Analyse de regroupements

Analyse multidimensionnelle appliquée

Léo Belzile

HEC Montréal

automne 2022

Algorithmes pour l'analyse de regroupements

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-médoides PAM, CLARA)
3. mélanges de modèles (mélanges Gaussiens, etc.)
4. méthodes basées sur la densité (DBScan)
5. méthodes spectrales

K-médianes

Illustration de la répartition/interprétation avec $K = 5$ groupes

```
set.seed(60602)
kmed5 <- flexclust::kcca(
  x = donsmult_std,
  k = 5,
  family = flexclust::kccaFamily("kmedians"),
  control = list(initcent = "kmeanspp"))
```

Différences de segmentation

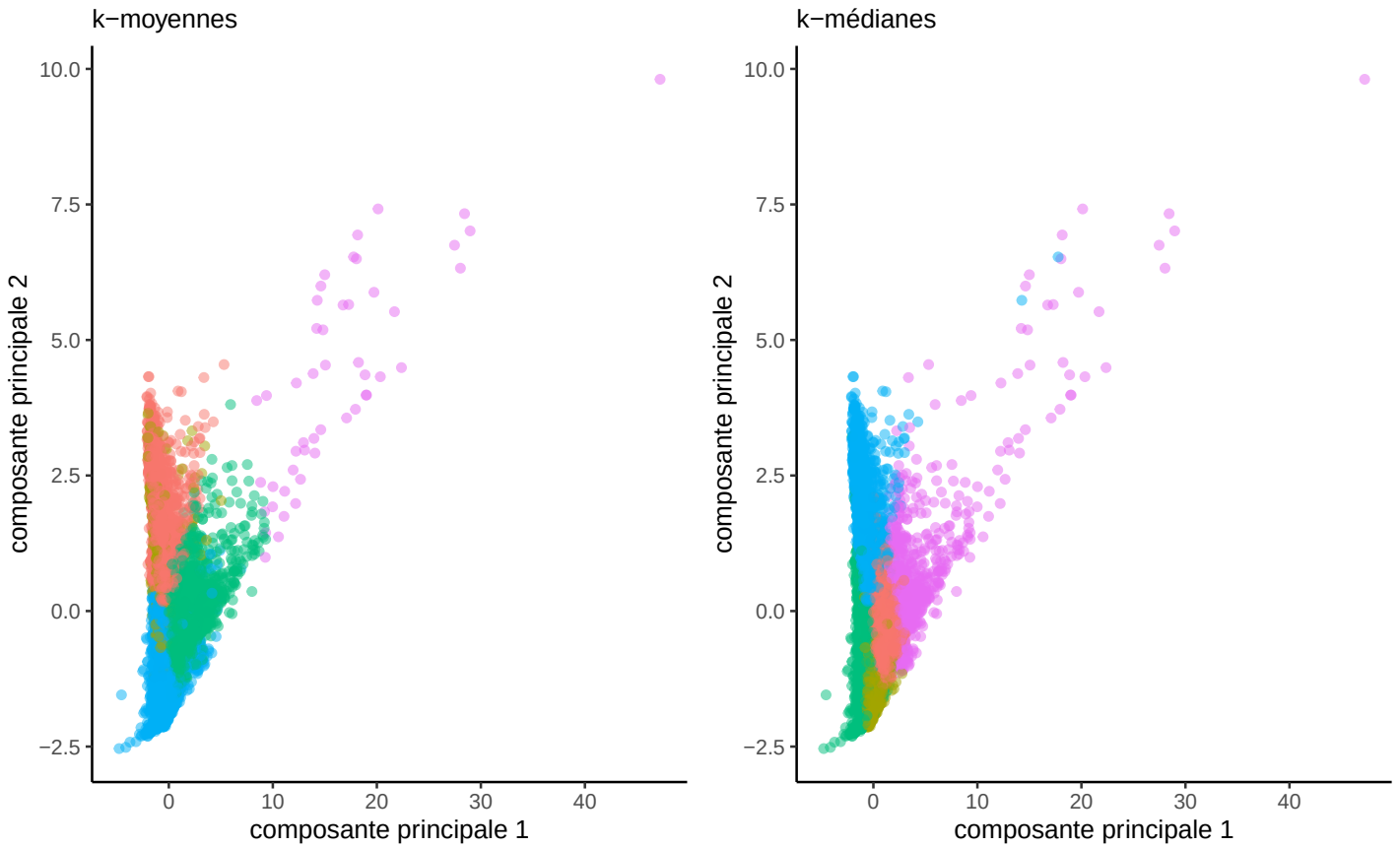


Figure 1: Nuage de points des deux premières composantes principales des observations de dons multiples avec les étiquettes des regroupements obtenus selon la méthodes des K-moyennes et K-médianes avec $K = 5$ regroupements.

Commentaire sur K-médianes

Avec les K-médianes, les personnes qui ont fait des dons plus élevés sont fusionnés avec d'autres personnes qui ont fait des dons moins élevés et les groupes sont davantage de taille comparable.

Selon l'objectif des regroupements, cela peut être avantageux, mais cibler les donateurs les plus généreux semble plus logique dans le contexte.

K-médoïdes.

Dans les K-médoïdes, on choisit une observation comme prototype.

Puisque qu'on considère chaque observation comme candidat à devenir un médoïde à chaque étape, le coût de calcul est prohibitif en grande dimension.

Algorithme de partition autour des médoïdes (PAM)

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.

Algorithme CLARA (1/2)

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

- divisant l'échantillon en S sous-échantillons de taille approximativement égale (par défaut $S = 5$)
- et en utilisant l'algorithme PAM sur chacun.

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

Algorithme CLARA (2/2)

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

On retourne la meilleure (celle qui a la plus petite distance moyenne).

PAM et CLARA dans **R**

Disponible depuis le paquet `cluster`.

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

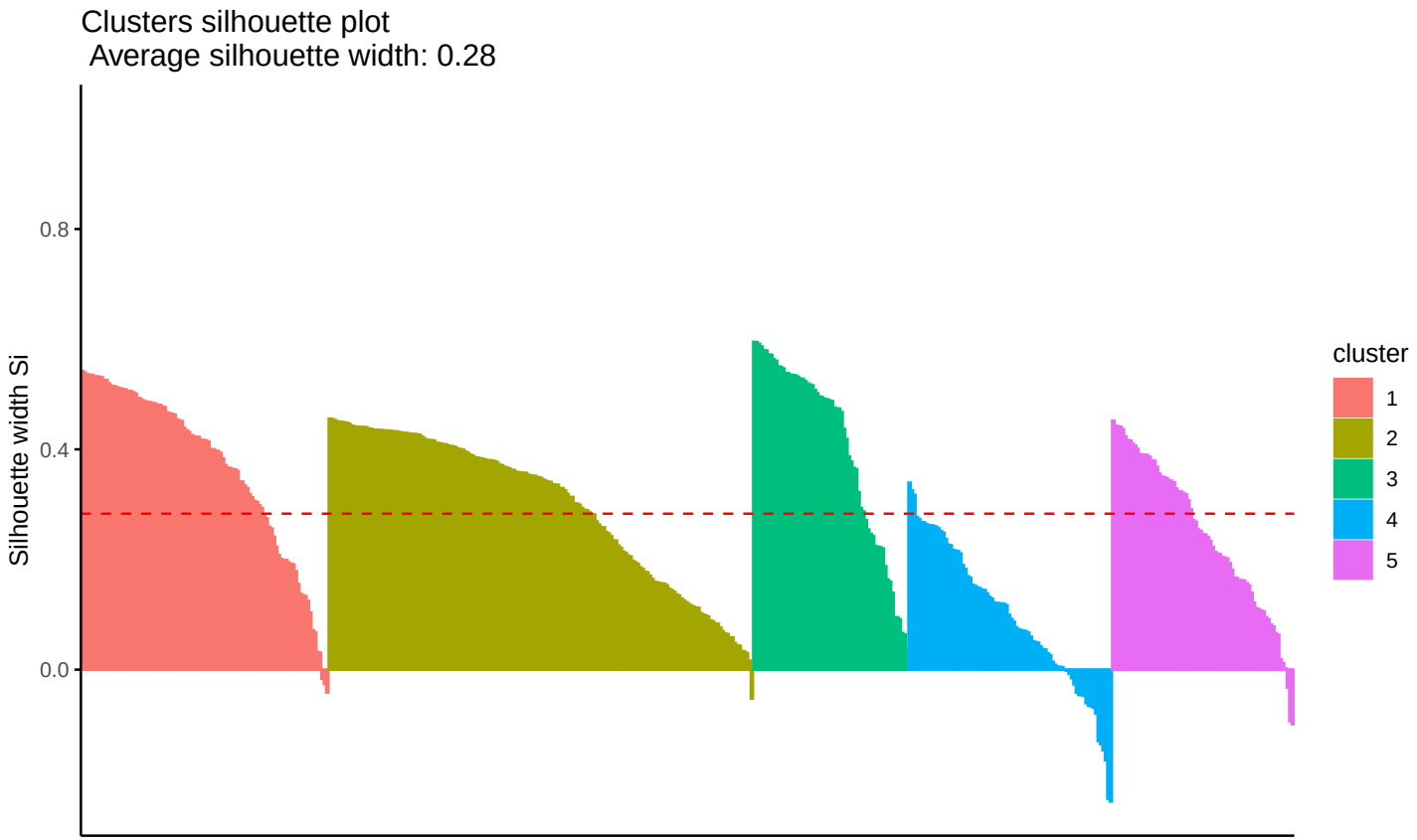
Avantages et inconvénients des K-médoïdes

- (+) 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.

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 2)



Prototypes

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

```
medoides_orig <- donsmult[kmedoide[[4]]$i.med,]
medoides_orig
# Taille des regroupements
kmedoide[[4]]$clusinfo
```

Mesures de similarité

Certains algorithmes utilisent directement une matrice de similarité $\mathbf{S}$ qui encode plutôt l'information à propos des points avoisinants.

Plus les observations sont similaires, plus elles sont proches.

Les mesures dissemblance peuvent être convertie en mesure de similarité.

En haute dimension, il est intéressant d'obtenir une matrice de similarité creuse (avec beaucoup de zéros).

- Méthodes de noyau à support compact
- voisinage ϵ: toute paire d'observation à distance au plus ϵ a une similarité de $s = 1$ et $s = 0$ sinon.
- k plus proches voisins: similarité de $S_{ij} = 1$ si l'observation $\mathbf{X}_j$ est un des k plus proches voisins d'observation $\mathbf{X}_i$ (ou vice-versa)