyperparamètres ont une interprétation physique
- (—) leur choix n'est pas aisé
- (—) DBSCAN ne permet pas de traiter le cas où la densité des regroupements change et risque de fusionner des regroupements s'il y a une série d'observations qui permet de relier deux regroupements.
- (—) comme la plupart des algorithmes, le voisinage des points devient épars quand p augmente en raison du fléau de la dimension.

Méthode déterministe de regroupement à partir d'une matrice de dissimilarité.

1. Initialisation: chaque observation est assignée à son propre groupe.
2. les deux groupes les plus rapprochés sont fusionnés; la distance entre le nouveau groupe et les autres regroupements est recalculée.
3. on répète l'étape 2 jusqu'à obtenir un seul regroupement.

Il y a plusieurs façons de calculer la distance entre deux groupes d'observations de plusieurs observations, notamment

- liaison simple (`method = single`): plus proches voisins
- liaison complète (`method = complete`): voisins les plus éloignés
- liaison moyenne (`method = average`): utilise la moyenne des distances entre toutes les paires de sujets (un pour chaque groupe) provenant des deux groupes.
- méthode de Ward (`method = ward.D2`): calcul de l'homogénéité globale

Illustration des mesures de liaison

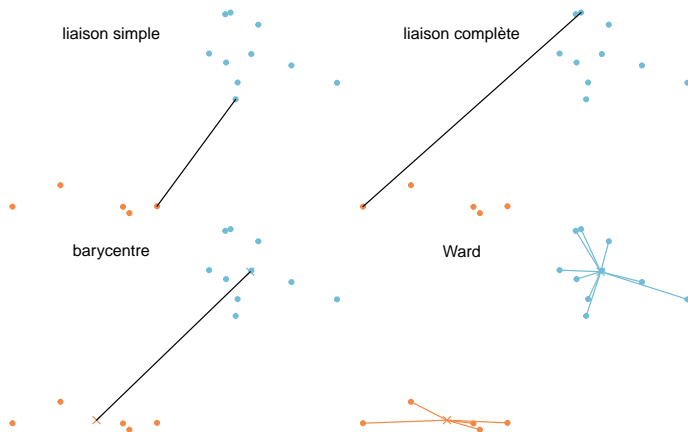


Figure 7: Distances entre regroupements selon la liaison (simple, complète, barycentre, homogenéité de Ward).

La méthode de Ward utilise l'homogénéité comme critère.

Pour chaque groupe, on calcule la somme des carrés des distances par rapport à la moyenne du groupe, disons SCD_k ($k = 1, \dots, M$).

On calcule ensuite l'homogénéité globale en faisant la somme, $H^{(M)} = SCD_1 + \dots + SCD_M$.

La méthode de Ward va fusionner les deux groupes qui feront augmenter le moins possible l'homogénéité.

Les algorithmes de regroupement hiérarchiques stockent une matrice de dissemblance $n \times n$: coût de stockage quadratique $O(n^2)$.

Généralement, le coût de calcul est au mieux $\Omega(n^2)$ et au pire $O(n^3)$.

Pour la méthode de liaison simple, un algorithme permet d'obtenir un coût de calcul quadratique de $O(n^2)$ sans stocker la matrice de dissemblance, d'où un coût de stockage linéaire de $O(n)$.

`stat::hclust` permet de faire des regroupements agglomératifs, mais le paquet `fastcluster` propose une version avec une empreinte mémoire inférieure (*plus rapide!)

Alternative de Gagolewski (2016) qui modifie la fonction de liaison simple en retenant son efficacité de calcul.

Plutôt que de simplement trouver la paire de regroupements à distance minimale, cette fusion n'est appliquée que si une mesure d'inéquité est inférieur à un seuil spécifié par l'utilisateur.

Si les regroupements sont fortement inéquitables, la fusion survient entre les regroupements dont un de la taille minimale courante.

L'implémentation **R** dans le paquet `genieclust` est nettement plus rapide que les autres alternative.

- **Liaison simple:** fonctionne bien si l'écart entre deux regroupements est suffisamment grand. S'il y a du bruit entre deux regroupements, la qualité des regroupements en sera affectée. Souvent quelques valeurs isolées et un seul grand regroupement
- **Liaison complète:** moins sensible au bruit et aux faibles écarts entre regroupements, mais a tendance à casser les regroupements globulaires.
- **Homogénéité de Ward:** le critère ressemble à celui des K -moyennes.

Voir la page scikit-learn pour une illustration.

1. choix de la fonction de liaison (et hyperparamètres associés)
2. mesure de dissemblance
3. nombre de regroupements

On peut représenter le modèle à l'aide d'un arbre, où les feuilles indiquent les regroupements à chaque étape jusqu'à la racine à la dernière étape (**dendrogramme**).

La distance entre chaque embranchement est déterminée par notre critère: cela nous permet de sélectionner un nombre de regroupements K après inspection visuelle du dendrogramme.

On élague l'arbre à la hauteur voulue.

Dendrogramme

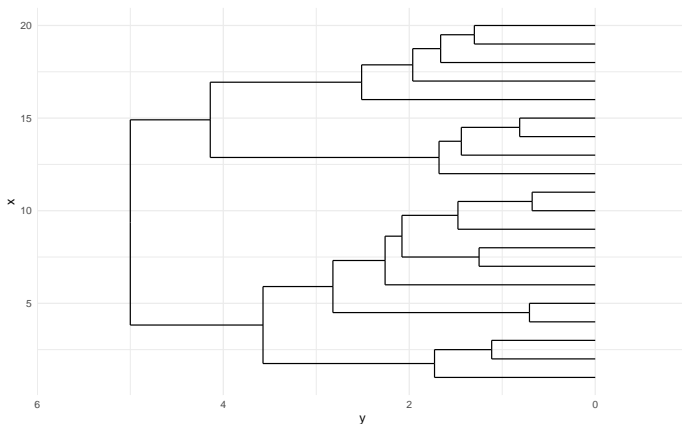


Figure 8: Dendrogramme pour l'exemple de regroupement hiérarchique avec la méthode de Ward et 20 observations.

On peut choisir K à partir du pourcentage de variance expliquée, R^2 en calculant

$$R^2_{(M)} = 1 - H_{(M)}/H_{(1)},$$

où $H_{(1)}$ est l'homogénéité globale avec un seul groupe.

Le R-carré semi-partiel mesure la perte d'homogénéité d'une étape à l'autre, renormalisée par

$$R^2_{\text{sp}(M)} = \frac{H_{(M)} - H_{(M-1)}}{H_{(1)}},$$

mesure la perte d'homogénéité (relative) en combinant ces deux groupes.

On cherche un point d'inflexion (un coude).

- (+) la solution du regroupement hiérarchique est toujours la même (déterministe)
- (—) l'assignation d'une observation à un regroupement est finale
- (—) les aberrances ne sont pas traitées et sont souvent assignées dans des regroupements à part
- (+) les méthodes d'arborescence sont faciles à expliquer
- (—) le nombre de groupes n'a pas à être spécifié apriori (une seule estimation)
- (—) le coût de calcul est prohibitif, avec une complexité quadratique de $O(n^2)$ pour la méthode de liaison simple et autrement $O(n^3)$ pour la plupart des autres fonctions de liaison.