

Analyse de regroupements

Analyse multidimensionnelle appliquée

Léo Belzile

HEC Montréal

On continue notre couverture des algorithmes de segmentation.

Les diapositives expliquent

1. les étapes des regroupements
2. les avantages et inconvénients des méthodes
3. les conditions pour lesquelles la segmentation risque d'être applicable.

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

1. méthodes basées sur les centroïdes et les médoïdes (k -moyennes, k -médoïdes)
2. mélanges de modèles
3. méthodes basées sur la connectivité (regroupements hiérarchiques agglomératifs et divisifs)
4. méthodes basées sur la densité (pas couvertes)

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.

Pour chacune des S segmentations, on calcule la distance moyenne entre les médoïdes et les observations.

La meilleure segmentation est retournée: c'est celle qui a la plus petite distance moyenne parmi les S .

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.

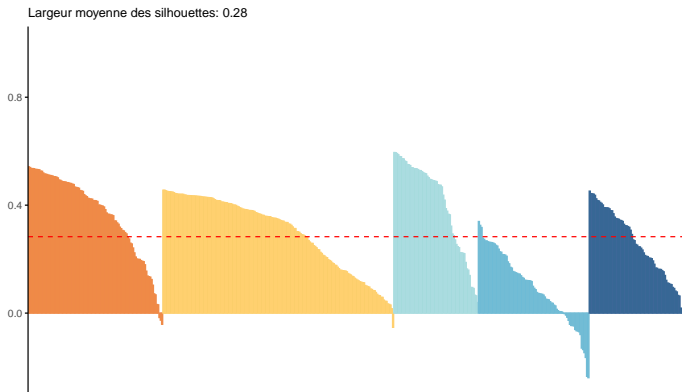


Figure 1: Silhouettes pour les données de dons multiples avec l'algorithme CLARA pour $K = 5$ regroupements.

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
```


Avantages et inconvénients des K -médoides

- (+) 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). PAM fonctionne avec maximum 1000 observations.
- (—) solution approximative pour grand échantillons avec CLARA

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 ,

$$X \mid G = k \sim \text{Normal}_p(\mu_k, \Sigma_k)$$

La probabilité qu'une observation soit tirée du groupe k est π_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 matrices de variance 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 trois composantes:

- 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 sous-indice k spécifie que cette composante varie d'un regroupement à l'autre.

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

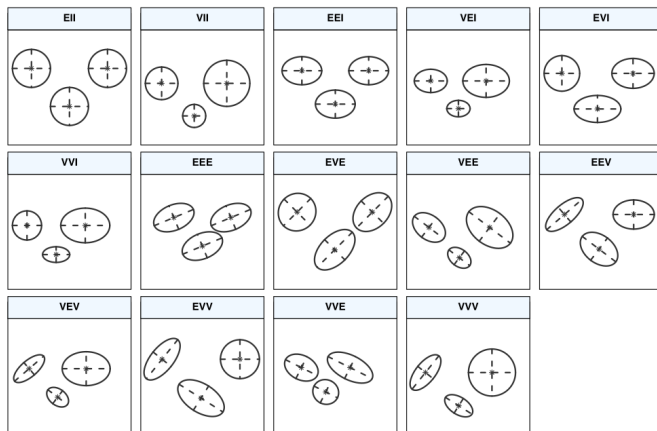


Figure 2: Forme des ellipsoïdes pour le mélange de modèle selon la covariance. Tirée de `mclust5`, licence CC BY 4.0.

```
1  ## Mélanges de modèles gaussiens
2  set.seed(60602)
3  library(mclust)
4  mmg <- Mclust(data = donsmult_std,
5               G = 1:10,
6               # Matrice de covariance (par défaut, tous choix)
7               # modelNames = mclust.options("emModelNames"),
8               ## Ajouter composante uniforme
9               ## pour bruit (aberrances)
10              initialization = list(noise = TRUE))
11  # Résumé de la segmentation
12  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.

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


Sélection des hyperparamètres

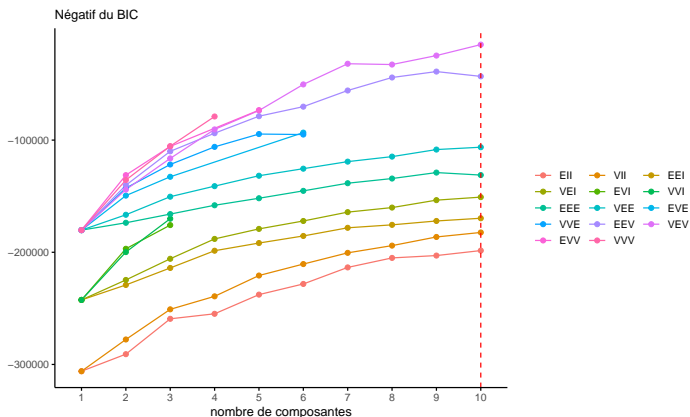


Figure 3: Valeur du négatif du BIC pour les mélanges de modèles gaussiens selon le nombre de regroupements et la structure de covariance.

Le logiciel n'arrive pas à estimer certains modèles complexes quand le nombre de groupes est trop élevé.

- (+) approche plus flexible que les K -moyennes.
- (+) l'ajout d'une composante uniforme permet de gérer les aberrances (supporté par `mclust`).
- (+) l'algorithme EM garantie la convergence à un optimum 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 paramètres des matrices de covariance augmente rapidement avec le nombre de variables explicatives p .

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

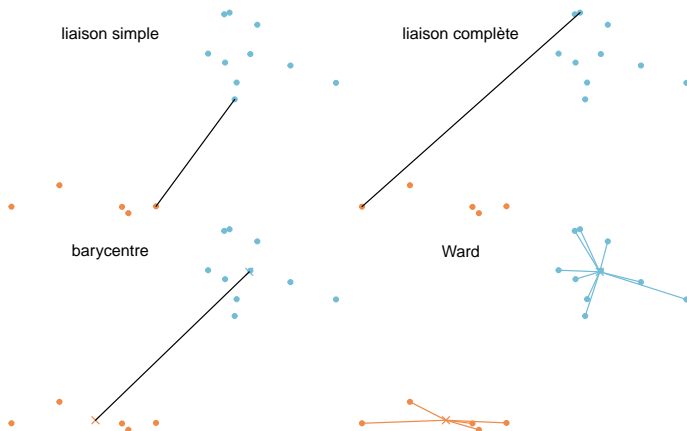


Figure 5: Distances entre regroupements selon la liaison (simple, complète, barycentre, homogenéité de Ward).

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 $SCD_{k,M}$ ($k = 1, \dots, M$).

On calcule ensuite la somme des distances,

$$SCD_{(M)} = SCD_{1,M} + \dots + SCD_{M,M}.$$

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

- 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.

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

Généralement, le coût de calcul est au mieux quadratique et au pire cubique en n .

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!)

La fonction de liaison simple permet des calculs rapides, mais le résultat est rarement satisfaisant.

Une proposition de Gagolewski (2016), implémentée dans le paquet R `genieclust`, 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, le coefficient de Gini, 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.

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

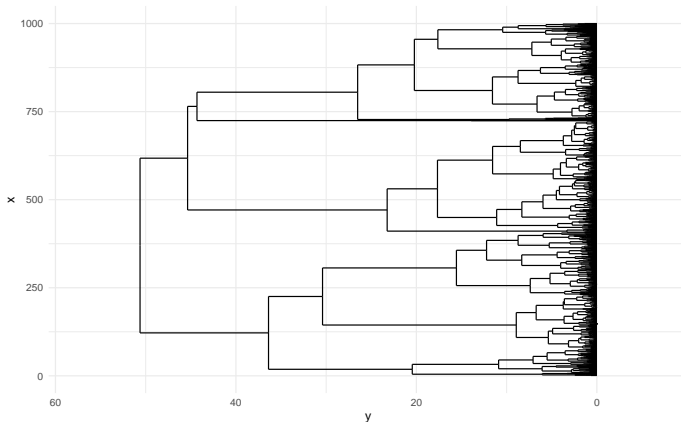


Figure 6: Dendrogramme pour l'exemple de regroupement hiérarchique avec la méthode de Ward et 100 premières observations.

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

$$R^2_{(M)} = 1 - \text{SCD}_{(M)} / \text{SCD}_{(1)},$$

où $\text{SCD}_{(1)}$ est l'homogénéité globale avec un seul groupe.

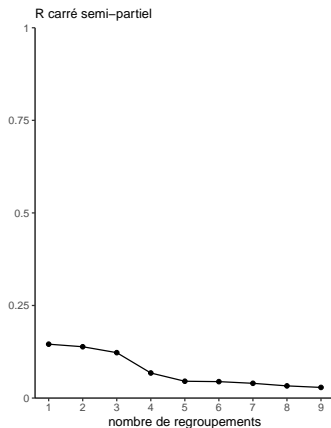
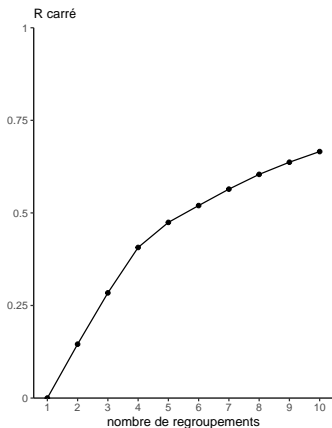
Le R-carré semi-partiel mesure la perte d'homogénéité d'une étape à l'autre, renormalisée par

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

Critères d'homogénéité (Ward)

```
1 ward <- fastcluster::hclust(  
2   d = dist(donsmult_std),  
3   method = "ward.D2")  
4 hecmulti::homogeneite(rhier = ward,  
5                        ngroupes = 10,  
6                        data = donsmult_std)
```

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



Avantages et inconvénient, regroupements hiérarchiques

- (+) la solution du regroupement hiérarchique est toujours la même (déterministe)
- (+) les méthodes d'arborescence sont faciles à expliquer
- (—) 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
- (+) le nombre de groupes n'a pas à être spécifié apriori (une seule estimation)
- (—) le coût de calcul est très élevé, 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.

On veut parfois comparer les regroupements de différentes méthodes.

Les étiquettes ne sont pas nécessairement identiques même si les regroupements le sont (permutation des étiquettes).

Une mesure de similarité, l'indice de Rand, permet de comparer deux vecteurs d'observations catégorielles (étiquettes des regroupements).

- même longueur (même nombre d'observations)
- mais pas nécessairement le même nombre de modalités (nombre de regroupements potentiellement différents d'une partition à l'autre).

Idée: comparer à tour de rôle chacune des $n(n - 1)/2$ paires d'observation.

On indique si les paires d'observations sont dans le même groupe (1) ou un groupe différent (0).

Exemple: deux partitions de 4 observations

- Étiquettes de la méthode A avec 2 regroupements: (2, 2, 2, 1)
- Étiquettes de la méthode B avec 3 regroupements: (3, 1, 3, 2)

paire	{1,2}	{1,3}	{1,4}	{2,3}	{2,4}	{3,4}
A, B	1, 0	1, 1	0, 0	1, 0	0, 0	0, 0

L'indice de Rand est le taux de bonne classification du tableau de contingence résultant. Parmi les six paires, quatre sont concordantes (0, 0 ou 1, 1), d'où un indice de Rand = 0.66

Une valeur de 1 indique que les deux partitions sont identiques.

Les étapes d'une analyse de regroupements

1. Choisir les variables pertinentes à l'analyse. Cette étape peut nécessiter de créer, transformer de nouvelles variables ou d'aggréger les données.
2. Décider quel méthode sera utilisée pour la segmentation.
3. Choisir les hyperparamètres de l'algorithme (nombre de regroupements, rayon, etc.) et la mesure de dissemblance.
4. Effectuer la segmentation et valider la qualité de cette dernière (interprétabilité, taille des groupes, homogénéité des regroupements).
5. Avec les étiquettes, calculer un prototype de groupe.
6. Interpréter les regroupements obtenus à partir des prototypes.

- L'analyste a une grande marge de manoeuvre.
- Il n'y a pas de vérité: la segmentation n'est utile que si elle a une valeur ajoutée.
- Aucun algorithme ne performe uniformément mieux, mais certains sont plus faciles à employer que d'autres.
 - avec des mégadonnées, la complexité (coût de calcul et de stockage) est un facteur important pour choisir la méthode.
 - la plupart du temps, le choix des hyperparamètres nécessite un peu d'essai-erreur.
 - la segmentation peut être médiocre parce que les hyperparamètres sont mal choisis.

Le Diable est dans les détails:

- variables catégorielles et binaires vs standardisation
- valeurs manquantes
- aberrances

Le nombre de groupes peut être guidé par le contexte: les formules et indicateurs de qualité servent de balises.