

Méthode de partitionnement

Yanice

13 décembre 2016

Contents

Méthode de partitionnement	1
I. Méthode des centres mobiles (forgy)	1
II. Méthode de k-means	2

Méthode de partitionnement

Principe : on part d'une partition arbitraire en K classes (K choisi) que l'on améliore itérativement jusqu'à la convergence du critère choisi.(On minimise l'inertie Intra class (équivalent à on maximise l'inertie interne))

- Méthode des centres mobiles
- Méthode des k-means

Remarque :

- L'algorithme fait diminuer l'inertie interclasses à chaque itération donc l'algorithme converge. Le nombre d'itérations nécessaires est très faible.
- Inconvénients des algorithmes de partitionnement:
- **Instabilité : Le minimum obtenu est un minimum local: la répartition en classes dépend du choix initial des centres(faire tourner l'algorithme plusieurs fois pour identifier des formes fortes)**
- **Le nombre de classes est fixé par avance (on peut s'aider d'une ACP pour le déterminer)**

I. Méthode des centres mobiles (forgy)

- Initialisation : Choix aléatoire de k centres des classes.
- Itérer les deux étapes suivantes jusqu'à ce que le critère à minimiser (inertie intraclasses) ne décroisse plus de manière significative (minimum local (pas global car dépend des centres de classes initiaux)), ou bien jusqu'à atteindre un nombre d'itérations fixées:
- **Tous les individus sont affectés à la classe dont le centre est le plus proche au sens de la distance choisie.** On construit ainsi k classes d'individus
- On calcule les barycentres des classes créées qui deviennent les k nouveaux centres

II. Méthode de k-means

« k-means » (Mc Queen): les barycentres des classes ne sont pas recalculés à la fin des affectations , mais à la fin de chaque allocation d'un individu à une classe. L'algorithme est ainsi plus rapide, mais l'ordre d'apparition des individus dans le fichier n'est pas neutre.

Il est possible de réaliser plusieurs essais et de garder celui correspondant au plus grand R2. L'argument **nstart** permet de fixer le nombre d'essais - sa valeur implicite est fixée à 1. Donnez une grande valeur à ce paramètre et faites plusieurs essais en observant le R2.

```
##-----K-means-----
centreGravite=matrix(c(x1,y1,x2,y2,...),ncol=2,byrow=TRUE)
NuagedePoint=

KM=kmeans(NuagedePoint,centreGravite)

#centers=
#soit un vecteur indiquant les centres de départ, soit le nombre de classes choisies
# Dans ce cas, les centres de classes initiaux sont choisis aléatoirement.
#iter.max=
#nombre d'itérations maximale; jusqu'à convergence par défaut
#algorithm=
#« Forgy » pour centres mobiles,
#« MacQueen » pour kmeans,
#nuées dynamiques par défaut
#nstart=
#nombre d'essais effectuer et il garde le plus grand R2 de tous les essaies

#iter.max=
#indique un nombre maximal d'itérations dans la boucle du pseudo-code.

##-----Sorties-----

#Vecteur indiquant à quel Centres de classes appartient chaque point
#Affectation des points aux classes
NouvClasse=KM$cluster;NouvClasse

#Matrice des centre de classes coordonnées
MCentreClasse=KM$centers;MCentreClasse

#TSS : Variance totale = Inertie totale = Total sum of squares
Itot=KM$totss;Itot

#Inertie de chaque classe = Inertie Intra classe = Within-cluster sum of squares
Iintra_classe=KM$withinss;Iintra

#Inertie intra classe totale
Iintra=KM$tot.withinss;Iintra

#Inertie interclasse = Between-cluster sum of squares
Iinter=KM$betweenss;Iinter
Iinter==Itot-Iintra
```

```

#Nombre d'itération
n_Iteration=KM$iter;n_Iteration

##-----Critères-----

#R2
R2=Iintra/Iinter

##-----Tracé des individus et centres de classes-----
CK=#Centre de classe
X=rbind(NuagedePoint,CK)

#Donne le plus petit et le plus garnd élément du vecteur
XLIM=range(X[,1])
YLIM=range(X[,2])

plot(NuagedePoint,xlim=XLIM,ylim=YLIM,asp=1,col=rainbow(max(KM$cluster))[KM$cluster])
#Rajout des centres de classes
points(CK,pch=19,cex=4) #pour représenter les centres de classes initiaux

```

La méthode de Lloyd présente le risque de construire des classes vides. De ce point de vue, celle de Forgy, Mac Queen et Hartigan-Wong sont meilleures. Ces deux dernières présentent la particularité de recalculer le centre de classes dès qu'une classe est modifiée. Quid de leurs performances ? La méthode de Hartigan-Wong est réputée être la meilleure.

```

### 6EME ETAPE : affiche les individus sur graph acp avec les groupe trouv  avec clustering
s.label(acp$li,label=as.character(cut))

plot(acp$li[,1],acp$li[,2],type="n")
text(acp$li[,1],acp$li[,2],col=rainbow(3)[cut],cex=0.9,labels=as.character(cut))

```