TP 2 : Modèles de mélange, Model-Based Clustering, algorithme EM

L'objectif de ce TP est d'abord d'explorer le comportement de l'algorithme EM dans le cadre d'un modèle simple de mélange gaussien à $J \in \mathbb{N}^*$ composantes en dimension 1 :

$$\mathcal{M} = \left\{ \sum_{j=1}^{J} \pi_j \phi(\cdot; \mu_j, \sigma_j^2) : (\pi_1, \dots, \pi_J) \in \Pi_J, \mu_1, \dots, \mu_J \in \mathbb{R}, \sigma_1^2, \dots, \sigma_J^2 \in \mathbb{R}^{+*} \right\}$$

avec $\Pi_J = \{(\pi_1, \dots, \pi_J) \in [0,1]^J : \sum_{j=1}^{J} \pi_j = 1\}$. C'est ensuite de s'initier au model-based clustering en dimension supérieure avec le package Rmixmod, qui permet d'ajuster des modèles de mélange tout à fait dans l'esprit présenté en cours. N'oubliez pas d'utiliser l'aide fournie par les auteurs du package (directement depuis R, ou dans le pdf fourni sur la page Internet de Mixmod).
Certains étudiants des années précédentes préféraient travailler avec le package mclust, qui est excellent également, et qui permet de mettre en œuvre une approche model-based clustering tout à fait proche de celle présentée en cours (avec des nuances concernant les modèles de mélange utilisés : attention à cela si vous utilisez ce package). Je n'y vois aucun inconvénient.
Outre les questions posées, n'hésitez pas à laisser libre cours à votre curiosité.

Modèles de mélange

1. Tracer la densité d'une loi de mélange (trois composantes par exemple) en dimension 1. Jouer avec les paramètres : obtenir une densité unimodale, bimodale, « oblique »...
2. Simuler des échantillons de taille 200 de la loi de densité

   $$f\big(\,.\,; (1/3, 1/6, 1/2, 0, 5, 10, 1, 1, 4)\big) = \frac{1}{3}\phi(\,.\,; 0, 1) + \frac{1}{6}\phi(\,.\,; 5, 1) + \frac{1}{2}\phi(\,.\,; 10, 4)$$

   et d'autres et les représenter graphiquement.
3. Exécuter ces simulations de façon à conserver la « vraie » composante de simulation de chaque observation (les z_i). Représenter les observations avec leur composante sur un même graphique. Utiliser par exemple des couleurs.
4. Comparer les « vraies » classes et celles obtenues par la règle du MAP calculée avec les vrais paramètres du mélange.
5. Représenter la vraisemblance. L'objectif est notamment d'observer les maxima locaux. Pour cela, fixer par exemple tous les paramètres à leur vraie valeur, sauf deux. Interpréter certains de ce ces maxima.

6. On s'intéresse dans cette question à la situation en dimension 2. Choisir un mélange de deux composantes bien séparées dans le modèle $[p\_L_k\_B_k]$. Notons p la proportion d'une des deux composantes au choix et λA la décomposition de sa matrice de covariances.
   (a) Représenter une isodensité ainsi qu'un échantillon de taille 1000 de cette loi.
   (b) Doubler λ et représenter une isodensité ainsi qu'un échantillon de taille 1000 de la loi obtenue.
   (c) À nouveau avec la valeur d'origine de λ, remplacer la proportion p par $\frac{p}{2}$ et représenter une isodensité ainsi qu'un échantillon de taille 1000 de la loi obtenue.

Algorithme EM, dimension 1

1. Implémentation de l'algorithme EM (en dimension 1).
   (a) Comment s'écrit l'algorithme EM dans ce cadre ? Faire (avec un papier et un crayon) les calculs nécessaires pour toutes les étapes.
   (b) Coder l'algorithme EM obtenu sous forme d'une fonction qui prend en entrées les données sous forme d'un data.frame ou une matrice ainsi que le nombre de composantes voulues, et produit en sortie les paramètres du mélange ajusté. Fournir une fonction qui prenne ces sorties en entrée, et qui représente graphiquement la densité du mélange ajusté ainsi que la classification des observations par MAP.
2. Jouer avec cet EM sur des données simulées. On pourra par exemple :
   (a) Suivre l'évolution de la log-vraisemblance au cours des itérations ;
   (b) Essayer de lancer l'algorithme avec plusieurs paramètres initiaux différents ;
   (c) Représenter la densité ajustée ;
   (d) Comparer les classifications obtenues par MAP ;
   (e) Trouver un cas de croissance avec palier de la log-vraisemblance lors des itérations de EM.

Algorithme EM, dimension quelconque (Mixmod)

1. Il faut tout d'abord vous assurer que le package *Rmixmod* soit installé sur la machine sur laquelle vous travaillez.
2. Essayer *Rmixmod* sur des données simulées et comparer la densité du mélange estimé à la vraie densité de l'échantillon (graphiquement par exemple).
3. Jouer avec les arguments d'entrée de *Rmixmod*, et notamment *nbCluster* et *models*.
4. Que fait *Rmixmod* lorsque vous donnez en argument d'entrée $nbCluster = 2 : 8$?

5. Nous nous intéressons maintenant au choix du nombre de classes.

   (a) Pour un mélange unidimensionnel de deux composantes gaussiennes de variance 1 dans des proportions égales, à quelles conditions BIC sélectionne-t-il le vrai nombre de composantes ? De même pour ICL. Vous répondrez par une étude numérique basée sur des simulations bien choisies.

   (b) On s'intéresse maintenant à des données multidimensionnelles, par exemple les données *seeds*. Utiliser *Rmixmod* pour sélectionner le nombre de classes et la forme du modèle par BIC d'une part ; par ICL d'autre part. Visualiser les résultats obtenus.

   (c) Même question mais en sélectionnant uniquement le nombre de classes, la forme de modèle étant fixée (par exemple, pour la forme [pkLI] d'une part ; pour la forme [pkLkC] d'autre part).

Question bonus

On se propose de montrer que des données pourtant gaussiennes peuvent sembler provenir d'un mélange de plusieurs composantes bien séparées... Que proposez-vous de simuler et comment le représenter pour augmenter les chances d'observer un tel phénomène ?
Indice. Simuler en grande dimension et utiliser *Rmixmod*...