TP : Clustering

Durée : 6h

L'objectif de ce TP est l'étude des deux algorithmes de clustering k-means et CAH. Les exercices à faire « à la main » permettent de mieux comprendre l'algorithme. Le logiciel R permet d'étudier le comportement des algorithmes suivant différents cas de figures illustrés par des jeux de données simulées.

Exercice 1 : Algorithme k-means

1) On considère 5 points x_1=1, x_2=2, x_3= 9, x_4=12 et x_5=20. Appliquer l'algorithme des k-means avec les valeurs de k et les points de départ suivants. Calculer le pourcentage d'inertie expliquée par la partition obtenue.
   a) k=2, g_1=1 et g_2=20
   b) k=2, g_1=2 et g_2=9
   c) k=3, g_1=1 , g_2=9 et g_3=12

Quel est le meilleur regroupement des trois ?

2) Le langage R propose un algorithme k-means basique.

`kmeans(x,centers,nstart,…)`

Entrées :

- x = données de type matrice
- centers = soit le nombre de classes, soit une matrice contenant les coordonnées des points initiaux
- nstart=nombre d'initialisations

Sorties :

- $cluster = vecteur d'entiers indiquant le numéro de la classe de chaque individu
- $centers = matrice des distances entre les individus et les centres de chaque classe
- $size = vecteur indiquant la taille de chaque classe
- $iter = nombre d'itérations

```
# a 2-dimensional example
A = matrix(rnorm(100,sd=0.3), ncol=2))          # rnorm génère une matrice de
B = matrix(rnorm(100,mean=1,sd=0.3), ncol=2) # réalisations d'une loi normale
x = rbind(A,B) # rbind concatène des matrices
```

```
plot(x)
x=scale(x) #centre et réduit
cl= kmeans(x,center=2,nstart=5)  #clustering à 2 classes
print(cl) # affiche les résultats
plot(x, col = cl$cluster) # affiche les points avec une couleur différente par classe
                          # indexée par le numéro de la classe
points(cl$centers, col = 1:2, pch = 8, cex = 2) # ajoute les centres des classes
```

a) Tester l'algorithme des kmeans sur les données simulées Test_Clusters_Distincts.txt.

b) Ecrire une fonction qui affiche la valeur du R^2 en fonction du nombre de classes. En déduire le nombre de classes le plus pertinent.

c) Tester l'algorithme des kmeans sur les données simulées Test_Clusters_Distincts.txt, Test_Clusters_Melanges.txt et Test_Clusters_Random.txt. Constater l'évolution de l'inertie expliquée.

Exercice 2 : CAH

1) On considère 5 points x1=1, x2=2, x3= 9, x4=12 et x5=20. Appliquer une méthode de classification hiérarchique ascendante en utilisant la distance Ward comme critère de dissimilarité entre classes. Tracer le dendrogramme (avec en ordonnée la distance Ward). Quel regroupement vous parait correct ?

2) Le langage R propose un algorithme CAH.

`hclust(d, method = "ward.D2",…)`

Entrées :

- d=structure de dissimilarité entre les individus générée par la fonction `dist`
- method = mesure de dissimilarité entre les classes

Sorties : plusieurs attributs décrivant l'arbre. On retient

- $height = vecteur indiquant la valeur du critère à chaque branche

Pour représenter le dendrogramme, on utilise les fonctions

`plot(tree)`

pour afficher le dendrogramme

`rect.hclust(tree,k=nclusters)`

pour ajouter les classes sur le dendrogramme. Pour récupérer les classes, on utilise la fonction

`cutree(tree, k = 2,…)`

```
# a 2-dimensional example
A = matrix(rnorm(100,sd=0.3), ncol=2))         # rnorm génère une matrice de
B = matrix(rnorm(100,mean=100,sd=10), ncol=2) # réalisations d'une loi normale
```

```
x = rbind(A,B) # rbind concatène des matrices
plot(x)
x=scale(x) #centre et réduit
distance=dist(x,"euclidean")  #crée une structure de distance entre les individus

h=hclust(distance, "ward.D2") # crée l'arbre
plot(h$height) # affiche l'évolution du critère de dissimilarité entre classes
plot(h) # affiche le dendrogramme
rect.hclust(h,k=2) # ajoute les classes

c=cutree(h,k=2) # récupère les classes
plot(x, col = c)  # affiche les points avec une couleur différente par classe indexée
                  # par le numéro de la classe
```

Tester l'algorithme CAH sur les données *Test_Clusters_Distincts.txt*, *Test_Clusters_Melanges.txt* et *Test_Clusters_Random.txt*. Constater l'évolution du dendrogramme.

Exercice 3 : Cas particuliers

1) Outliers
   a) Tester l'algorithme des kmeans sur les données Test_Clusters_Atypique.txt avec les individus n°1 et n°1499 pour initialisation
   b) Tester l'algorithme CAH sur les données Test_Clusters_Atypique.txt avec la méthode « ward.D2 » et la méthode « average ».
2) Corrélation

Tester les algorithmes des kmeans et CAH sur les données Test_Clusters_Corr.txt. Pouvez-vous trouver un paramétrage qui améliore le résultat ? Que pourrait-on faire pour améliorer le résultat ?

Exercice 4 : jeu de données

1) Appliquer les méthodes vues précédemment sur le célèbre jeu de données Iris. Comparer les classes obtenues avec l'espèce de la plante.

```
help(iris)
data(iris)
```

2) Appliquer les méthodes vues précédemment sur le jeu de données de fromages. Interpréter les clusters obtenus grâce à une ACP.