SY19 Automne 2010

TP 2 : Classification et mélange

Exercice 1.

Le jeu de données considéré est constitué de 150 iris décrits par quatre variables : longueur des sépales, largeur des sépales, longueur des pétales et largeur des sépales. Charger le jeu de données, sélectionner les variables quantitatives et normaliser en utilisant le code R suivant :

```
> library(MASS)
> data(iris)
> donnees$num <- iris[,c(1:4)]
> donnees$cls <- iris[,5]
```

1. Tenter une partition en $K \in \{2, 3, 4\}$ classes avec la fonction `kmeans` ; visualiser et commenter.
2. Étudier la stabilité du résultat de la partition : effectuer $n = 100$ classifications en trois classes du jeu de données ; pour chacune, stocker en mémoire les partitions obtenues et la valeur d'inertie intra-classes. Commenter et interpréter.
3. Choix du nombre de classes optimal : calculer la valeur moyenne d'inertie intra-classe obtenue sur $n = 100$ classifications, pour $K \in \{2, 3, 4, 5\}$ classes. Représenter la variation de l'inertie moyenne en fonction de K. Proposer un nombre de classes en se basant sur cette courbe.
4. Comparer les résultats de la partition obtenue par les centres mobiles avec la partition réelle des iris en trois groupes.

Exercice 2.

L'objectif de cet exercice est d'implémenter et d'étudier les algorithmes EM et CEM dans le cas de données monodimensionnelles ($p = 1$).

Données synthétiques

On s'intéressera dans un premier temps à un jeu de données synthétique x issu d'un mélange gaussien de deux classes présentes en proportions identiques $\pi_1 = \pi_2 = \frac{1}{2}$:

```
x<-c(rnorm(1000),rnorm(1000,mean=6,sd=5))
```

En reprenant les exemples, vus en cours, de l'algorithme EM :

1. implémenter les algorithmes EM et CEM en complétant la fonction gmixtmono dont l'ossature est fournie dans le fichier mixtures.r (disponible sur le site de SY19) ;
2. comparer les paramètres du modèle de mélange (μ_k et σ_k, pour chaque classe ω_k) calculés à l'étape M dans le cas de l'algorithme EM à ceux obtenus dans le cas de l'algorithme CEM ;
3. comparer les partitions en deux classes obtenues par l'algorithme des *k-means*, par l'algorithme EM en utilisant la règle du MAP (*maximum a posteriori*), et par l'algorithme CEM ; discuter.

Application à des données réelles : traitement d'images

Le jeu de données lena correspond à une image fréquemment utilisée pour illustrer les propriétés de méthodes de traitement d'images. Télécharger le fichier correspondant sur le site de SY19, et charger les données au moyen du code suivant :

```
lena <- read.csv("lena_bw.csv", header=F)
lena <- as.matrix(lena)
```

Rappelons qu'une image en niveaux de gris (image NG) se présente sous la forme d'un tableau à deux dimensions contenant des nombres entiers compris entre 0 (couleur noire) et 255 (couleur blanche). Les fonctions implotbw, immat2imvec et imvec2immat (disponibles dans le fichier fct_images.r, téléchargeable sur le site de SY19) permettent respectivement : d'afficher une image en niveaux de gris décrite par une matrice d'entiers, de transformer une image NG en vecteur d'entiers, et réciproquement de transformer un vecteur d'entiers en image NG.

Effectuer une classification des pixels de l'image lena au moyen des algorithmes EM et CEM. Afficher et comparer les résultats obtenus.

Exercice 3.

L'objectif de cet exercice est d'implémenter et d'étudier les algorithmes EM et CEM dans le cas le plus général d'un mélange gaussien multidimensionnel (matrices de variance-covariance quelconques). On dispose pour cela de la fonction mvdnorm fournie lors du premier TP, et de la fonction gmixtmulti à compléter (l'ossature de cette fonction est fournie dans le fichier mixtures.r).

1. Implémenter les algorithmes EM et CEM, à chaque fois en complétant la fonction gmixtmulti.
2. On considère dans un premier temps un jeu de données synthétiques en dimension $p = 2$, contenu dans le fichier toydata.csv disponible sur le site de l'UV. Charger les données au moyen du code suivant :

   ```
   > Eapp <- read.csv("toydata.csv")
   > Xapp <- as.matrix(Eapp[,1:2])
   > yapp <- as.vector(Eapp[,3])
   ```

   Effectuer la classification des données en $K \in 2, 3, 4$ classes par l'algorithme des centres mobiles, l'algorithme EM, et l'algorithme CEM. Analyser et comparer les résultats obtenus.
3. On s'intéresse à présent au jeu de données des crabes, qui mesure cinq attributs morphologiques sur 200 crabes de deux espèces et de deux sexes différents :

   ```
   > library(MASS)
   > data(crabs)
   > crabsquant <- crabs[,c(4:8)]/rowSums(crabs[,c(4:8)])
   ```

   Effectuer une classification des données en $K = 4$ classes, au moyen de l'algorithme des centres mobiles, de l'algorithme EM, et de l'algorithme CEM. Analyser et comparer les résultats obtenus.