Projet Master 1 SSD

Un modèle pour les nids de mouettes

Rédigé par

CARVAILLO Thomas

CÔME Olivier

PRALON Nicolas

Encadrante : Elodie Brunel-Piccinini

Table des matières

Introduction

Chapitre 1

Modélisation du problème et liminaire mathématique

Le présent chapitre s'inspirera indifféremment des références [1], [2] et [3]

1.1 Modélisation du problème

Nous allons pour commencer donner une première définition, qui est au coeur du présent projet.

Définition 1 (Loi de mélange)**.** On appelle loi de mélange toute loi dont la densité s'écrit sous la forme d'une combinaison convexe de plusieurs densités. Si l'on se donne J densités $f_1(x), \cdots, f_J(x)$, alors toute variable aléatoire X dont la densité f s'exprime, pour tout $x \in \mathbb{R}$, sous la forme

$$f(x) := \sum_{i=1}^{J} \alpha_i f_i(x) \ , \ \alpha_i \in \mathbb{R}_+^* \text{ et } \sum_{i=1}^{n} \alpha_i = 1$$

suit une loi de mélange.

Afin de modéliser commodément le problème, nous introduisons les variables aléatoires suivantes :

- La variable aléatoire X, modélisant la taille des nids, de densité f
- Z, la variable aléatoire discrète et à valeurs dans $[\![1, J]\!]$, représentant l'espèce de mouette qui a construit le nid

Enfin, nous nous placerons sous les hypothèses suivantes :

Hypothèse 1. *Nous supposerons que, $\forall j \in [\![1, J]\!]$, la taille des nids d'une espèce j (i.e. X conditionnellement à $(Z = j)$) suit une loi normale $\mathcal{N}(\mu_j, v_j)$. Nous dénoterons par $f(x|Z = j) := \gamma_{\mu_j, v_j}(x)$ cette densité.*

Hypothèse 2. *Soit $\Theta := \{\theta = (\alpha_j, \mu_j, v_j)_{1 \leq j \leq J}$ tels que $\alpha_j > 0 \ \forall j \in [\![1, J]\!]$ et $\sum_{j=1}^{J} \alpha_j = 1\}$. Soient $X_1, \cdots, X_n$ un échantillon de même loi que X. On supposera qu'il existe un $\theta \in \Theta$ tel que les données récoltées, ici les tailles des nids, soient la réalisation du précédent échantillon.*

Remarque 1. La variable Z est discrète et à valeur dans un sous-ensemble fini de $\mathbb{N}$, elle suit donc une loi

$$\sum_{j=1}^{J} \alpha_j \delta_j$$

où J représente le nombre d'espèce de mouettes considéré et les α_j sont des réels, positifs stricts, représentant la proportion de nids de l'espèce j, tels que $\sum_{i=1}^{J} \alpha_j = 1$.

Il s'ensuit la proposition suivante, qui sera essentielle dans la suite.

Proposition 1. *La distribution de la taille des nids de mouettes, i.e. X, admet pour densité ,au point x et par rapport à la mesure de Lebesgue sur $\mathbb{R}$, la fonction f_θ définie comme suit*

$$f_\theta(x) = \sum_{j=1}^{J} \alpha_j \gamma_{\mu_j, v_j}(x)$$

Démonstration. En effet,

$$\begin{aligned}
\mathbb{P}(X \leq x) &= \mathbb{P}\left(\bigcup_{j=1}^{J} (X \leq x) \cap (Z = j)\right) \\
&= \sum_{j=1}^{J} \mathbb{P}((X \leq x) \cap (Z = j)) \\
&= \sum_{i=1}^{J} \mathbb{P}(Z = j) \times \mathbb{P}(x \leq X | Z = j) \\
&= \sum_{i=1}^{J} \alpha_i \times \mathbb{P}(x \leq X | Z = j) \\
&= \sum_{i=1}^{J} \alpha_i \times f(x | Z = j)
\end{aligned}$$

❧

Le but de ce projet sera d'étudier des méthodes permettant l'estimation des divers paramètres de cette densité. Nous dénoterons par $\theta := (\alpha_j, \mu_j, v_j)_{1 \leq j \leq J}$ le vecteur des paramètres.

1.2 Une histoire de densités

Introduisons une dernière densité et une dernière probabilité, qui nous seront fort utile dans la suite :

Proposition 2. *Nous avons les résultats suivant :*

1. *La loi du couple (X, Z) est donnée par*

$$\begin{aligned}
\mathbb{P}(X \leq x, Z = j) &= \int_{-\infty}^{x} f_\theta(u | Z = j) \times \mathbb{P}(Z = j) du \\
&= \int_{-\infty}^{x} \underbrace{\gamma_{\mu_j, v_j}(u) \times \alpha_j}_{:= h_\theta(u, j)} du
\end{aligned}$$

   où $h_\theta(u, j)$ est la "sous-densité" de $X \times \mathbb{1}_{(Z=j)}$

2. *La probabilité de la loi de Z sachant $X = x$ est donnée par :*

$$\mathbb{P}_\theta(Z = j | X = x) = \frac{\gamma_{\mu(z), v(z)} \times \alpha_z}{f_\theta(x)}$$

   où $f_\theta(x)$ est donnée par la proposition 1.

Démonstration. En effet,

$$\mathbb{P}(X \leq x, Z = j) = \int_{-\infty}^{x} \mathbb{P}(Z = j | X = u) \times f_\theta(u) du$$

On obtient dès lors

$$\mathbb{P}(Z = j | X = u) \times f_\theta(u) = \gamma_{\mu_j, v_j}(u) \times \alpha_j$$

Soit

$$\mathbb{P}(Z=j|X=u) = \frac{\gamma_{\mu_j,v_j}(u)\times\alpha_j}{f_\theta(u)} := \frac{h_\theta(u,j)}{f_\theta(u)}$$

❧

Remarque 2. Nous pouvons dès à présent noter que pour un échantillon $X_1,\cdots,X_n$ de même loi que X, nous avons

$$\forall i\in[\![1,n]\!],\ h_\theta(X_i,j) = f_\theta(X_i)\times\mathbb{P}_\theta(Z=j|X=X_i)$$

Ceci nous sera utile dans la suite.

Nous allons dès à présent nous intéresser à l'estimation de ces paramètres.

1.3 Une approche idéaliste

Regardons dans un premier temps un cas simplifié, un cas ne décrivant pas la réalité des observations mais qui a le mérite de constituer une agréable entrée en matière.
Nous supposerons ici qu'ont été relevés simultanément et les mesures des tailles des nids et l'espèce de mouette qui l'a construit. Les observations considérées ici sont donc composées des couples (X_i,Z_i), $i\in[\![1,n]\!]$. On considérera dès lors la fonction de densité $h_\theta(x,z)$, donnée par laproposition 2.
L'estimation des divers paramètres est alors élémentaire, en témoigne les propositions suivantes :

Proposition 3 (Fonction de Log-vraisemblance)**.** *La Log-vraisemblance du modèle s'écrit*

$$\mathscr{L}_\theta(X_1,\cdots,X_n,Z_1,\cdots,Z_n) = \sum_{j=1}^{J}\#A_j ln(\alpha_j) + \sum_{j=1}^{J}\sum_{i\in A_j} ln(\gamma_{\mu_j,v_j}(X_i))$$

où les A_j sont définis par $A_j := \{i\in[\![1,n]\!]$ tels que $Z_i=j\}$ i.e. $\bigcup_{j=1}^{J} A_j = [\![1,n]\!]$

Avant de démontrer cette proposition, nous allons introduire une notation qui nous sera immédiatement utile :

Notation 1. Dans ce qui suit, nous noterons

$$\delta_j := \mathbb{1}_{(Z_i=j)}(Z_i)$$

Ainsi,

$$h_\theta(X_i,Z_i) = \prod_{j=1}^{J} h_\theta(X_i,Z_i)^{\delta_j}$$

Démonstration. La Log-vraisemblance du modèle s'écrit :

$$\begin{aligned}\mathscr{L}_\theta(X_1,\cdots,X_n,Z_1,\cdots,Z_n) &= ln\left(\prod_{i=1}^{n} h_\theta(X_i,Z_i)\right)\\ &= ln\left(\prod_{i=1}^{n}\prod_{j=1}^{J} h_\theta(X_i,Z_i)^{\delta_j}\right)\\ &= \mathbb{1}_{(Z_i=j)}(Z_i)\times ln\left(\prod_{i=1}^{n}\prod_{j=1}^{J} h_\theta(X_i,Z_i)\right)\end{aligned}$$

Z_i est à valeur dans $j\in[\![1,J]\!]$, on partitionne donc $I := [\![1,n]\!]$ comme $I = \bigcup_{j=1}^{J} A_j$.

Ceci va nous permettre de nous désencombrer de l'indicatrice en réindexant la somme. Nous obtenons dès lors :

$$\begin{aligned}
\mathscr{L}_\theta(X_1,\cdots,X_n,Z_1,\cdots,Z_n) &= ln\left(\prod_{i\in A_i}\prod_{j=1}^{J} h_\theta(X_i,Z_i)\right)\\
&= ln\left(\prod_{i\in A_i}\prod_{j=1}^{J} \alpha_j\gamma_{\mu_j,v_j}(X_i)\right)\\
&= \sum_{i\in A_i}\sum_{j=1}^{J} ln(\alpha_j\gamma_{\mu_j,v_j}(X_i))\\
&= \sum_{j=1}^{J}\sum_{i\in A_j} ln(\alpha_j) + \sum_{j=1}^{J}\sum_{i\in A_j} ln(\gamma_{\mu_j,v_j}(X_i))\\
&= \sum_{j=1}^{J}\#A_j ln(\alpha_j) + \sum_{j=1}^{J}\sum_{i\in A_j} ln(\gamma_{\mu_j,v_j}(X_i))
\end{aligned}$$

❧

Nous pouvons dès lors maximiser la log-vraisemblance afin d'obtenir les estimateurs souhaités :

Proposition 4 (Estimateurs)**.** *Les estimateurs du maximum de vraisemblance $\widehat{\alpha_j}$ (resp. $\widehat{\mu_j}$, et $\widehat{v_j}$) de α_j (resp. μ_j et v_j) sont donnés par*

$$\widehat{\alpha_j} = \tfrac{\#A_j}{n}$$

$$\widehat{\mu_j} = \frac{\sum_{i\in A_j} X_i}{\#A_j}$$

$$\widehat{v_j} = \frac{\sum_{i\in A_j}(X_i-\widehat{\mu_j})^2}{\#A_j}$$

Démonstration. Soit $\theta = (\alpha_j,\mu_j,v_j)_{j\in[\![1,J]\!]}$. Il s'agit de déterminer

$$\underset{\theta\in\mathbb{R}^{3J},\sum_{j=1}^{J}\alpha_j=1}{\operatorname{argmax}}\left(\sum_{j=1}^{J}\#A_j ln(\alpha_j) + \sum_{j=1}^{J}\sum_{i\in A_j} ln(\gamma_{\mu_j,v_j}(X_i))\right)$$

Nous avons donc à résoudre un programme de minimisation d'une fonction convexe sur un convexe avec une contraire égalité, il est ainsi naturel de faire appel au Lagrangien.
Ce dernier s'écrit

$$\begin{aligned}
L(\theta) &= \sum_{j=1}^{J}\#A_j ln(\alpha_j) + \sum_{j=1}^{J}\sum_{i\in A_j} ln(\gamma_{\mu_j,v_j}(X_i)) - \lambda\times\left(\sum_{j=1}^{J}\alpha_j - 1\right)\\
&= \sum_{j=1}^{J}\#A_j ln(\alpha_j) + \sum_{j=1}^{J}\sum_{i\in A_j} ln\left(\frac{1}{\sqrt{2\pi v_j}}\exp\left(-\frac{(X_i-\mu_j)^2}{2v_j}\right)\right) - \lambda\times\left(\sum_{j=1}^{J}\alpha_j - 1\right)\\
&= \sum_{j=1}^{J}\#A_j ln(\alpha_j) + \sum_{j=1}^{J}\sum_{i\in A_j}\left(\frac{-1}{2}ln(2\pi v_j) - \frac{(X_i-\mu_j)^2}{2v_j}\right) - \lambda\times\left(\sum_{j=1}^{J}\alpha_j - 1\right)
\end{aligned}$$

Il reste maintenant à résoudre le système suivant, afin d'obtenir le vecteur $\widehat{\theta} := (\widehat{\alpha_j},\widehat{\mu_j},\widehat{v_j})_{j\in[\![1,J]\!]}$ solution du programme.

$$\begin{cases} \dfrac{\#A_j}{\widehat{\alpha_j}} - \lambda & = 0 \ \forall j \in [\![1, J]\!] \\ \sum\limits_{i \in A_j} (X_i - \widehat{\mu_j})/\widehat{v_j} & = 0 \ \forall j \in [\![1, J]\!] \\ \sum\limits_{i \in A_j} \dfrac{-0.5 \times 2 \times \pi}{2\pi \widehat{v_j}} + \dfrac{(X_i - \widehat{\mu_j})^2}{2\widehat{v_j}^2} & = 0 \ \forall j \in [\![1, J]\!] \\ \sum\limits_{j=1}^{J} \widehat{\alpha_j} = 1 \end{cases}$$

Ceci équivaut à

$$\begin{cases} \dfrac{\#A_j}{\widehat{\alpha_j}} & = \lambda \ \forall j \in [\![1, J]\!] \\ \sum\limits_{i \in A_j} X_i & = \sum\limits_{i \in A_j} \widehat{\mu_j} \ \forall j \in [\![1, J]\!] \\ \sum\limits_{i \in A_j} (X_i - \widehat{\mu_j})^2 & = \sum\limits_{i \in A_j} \widehat{v_j} \ \forall j \in [\![1, J]\!] \\ \sum\limits_{j=1}^{J} \widehat{\alpha_j} = 1 \end{cases} \Leftrightarrow \begin{cases} \dfrac{\#A_j}{\widehat{\alpha_j}} & = \lambda \ \forall j \in [\![1, J]\!] \\ \sum\limits_{i \in A_j} \dfrac{X_i}{\#A_j} & = \widehat{\mu_j} \ \forall j \in [\![1, J]\!] \\ \sum\limits_{i \in A_j} \dfrac{(X_i - \widehat{\mu_j})^2}{\#A_j} & = \widehat{v_j} \ \forall j \in [\![1, J]\!] \\ \sum\limits_{j=1}^{J} \widehat{\alpha_j} = 1 \end{cases}$$

En sommant les J premières lignes du système, on obtient $\sum\limits_{j=1}^{J} \#A_j = \sum\limits_{j=1}^{J} \widehat{\alpha_j}\lambda$, i.e. $\lambda = n$. En injectant ceci dans le précédent système, on obtient finalement ce qui était annoncé :

$$\begin{cases} \widehat{\alpha_j} & = \dfrac{\#A_j}{n} \ \forall j \in [\![1, J]\!] \\ \widehat{\mu_j} & = \sum\limits_{i \in A_j} \dfrac{X_i}{\#A_j} \ \forall j \in [\![1, J]\!] \\ \widehat{v_j} & = \sum\limits_{i \in A_j} \dfrac{(X_i - \widehat{\mu_j})^2}{\#A_j} \ \forall j \in [\![1, J]\!] \end{cases}$$

1.4 Une situation concordante à la réalité

Nous nous placerons désormais dans un contexte tout autre que celui du paragraphe précédent, un contexte concordant davantage à la réalité. Dans ce qui suit, nous supposerons que ne sont observées que les tailles des nids, les diverses espèces de mouettes les ayant construit étant en quelque sorte des données inobservées ou "cachées". Nous avons donc un échantillon $X_1, \cdots, X_n$ de même loi que la variable X comme définie ci-dessus. On définit $\mathscr{L}_{obs}$ la log-vraisemblance des observations, nous obtenons ainsi

Définition 2. La log-vraisemblance des observations s'écrit

$$\mathscr{L}_{obs}(\theta, X_1, \cdots, X_n) := ln\left(\prod_{i=1}^{n} f_\theta(X_i)\right) = \sum_{i=1}^{n} ln\left(\sum_{j=1}^{J} \alpha_j \gamma_{\mu_j, v_j}(X_i)\right)$$

Nous voyons dès lors que l'existence d'une expression analytique du maximum de la log-vraisemblance n'est pas assurée. Il est donc nécessaire de trouver un moyen d'approcher les valeurs des différents estimateurs.
Pour ce faire, on définit une log-vraisemblance des couples (X_i, Z_i) sachant le vecteurs des observations $X_1, \cdots, X_n$.

Proposition 5 (log-vraisemblance conditionnelle). *On définit la log-vraisemblance $\mathscr{L}_c(\theta, \tilde{\theta}, X_1, \cdots, X_n)$ conditionnelle par*

$$\mathscr{L}_c(\theta, \tilde{\theta}, X_1, \cdots, X_n) = \mathbb{E}_{\tilde{\theta}}[\mathscr{L}_\theta(X_1, \cdots, X_n, Z_1, \cdots, Z_n) | X_1, \cdots, X_n]$$

Nous allons maintenant travailler sur l'expression de la log-vraisemblance conditionnelle et en donner une expression simplifiée, qui nous sera fort utile ultérieurement, et une expression plus substancielle, qui nous sera immédiatement utile.

Proposition 6. *Nous avons*

$$\mathscr{L}_c(\theta, \tilde{\theta}, X_1, \cdots, X_n) = \sum_{i=1}^{n} \sum_{j=1}^{J} ln(h_\theta(X_i, j))\mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i)$$

Démonstration. En effet

$$\begin{aligned}
\mathscr{L}_c(\theta, \tilde{\theta}, X_1, \cdots, X_n) &= \mathbb{E}_{\tilde{\theta}}[\mathscr{L}_\theta(X_1, \cdots, X_n, Z_1, \cdots, X_n)|X_1, \cdots, X_n] \\
&= \mathbb{E}_{\tilde{\theta}}\left[ln \left(\prod_{i=1}^{n} \prod_{j=1}^{J} h_\theta(X_i, Z_i)^{\delta_j} \right) \middle| X_1, \cdots X_n \right] \\
&= \sum_{i=1}^{n} \sum_{j=1}^{J} \mathbb{E}_{\tilde{\theta}}\left[\, \delta_j \times ln(h_\theta(X_i, Z_i)) \right| X_1, \cdots, X_n] \\
&= \sum_{i=1}^{n} \sum_{j=1}^{J} \mathbb{E}_{\tilde{\theta}}\left[\, \mathbb{1}_{(Z_i=j)}(Z_i) \times ln(h_\theta(X_i, Z_i)) \right| X_1, \cdots, X_n]
\end{aligned}$$

Or, les couples (X_i, Z_i) sont indépendant, donc

$$\begin{aligned}
\mathscr{L}_c(\theta, \tilde{\theta}, X_1, \cdots, X_n) &= \sum_{i=1}^{n} \sum_{j=1}^{J} \mathbb{E}_{\tilde{\theta}}\left[\, \mathbb{1}_{(Z_i=j)}(Z_i) \times ln(h_\theta(X_i, Z_i)) \right| X_i] \\
&= \sum_{i=1}^{n} \sum_{j=1}^{J} \mathbb{E}_{\tilde{\theta}}\left[\, \mathbb{1}_{(Z_i=j)}(Z_i) \times ln(h_\theta(X_i, j)) \right| X_i] \\
&= \sum_{i=1}^{n} \sum_{j=1}^{J} ln(h_\theta(X_i, j))\mathbb{E}_{\tilde{\theta}}\left[\, \mathbb{1}_{(Z_i=j)}(Z_i) \right| X_i] \\
&= \sum_{i=1}^{n} \sum_{j=1}^{J} ln(h_\theta(X_i, j))\mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i)
\end{aligned}$$

❧

Nous nous appuierons sur l'expression suivante pour l'expression des estimateurs du maximum de vraisemblance :

Proposition 7. *La fonction* $\mathscr{L}_c(\theta, \tilde{\theta}, X_1, \cdots, X_n)$ *se réécrit sous la forme suivante :*

$$\begin{aligned}
\mathscr{L}_c(\theta, \tilde{\theta}, X_1, \cdots, X_n) = &-\frac{n}{2} ln(2\pi) + \sum_{j=1}^{J} \left(\sum_{i=1}^{n} \mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i) \right) ln(\alpha_j) \\
&- \frac{1}{2} \sum_{j=1}^{J} \left(\sum_{i=1}^{n} \mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i) \left(log(v_j) + \frac{(X_i - \mu_j)^2}{v_j} \right) \right)
\end{aligned}$$

Démonstration. Il suffit de partir de la forme précédente de la log-vraisemblance conditionnelle, on a ainsi :

$$
\begin{aligned}
\mathscr{L}_c(\theta,\tilde{\theta},X_1,\cdots,X_n) &= \sum_{i=1}^{n}\sum_{j=1}^{J} ln(h_\theta(X_i,j))\mathbb{P}_{\tilde{\theta}}(Z=j|X=X_i)\\
&= \sum_{i=1}^{n}\sum_{j=1}^{J} ln(\alpha_j\gamma_{\mu_j,v_j}(X_i))\times\mathbb{P}_{\tilde{\theta}}(Z=j|X=X_i)\\
&= \sum_{i=1}^{n}\sum_{j=1}^{J} \big(ln(\alpha_j)+ln(\gamma_{\mu_j,v_j}(X_i))\big)\times\mathbb{P}_{\tilde{\theta}}(Z=j|X=X_i)\\
&= \sum_{i=1}^{n}\sum_{j=1}^{J} ln(\alpha_j)\mathbb{P}_{\tilde{\theta}}(Z=j|X=X_i)+\sum_{i=1}^{n}\sum_{j=1}^{J} ln(\gamma_{\mu_j,v_j}(X_i))\mathbb{P}_{\tilde{\theta}}(Z=j|X=X_i)
\end{aligned}
$$

Traitons pour commencer la double somme

$$
\Delta := \sum_{i=1}^{n}\sum_{j=1}^{J} ln(\gamma_{\mu_j,v_j}(X_i))\mathbb{P}_{\tilde{\theta}}(Z=j|X=X_i)
$$

Nous avons :

$$
\gamma_{\mu_j,v_j}(X_i) = \tfrac{1}{\sqrt{2\pi v(j)}}e^{-\frac{1}{2}\frac{(X_i-\mu_j)^2}{v_j}}
$$

$$
\mathbb{P}_{\tilde{\theta}}(Z=j|X=X_i) = \frac{\alpha_j\gamma_{\mu_j,v_j}}{\displaystyle\sum_{i=1}^{J}\alpha_j\gamma_{\mu_i,v_i}}
$$

La double somme devient alors

$$
\begin{aligned}
\Delta &= \sum_{i=1}^{n}\sum_{j=1}^{J} ln\left(\frac{1}{\sqrt{2\pi v_j}}e^{-\frac{1}{2}\frac{(X_i-\mu_j)^2}{v_j}}\right)\times\mathbb{P}_{\tilde{\theta}}(Z=j|X=X_i)\\
&= \sum_{i=1}^{n}\sum_{j=1}^{J} ln\left(\frac{1}{\sqrt{2\pi v_j}}\right)\times\mathbb{P}_{\tilde{\theta}}(Z=j|X=X_i)-\frac{1}{2}\left(\frac{(X_i-\mu_j)^2}{v_j}\right)\times\mathbb{P}_{\tilde{\theta}}(Z=j|X=X_i)\\
&= \sum_{i=1}^{n}\sum_{j=1}^{J} -\frac{1}{2}ln(2\pi)\times\mathbb{P}_{\tilde{\theta}}(Z=j|X=X_i)-\frac{1}{2}ln(v_j)\times\mathbb{P}_{\tilde{\theta}}(Z=j|X=X_i)-\frac{1}{2}\left(\frac{(X_i-\mu_j)^2}{v_j}\right)\times\mathbb{P}_{\tilde{\theta}}(Z=j|X=X_i)\\
&= -\frac{n}{2}ln(2\pi)-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{J}\left(ln(v_j)+\frac{(X_i-\mu_j)^2}{v_j}\right)\times\mathbb{P}_{\tilde{\theta}}(Z=j|X=X_i)\\
&= -\frac{n}{2}ln(2\pi)-\frac{1}{2}\sum_{j=1}^{J}\left(\sum_{i=1}^{n}\mathbb{P}_{\tilde{\theta}}(Z=j|X=X_i)\times\left(ln(v_j)+\frac{(X_i-\mu_j)^2}{v_j}\right)\right)
\end{aligned}
$$

On obtient de fait le résultat espéré :

$$
\begin{aligned}
\mathscr{L}_c(\theta,\tilde{\theta},X_1,\cdots,X_n) = -\frac{n}{2}log(2\pi)&+\sum_{j=1}^{J}\left(\sum_{i=1}^{n}\mathbb{P}_{\tilde{\theta}}(Z=j|X=X_i)\right)\times ln(\alpha_j)\\
&-\frac{1}{2}\sum_{j=1}^{J}\left(\sum_{i=1}^{n}\mathbb{P}_{\tilde{\theta}}(Z=j|X=X_i)\times\left(ln(v_j)+\frac{(X_i-\mu_j)^2}{v_j}\right)\right)
\end{aligned}
$$

Nous allons dès à présent énoncer une proposition vitale, celle de l'expression des estimateurs du maximum de vraisemblance de la log-vraisemblance conditionnelle. L'expression de ces derniers seront le pivot de l'algorithme EM, que nous présenterons dans le chapitre suivant.

Proposition 8. *La fonction $\theta \mapsto \mathscr{L}_c(\theta, \tilde{\theta}, X_1, \cdots, X_n)$ admet un unique maximum θ_M donné par :*

$$\hat{\alpha_j} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i)$$

$$\hat{\mu_j} = \frac{\displaystyle\sum_{i=1}^{n} X_i \mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i)}{\displaystyle\sum_{i=1}^{n} \mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i)}$$

$$\hat{v_j} = \frac{\displaystyle\sum_{i=1}^{n} (X_i - \mu_j)^2 \mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i)}{\displaystyle\sum_{i=1}^{n} \mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i)}$$

Démonstration. Soit $\theta = (\alpha_j, \mu_j, v_j)$. Il s'agit ici de maximiser la fonction $\theta \mapsto \mathscr{L}_c(\theta, \tilde{\theta}, X_1, \cdots, X_n)$

Puisqu'il s'agit d'un problème d'optimisation, nous appliquons la même méthode que précédemment, en introduisant le Lagrangien du problème sous la contrainte $\displaystyle\sum_{i=1}^{n} \alpha_i = 1$.

Nous reprenons ici l'écriture de $\mathscr{L}_c(\theta, \tilde{\theta}, X_1, \cdots, X_n)$ donnée dans la précédente proposition, nous obtenons ainsi l'expression suivante du Lagrangien

$$\begin{aligned} L(\theta, \lambda) = & -\frac{n}{2} log(2\pi) + \sum_{j=1}^{J} \left(\sum_{i=1}^{n} \mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i) \right) log(\alpha_j) \\ & - \frac{1}{2} \sum_{j=1}^{J} \left(\sum_{i=1}^{n} \mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i) \left(log(v_j) + \frac{(X_i - \mu_j)^2}{v_j} \right) \right) - \lambda \left(\sum_{i=1}^{n} \alpha_i - 1 \right) \end{aligned}$$

Le Lagrangien admet un maximum sous la contrainte et ce maximum θ^*vérifie le système suivant :

$$\begin{cases} \dfrac{\partial \mathscr{L}}{\partial \alpha_j}(\theta^*) = \dfrac{\displaystyle\sum_{i=1}^{n} \mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i)}{v_j} - \lambda & = 0 \ \forall j \in [\![1, J]\!] \\ \dfrac{\partial \mathscr{L}}{\partial \mu_j}(\theta^*) = \displaystyle\sum_{i=1}^{n} \mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i)(-2X_i + 2\mu_j) & = 0 \ \forall j \in [\![1, J]\!] \\ \dfrac{\partial \mathscr{L}}{\partial v_j}(\theta^*) = -\dfrac{1}{2v_j} \displaystyle\sum_{i=1}^{n} \mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i) + \dfrac{1}{2v_j^2} \displaystyle\sum_{i=1}^{n} \mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i)(X_i - \mu_j)^2 & = 0 \ \forall j \in [\![1, J]\!] \\ \dfrac{\partial \mathscr{L}}{\partial \lambda}(\theta^*) = \displaystyle\sum_{i=1}^{n} \alpha_i - 1 & = 0 \end{cases}$$

Sous $\tilde{\theta}$ fixé, et ce qui est bien le cas, on a $\mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i)$ une constante. Le système devient alors :

$$\begin{cases} \alpha_j = \dfrac{\displaystyle\sum_{i=1}^{n} g_{\tilde{\theta}}(j|X = X_i)}{\lambda} & \forall j \in [\![1, J]\!] \\ \mu_j = \dfrac{\displaystyle\sum_{i=1}^{n} X_i \mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i)}{\displaystyle\sum_{i=1}^{n} \mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i)} & \forall j \in [\![1, J]\!] \\ v_j = \dfrac{\displaystyle\sum_{i=1}^{n} (X_i - \mu_j)^2 \mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i)}{\displaystyle\sum_{i=1}^{n} \mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i)} & \forall j \in [\![1, J]\!] \\ \displaystyle\sum_{i=1}^{J} \alpha_i = 1 \end{cases}$$

Sous la contrainte $\sum_{i=1}^{n} \alpha_i = 1$ et la première équation du système précedent on obtient l'égalité suivante :

$$\begin{aligned} \sum_{i=1}^{J} \alpha_i &= \sum_{i=1}^{J} \left(\frac{\displaystyle\sum_{i=1}^{n} \mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i)}{\lambda} \right) \\ &= \frac{\displaystyle\sum_{i=1}^{J} \sum_{i=1}^{n} \mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i)}{\lambda} \\ &= \frac{\displaystyle\sum_{i=1}^{n} \sum_{i=1}^{J} \mathbb{P}_{\tilde{\theta}}(Z = j|X = X_i)}{\lambda} \\ &= \frac{\displaystyle\sum_{i=1}^{n} 1}{\lambda} = 1 \end{aligned}$$

On en déduit ainsi $\lambda = n$, ainsi que le résultat énoncé. ☙

Tous ces inesthétiques et fastidieux calculs n'ont pas été effectué en vain. Nous les avons réalisé suite à l'introduction d'une notion nouvelle, celle de la log-vraisemblance conditionnelle ; qui elle même à été introduite faute de ne pouvoir obtenir une expression analytique de la log-vraisemblance des observations. Nous allons maintenant tâcher de mettre en exergue le rapport entre ces deux log-vraisemblances.

Chapitre 2

L'algorithme EM

Dans le présent chapitre, nous nous intéresserons A REMPLIR

2.1 Présentation laconique et pseudo-code

présentation textuelle rapide de l'algo : etape E, etape M etc...

Pour l'implémentation de cet algorithme, nous nous sommes appuyés sur le pseudo-code suivant.

Algorithm 1 L'algorithme EM (Dempster et al., 1977).

Entrée(s) : $N \in \mathbb{N}$, $\widehat{\theta}_0 \in \Theta$, un jeu de données $x_1 \dots x_n$;
Initialisation ;
$k := 1$;
Tant que $K < N + 1$ **faire**
  ETAPE E : *Calculer la fonction* $Q(\theta; \widehat{\theta}_{k-1}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\widehat{\theta}_{k-1}}[log f(X_i, Z_i, \theta) | X_i = x_i]$;
  ETAPE M : $\widehat{\theta}_k = argmax \; Q(\theta; \widehat{\theta}_{k-1})$;
  $k \leftarrow k + 1$;
fin du Tant que ;
retourner $\widehat{\theta}_N$;

Il n'existe pas de preuve de convergence de l'algorithme EM ; ce dernier peut en effet stagner dans des extremas locaux. Le choix de bons paramètres initiaux est de fait primordial. Nous verrons cela dans une prochaine section. Toutefois, nous sommes assurés que l'algorithme croît, en temoigne la section suivante.

2.2 Une preuve de la croissance

Dans cette concise partie, nous donnons une preuve de la croissance de la log-vraisemblance conditionnelle au fur et à mesure des itérations de l'algorithme EM.

Théorème 1. *Soit $(\theta_k)_{k\in\mathbb{N}}$ la suite de paramètres construite à l'aide de l'algorithme EM. La log-vraisemblance $\mathscr{L}_{obs}$ des observations vérifie*

$$\mathscr{L}_{obs}(\theta_{k+1}, X_1, \cdots, X_n) \geq \mathscr{L}_{obs}(\theta_k, X_1, \cdots, X_n)$$

Démonstration. Nous allons commencer cette preuve en donnant une autre forme de la log-vraisemblance, dépendant de $\mathscr{L}_{obs}(\theta, X_1, \cdots, X_n)$ et d'un terme κ_{θ,θ_k}. Nous avons :

$$\begin{aligned}
\mathscr{L}_c(\theta, \theta_k, X_1, \cdots, X_n) &= \sum_{i=1}^{n}\sum_{j=1}^{J} ln(h_\theta(X_i, j))\mathbb{P}_{\theta_k}(Z = j|X = X_i) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{J} ln\big[f_\theta(X_i) \times \mathbb{P}_\theta(Z = j|X = X_i)\big]\mathbb{P}_{\theta_k}(Z = j|X = X_i) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{J} ln(f_\theta(X_i))\mathbb{P}_{\theta_k}(Z = j|X = X_i) + \sum_{i=1}^{n}\sum_{j=1}^{J} ln(\mathbb{P}_\theta(Z = j|X = X_i))\mathbb{P}_{\theta_k}(Z = j|X = X_i) \\
&= \sum_{i=1}^{n} ln(f_\theta(X_i)) \times \underbrace{\sum_{j=1}^{J} \mathbb{P}_{\theta_k}(Z = j|X = X_i)}_{=1} + \sum_{i=1}^{n}\sum_{j=1}^{J} ln(\mathbb{P}_\theta(Z = j|X = X_i))\mathbb{P}_{\theta_k}(Z = j|X = X_i) \\
&= \sum_{i=1}^{n} ln(f_\theta(X_i)) + \sum_{i=1}^{n}\sum_{j=1}^{J} ln(\mathbb{P}_\theta(Z = j|X = X_i))\mathbb{P}_{\theta_k}(Z = j|X = X_i) \\
&= \mathscr{L}_{obs}(\theta, X_1, \cdots, X_n) + \kappa_{\theta,\theta_k}
\end{aligned}$$

Dès lors, on obtient

$$\mathscr{L}_{obs}(\theta_{k+1}, X_1, \cdots, X_n) - \mathscr{L}_{obs}(\theta_k, X_1, \cdots, X_n) = \mathscr{L}_c(\theta_{k+1}, \theta_k, X_1, \cdots, X_n) - \kappa_{\theta_{k+1},\theta_k} - \mathscr{L}_c(\theta_k, \theta_k, X_1, \cdots, X_n) + \kappa_{\theta_k,\theta_k}$$

Or, la quantité $\mathscr{L}_c$ est maximisée en θ_{k+1} lors de l'étape M de l'algorithme, donc

$$\mathscr{L}_c(\theta_{k+1}, \theta_k, X_1, \cdots, X_n) - \mathscr{L}_c(\theta_k, \theta_k, X_1, \cdots, X_n) \geq 0$$

Il reste donc à prouver que

$$\kappa_{\theta_k,\theta_k} - \kappa_{\theta_{k+1},\theta_k} \geq 0$$

En effet, nous avons

$$\begin{aligned}
\kappa_{\theta_k,\theta_k} - \kappa_{\theta_{k+1},\theta_k} &= \sum_{i=1}^{n}\sum_{j=1}^{J} ln(\mathbb{P}_{\theta_k}(Z=j|X=X_i))\mathbb{P}_{\theta_k}(Z=j|X=X_i) \\
&- \sum_{i=1}^{n}\sum_{j=1}^{J} ln(\mathbb{P}_{\theta_{k+1}}(Z=j|X=X_i))\mathbb{P}_{\theta_k}(Z=j|X=X_i) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{J} ln\left(\frac{\mathbb{P}_{\theta_k}(Z=j|X=X_i)}{\mathbb{P}_{\theta_{k+1}}(Z=j|X=X_i)}\right)\mathbb{P}_{\theta_k}(Z=j|X=X_i) \\
&= -n\sum_{i=1}^{n}\sum_{j=1}^{J} ln\left(\frac{\mathbb{P}_{\theta_{k+1}}(Z=j|X=X_i)}{\mathbb{P}_{\theta_k}(Z=j|X=X_i)}\right)\mathbb{P}_{\theta_k}(Z=j|X=X_i)\times\frac{1}{n} \\
&\geq -n\times ln\left(\sum_{i=1}^{n}\sum_{j=1}^{J} \frac{\mathbb{P}_{\theta_{k+1}}(Z=j|X=X_i)}{\mathbb{P}_{\theta_k}(Z=j|X=X_i)}\mathbb{P}_{\theta_k}(Z=j|X=X_i)\times\frac{1}{n}\right) \\
&\left[\text{Cette dernière inégalité est due à la convexité de } ln \text{ et au fait que } \sum_{i=1}^{n}\sum_{j=1}^{J}\mathbb{P}_{\theta_k}(Z=j|X=X_i)\times\frac{1}{n}=1\right] \\
&= -n\times ln\left(\sum_{i=1}^{n}\sum_{j=1}^{J}\mathbb{P}_{\theta_{k+1}}(Z=j|X=X_i)\times\frac{1}{n}\right) \\
&= -n\times ln(1) \\
&= 0
\end{aligned}$$

On obtient ainsi

$$\kappa_{\theta_k,\theta_k} - \kappa_{\theta_{k+1},\theta_k} \geq 0$$

Et finalement

$$\mathscr{L}_{obs}(\theta_{k+1}, X_1, \cdots, X_n) \geq \mathscr{L}_{obs}(\theta_k, X_1, \cdots, X_n)$$

❧

Bibliographie

[1] Doc de Brunel ? ? ?
[2] https://members.loria.fr/moberger/Enseignement/AVR/Exposes/algo-em.pdf
[3] http://faculty.washington.edu/fxia/courses/LING572/EM_collins97.pdf
[4] https://core.ac.uk/download/pdf/155777956.pdf

Annexes

Annexe A

Le package *mclust*

Nous avons, dans un élan d'audace, commencé par programmer à la main l'algorithme EM, en nous appuyant sur le pseudo-code explicité en première partie du chapitre II.

Cependant, il existe une librairie *R* - la librairie *mclust* - contenant une implémentation de l'algorithme EM. Notre algorithme étant fonctionnel, nous ne détaillerons pas ici le fonctionnement de ce Package. Il est néanmoins pertinent de l'expérimenter, voire de comparer ces résultats avec ceux notre algorithme.

Pour commencer, installons et chargeons le Package *mclust*.

```
install.packages("mclust")
library("mclust")
```

Nous reprenons les données des nids d'oiseaux :

```
bird_names = c("European Goldfinch", "Common Linnet", "Common Chaffinch",
               "European Greenfinch", "Eurasian Bullfinch", "Hawfinch",
               "Stonechat", "European Robin", "Whinchat", "Song Thrush",
               "Common Blackbird", "Ring Ouzel", "Mistle Thrush")

mean_volume = c(38.0, 60.9, 58.3, 74.5, 45.0, 71.6, 91.0, 68.4, 51.9, 288.9,
                293.6, 298.6, 266.1)

sd_volume = c(9.1,  20.8, 15.0,  12.2,  3.8,  12.9,  46.5, 29.8, 27.4, 55.9,
              78.5,  125.1,  56.6)
```

Puis, il suffit de construire des dataframe. Ici, nous considérerons deux mélanges ; un mélange à deux lois et un autre à trois lois.

```
df_2= data.frame(bird_names = c("European Goldfinch", "Ring Ouzel")
                          ,proportion_alpha = c(0.3, 0.7), mean = c(38, 298.6),
                          sd = c(9.1, 125.1))

df_3= data.frame(bird_names = c("Common Linnet", "Common Chaffinch", "Hawfinch" )
                          ,proportion_alpha = c(0.6, 0.3, 0.1), mean = c(60.9, 58.3,
    71.6),
                          sd = c(20.8, 15.0, 12.9))
```

Nous reprenons ici notre propre fonction de simulation

```
simulation = function(data_th, n=100)
```

Les prérequis étant posés, nous simulons un échantillon $X2$ de deux espèces d'oiseaux et un autre $X3$ de trois espèces d'oiseaux :

```
set.seed(1907)
X2 <- simulation(df_2)
X3 <- simulation(df_3)
```

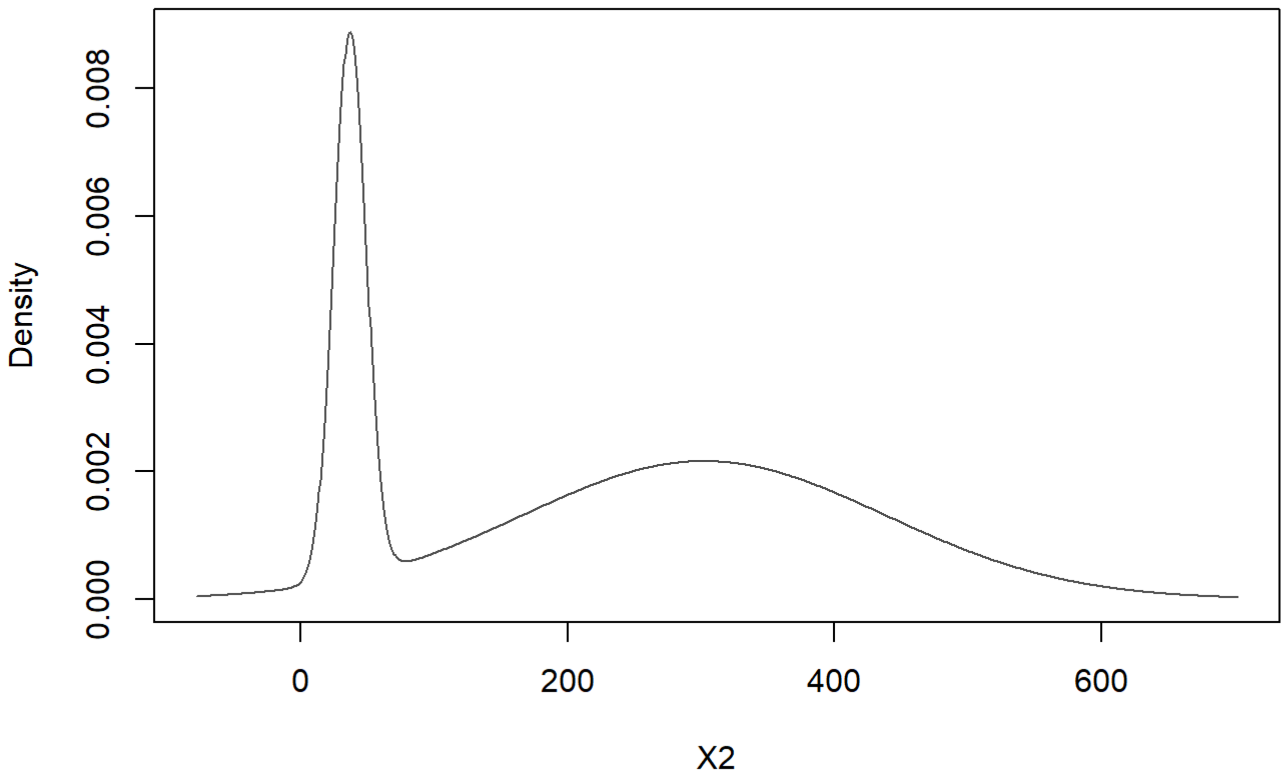
Le Package *mclust* est des plus complet ; les possibilités étant très vastes et hors du cadre de ce projet (notamment les fonctionnalités de clustering), nous regarderons uniquement la fonction qui nous intéresse, à savoir la fonction *densityMclust*.
Cette dernière prend en argument des fonctionnalités pertinentes, comme le nombre de mélanges, mais ne permet pas de régler manuellement des valeurs initiales pour les paramètres à estimer.

A.1 Un exemple sur un mélange à deux lois

Commençons par un executer la fonction *densityMclust* sur notre exemple de mélange à deux lois, contenu dans le dataframe $X2$:

```
est_2 <- densityMclust(X2)
```

Il est en premier lieu retourné le graphe de la densité du mélange de lois.



Nous pouvons nettement distinguer les deux "pics", correspondant aux deux gaussiennes mélangées. Intéressons-nous maintenant à l'objet créé *est_2*.

```
est_2
```

```
'densityMclust' model object: (V,2)
```

```
Available components:
 [1] "call"           "data"           "modelName"      "n"              "d"
 [6] "G"              "BIC"            "loglik"         "df"             "bic"
[11] "icl"            "hypvol"         "parameters"     "z"              "classification"
[16] "uncertainty"    "density"
```

Ici nous voulons les paramètres estimés, nous nous concentrerons donc que sur la treizième coordonnée de ce vecteur.
Rappelons que les divers paramètres de ce mélange sont : 0.3 et 0.7 en proportions ; 38 et 298.6 pour les moyennes ; et 9.1 et 125.1 en écart-types.

```
print("Proportions estimées:")
est_2[13]$parameters$pro
print("Moyennes estimées:")
est_2[13]$parameters$mean
print("Ecart-types estimés:")
(est_2[13]$parameters$variance$sigmasq)^(1/2)
```

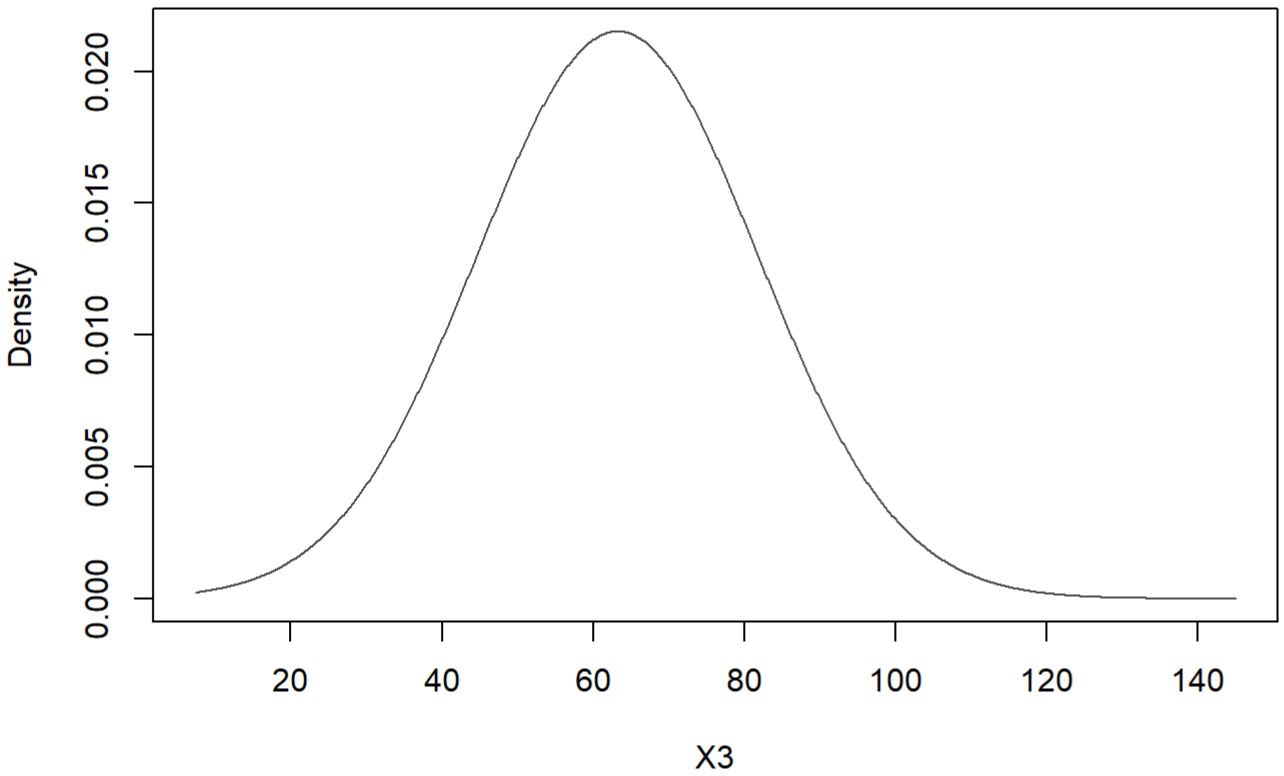
```
[1] "Proportions estimées:"
[1] 0.2612243 0.7387757
[1] "Moyennes estimées:"
        1         2
 36.80898 301.97108
[1] "Ecart-types estimés:"
[1]  12.16256 136.32256
```

Ici, le nombre de mélange est exact. Les proportions sont très bien estimées, l'erreur la plus importante est de l'ordre de 12%. Il en va de même pour les moyennes, les erreurs sont d'ordres inférieures à 10%. Les variances sont elles aussi très bien estimées, les erreurs sont négligeables.

A.2 Un exemple sur un mélange à trois lois

Regardons maintenant le cas d'un mélange de trois lois. Afin de mettre à rude épreuve l'algorithme, nous allons choisir les espèces telles que les moyennes et variances soient proches. Les proportions seront quant à elles bien distinctes, nous allons voir pourquoi.

```
est_3<- densityMclust(X3)
```



Nous obtenons ici quelque chose d'intéressant ; la fonction de densité de ce mélange de trois lois paraît toute à fait gaussienne. Sans une exploration plus approfondie des données, nous commettrions une chagrinante erreur et des conclusions totalement faussées...
Il est ici pertinent d'observer la structure des données ; plus précisemment, nous allons effectuer un test de *Shapiro*.

```
shapiro.test(X3)
```

```
Shapiro-Wilk normality test

data:  X3
W = 0.97982, p-value = 0.1287
```

La p-value est de 0.1287, ce qui est certes peu élevée, mais pas assez pour rejeter l'hypothèse (H_0) de normalité. Nous sommes ici dans une situation ambiguë.

Observons maintenant comment *densityMclust* se défend fasse à cette situation.
Rappelons que les divers paramètres de ce mélange sont : 0.6, 0.3 et 0.1 en proportions ; 60.9, 58.3 et 71.6 en moyenne ; et 20.8, 15 et 12.9 en écart-types.

```
print("Proportions estimées:")
est_3[13]$parameters$pro
print("Moyennes estimées:")
```

```
est_3[13]$parameters$mean
print("Ecart-types estimés:")
(est_3[13]$parameters$variance$sigmasq)^(1/2)
```

```
[1] "Proportions estimées:"
[1] 1
[1] "Moyennes estimées:"
[1] 63.20547
[1] "Ecart-types estimés:"
[1] 18.54033
```

Le premier élément notable est que l'algorithme échoue à établir le nombre correct de lois. L'unique moyenne et écart-type estimés ne sont quant à eux pas absurde.
Nous allons relancer la fonction sur le même jeu de données, en précisant cette fois-ci le nombre de lois.

```
est_3b <- densityMclust(X3, G = 3)
print("Proportions estimées:")
est_3b[13]$parameters$pro
print("Moyennes estimées:")
est_3b[13]$parameters$mean
print("Ecart-types estimés:")
(est_3b[13]$parameters$variance$sigmasq)^(1/2)
```

0.6, 0.3, 0.1 60.9, 58.3, 71.6 20.8, 15, 12.9

```
[1] "Proportions estimées:"
[1] 0.2552686 0.5642272 0.1805042
[1] "Moyennes estimées:"
       1        2        3
56.63179 62.36268 75.13635
[1] "Ecart-types estimés:"
[1] 17.51051
```

Les proportions sont plutôt bien estimées, quoique légèrement surestimées pour deux d'entres elles, mais les erreurs restent faibles. Il en est étonnament de même pour les moyennes, qui sont très bien estimées. Ceci est surprenant au vue de l'allure de la densité. Cependant, il n'est estimé qu'un unique écart-type, ce qui n'est guère étonnant. Notons que celle ci est à peu près égale à la moyenne des écart-types des différentes lois.

Ce cas ambigüe met en exergue les limites de l'algorithme implémenté dans ce package.