Examen de statistique multidimensionnelle

Groupe 8 :
HUYLENBROECK Florent
BOSSART Laurent

Juin 2020

Contents

1 Introduction

Dans le cadre de notre cours de statistique multidimensionnelle il nous a été demandé de, sur base d'un fichier de donnée nommé *XXData*.

- Effectuer une analyse univariée des données.
- Effectuer une ACP et en discuter les résultats.
- Effectuer une classification des individus et des variables et en discuter les résultats.

Pour cela, nous allons utiliser le langage de programmation *R* via l'outil *RStudio*.
Il nous a aussi été demandé de présenter une technique d'analyse multivariée non vue en cours : *CLARA* (Clustering Large Applications) et d'en décrire un exemple en R.

2 Analyse univariée des données

Pour commencer l'analyse de nos données, commencons par jeter un oeil au fichier de donnée en utilisant la fonction *head* de R.

Figure 1: Appel et résultat de la fonction head sur XXData

3 ACP

4 Classification

5 CLARA

5.1 Introduction à CLARA

Le clustering (regroupement) d'un ensemble d'objets avec CLARA (abréviation de Clustering LARge Applications) se fait en deux étapes. Premièrement, un échantillon est pioché aléatoirement dans l'ensemble des objets et regroupé en k sous-ensembles en utilisant la méthode k-medoid, qui donne aussi k objets représentatifs.

Ensuite, chaque objet n'appartenant pas a l'échantillon est affecté au plus proche des k objets représentatifs. De cette façon, on obtient un clustering de l'ensemble des données. Une mesure de la qualité de ce clustering est obtenue en calculant la distance moyenne entre chaque objet de l'ensemble des données et de son objet représentatif. Apres avoir pioché aléatoirementf et regroupé 5 échantillons, celui qui a la plus petite distance moyenne est séléctionné.

5.1.1 Méthode k-medoid en quelques mots

L'algorithme k-medoid est une approche de clustering utilisée pour partitionner un ensemble de données en k groupes ou clusters. Dans cette méthode, chaque cluster est représenté par un seul point de donnée du cluster. Ce point est ce que l'on appelle le medoid. Le terme medoid fait référence à un objet à l'intérieur du cluster pour lequel la dissimilitude moyenne entre ce point et tous les autres points du cluster est minimale. Il s'agit du point le plus au centre du cluster. Les medoids peuvent être considérés comme un example représentatif des membres de ce cluster. Cette méthode est aussi appelée PAM (Partioning Around Medoids).

5.2 Description de l'algorithme

Avec l'algorithme utilisé dans PAM, on séléctionne k objets (les medoids) qui sont représentatifs ou localisés centralement, et les k clusters sont construits autour de ces objets. L'effort principal de calcul qui sont fais dans l'algorithme PAM est une recherche parmis un grand nombre de sous-ensembles de k objets, pour un sous-ensemble produisant un regroupement satisfaisant et localement optimal. Si on augmente le nombre de données, la méthode k-medoid exacte est seulement faisable pour un nombre d'objets relativement petit car, dans le cas contraire, le temps de calcul devient énormement grand. L'allocation de la mémoire utilisée par PAM dépend principalement du nombre d'objets, qui est une fonction quadratique.

CLARA éffectue le clustering en conjonction avec la recherche par un ensemble d'objets représentatifs, qui devrait réprésenter un aspect différent de la structure de l'ensemble des données. La méthode utilisée par CLARA est la sélection aléatoire de 5 (ou plus) échantillons d'objets. La taille des échantillons dépend du nombre de clusters. Pour un clustering en k clusters, la taille de l'échantillon est donnée par $40 + 2k$. Le nombre de clusters doit varier entre 1 et 30, donc les échantillons contiennent entre 42 et 100 objets. Le choix d'utiliser une fonction du nombre de clusters pour la taille des échantillons est motivé par l'objectif d'avoir une probabilité raisonnable de trouver des objets de tous les échantillons "existants" dans au moins un des échantillons généré.

Pour la construction du premier échantillon, les objets sont séléctionnés par un nombre généré aléatoirement et ordonnés par un indice croissant. Chaque fois qu'un objet est pioché aléatoirement, on vérifie qu'il ne fait pas partie des objets déjà piochés. S'il n'avait pas encore été séléctionné, il est inséré à la bonne position dans le tableau.

Si la taille de l'échantillon est juste un petit peu plus petite que le nombre d'objets, il arrivera parfois que le même objet sera pioché plusieurs fois de manière aléatoire. C'est pour cette raison qu'à chaque fois que le nombre d'objets est inférieur au double de la taille de l'échantillon, que le générateurde nombres aléatoires est utilisé pour séléctionner les objets n'appartenants pas à l'échantillon. La construction d'autres échantillons est lancée en tenant compte des medoids qui ont été trouvés dans les échantillons précédents. A chaque étape de l'algorithme, le meilleur ensemble de medoids actuel est stocké dans un tableau. (Ce meilleur ensemble est celui pour lequel la distance moyenne pour l'ensemble des données est la plus petite trouvée jusqu'à présent). Un nouvel échantillon est construit en ajoutant des objets a ce meilleur ensemble, de la même manière que les objets ont été accumulés dans le premier échantillon.

Après le prélèvement d'un échantillon d'objets, celui-ci est divisé en k groupes en util-

isant le même algorithme que dans le programme PAM. Cet algorithme consiste en deux parties, appelées BUILD et SWAP. Dans BUILD, les objets représentatifs successifs sont séléctionnés dans le but d'obtenir la plus petite distance moyenne possible entre les objets (de l'échantillon) et leur object représentatif le plus similaire. Dans SWAP, on tente de diminuer la distance moyenne en remplacent les objets représentatifs.

Une fois que k objets représentatifs ont été séléctionnés, chaque objets de l'ensemble de données (et pas seulement de l'échantillon) est attribué à l'objet représentatif le plus proche. La distance moyenne obtenue pour l'affectation est utilisée comme une mesure de la qualité du clustering. Une fois ce calcul effectué pour chacun des 5 échantillons, l'échantillon retenu est celui avec la plus petite distance moyenne possible.

Une analyse plus poussée est alors effectuée sur la dernière partition. La liste d'objets de chaque cluster est donnée, ainsi que le medoid et la taille du cluster (dans l'ensemble des données). Le programme va alors lister, pour chaque cluster, la distance moyenne et maximale pour son medoid. Aussi, la distance maximale est divisée par la distance minimal du medoid par rapport a un autre medoid. Cette valeur donne des informations sur l'étroitesse du cluster. Une petite valeur indique un cluster très serré, alors qu'une valeur qui dépasse 1 suggère un cluster faible.

5.3 Exemple de code R

```
## générer 500 objets, divisés en 2 clusters.
x <- rbind(cbind(rnorm(200,0,8), rnorm(200,0,8)),
           cbind(rnorm(300,50,8), rnorm(300,50,8)))
clarax <- clara(x, 2)
clarax
clarax$clusinfo
plot(clarax)

## 'xclara' est un ensemble de donnés artificiel avec 3 clusters de 1000 objects bivarié
data(xclara)
(clx3 <- clara(xclara, 3))


## Essaie 100 échantillons aléatoires différents:
nSim <- 100
nCl <- 3
set.seed(421)
cl <- matrix(NA,nrow(xclara), nSim)
for(i in 1:nSim)
   cl[,i] <- clara(xclara, nCl, medoids.x = FALSE, rngR = TRUE)$cluster
tcl <- apply(cl,1, tabulate, nbins = nCl)
## ceux qui ne sont pas toujours dans le même cluster:
(iDoubt <- which(apply(tcl,2, function(n) all(n < nSim))))
if(length(iDoubt)) {
  tabD <- tcl[,iDoubt, drop=FALSE]
```

```
  dimnames(tabD) <- list(cluster = paste(1:nCl), obs = format(iDoubt))
  t(tabD)
}
```

6 Conclusion