Projet de Statistiques

Application de l'Algorithme EM à un mélange de Gaussiennes

Alexandre FILIOT
Sofiane FENIZA

Introduction

Le but de ce projet est d'étudier l'algorithme EM et de l'implémenter sur un jeu de données pour un modèle de mélange de gaussiennes.

Nous considérons un ensemble de n variables aléatoires $(X_i)_{i=1,...,n}$ à valeurs dans $\mathbb{R}^p$ indépendantes et identiquement distribuées (i.i.d) générées selon un mélange de K distributions distinctes de densités par rapport à la mesure de Lebesgue connues $(f_{\theta_k})_{k=1,...,K}$ où $\theta = (\theta_1, ..., \theta_K)$ est un paramètre inconnu. Chaque X_i est généré selon une loi caractérisée par f_{θ_k}, mais nous n'observons pas le k correspondant. Nous pouvons alors introduire la variable latente Z_i valant k si X_i est généré selon f_{θ_k}. La densité de la loi suivie par les variables générées par ce modèle peut s'écrire :

$$f(x) = \sum_{k=1}^{K} \alpha_k f_{\theta_k}(x) \qquad \forall\, x \in \mathbb{R}^p$$

où $\alpha = (\alpha_1, ..., \alpha_K)$ et $\theta = (\theta_1, ..., \theta_K)$ sont des paramètres inconnus, avec $\alpha_k \geq 0\ \forall\, k = 1, ..., K$ et $\sum_{k=1}^{K} \alpha_k = 1$.
Nous chercons à regrouper les observations qui ont été générées par la même densité.

1 Préliminaires

Question 1 : Justifier la forme de la densité du modèle de mélange en montrant le lien avec l'interprétation en termes de variables latentes.

Nous disposons de K classes $C_1, C_2, \ldots, C_K$. Une observation $X_i \in \mathbb{R}^p$ pour $1 \leq i \leq n$ appartient à la classe C_k si sa densité de probabilité est $f_{\theta_k}(x_i)$ avec θ_k un paramètre inconnu qu'il nous faudra estimer. Il paraît alors intuitif de pondérer l'appartenance de l'observation X_i aux différentes classes par les proportions du mélange. Ces proportions sont introduites dans l'énoncé par les paramètres α_k. Ces poids α_k vérifient bien les propriétés $\forall\, k \in [\![1; K]\!]\ \alpha_k \geq 0$ et $\sum_{k=1}^{K} \alpha_k = 1$.
Introduisons désormais des variables latentes $(Z_1, \ldots, Z_n)$ (définies comme dans l'énoncé), autrement dit des données cachées, à priori inconnues. Supposons que nous les connaissons. En écrivant les α_k comme des $\mathbb{P}(Z_i = k)$ (à i fixé), nous avons donc que si $\mathbb{P}(Z_i = k) = 1$ alors $X_i \sim f_{\theta_k}(x_i)$.

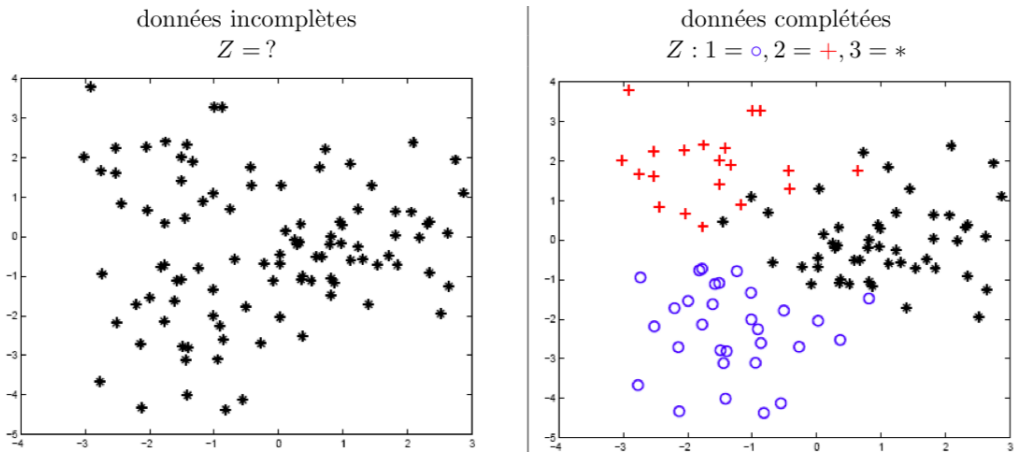

FIGURE 1 – *Influence de la connaissance des Z_i sur la classification des observations. Ici, $K = 3$.*

Nous pouvons écrire que si l'observation X_i appartient effectivement à la classe k :

$$X_i | Z_i = k \sim f_{\theta_k}(x_i)$$

Ceci nous permet d'écrire la distribution de X_i sous la forme :

$$\begin{aligned} f(x_i|\boldsymbol{\theta}, \boldsymbol{\alpha}) &= \mathbb{1}_{Z_i=1}\mathbb{P}(Z_i = 1) \times f(x_i|\theta_1) + \ldots + \mathbb{1}_{Z_i=K}\mathbb{P}(Z_i = K) \times f(x_i|\theta_K) \\ &= \sum_{k=1}^{K} \mathbb{1}_{Z_i=k}\mathbb{P}(Z_i = k) \times f(x_i|\theta_k) \\ &= \sum_{k=1}^{K} \mathbb{1}_{Z_i=k}\alpha_k f(x_i|\theta_k) \end{aligned}$$

Cette écriture est adaptée lorsque les Z_i sont connues. Or dans la pratique, nous ne disposons d'aucune information sur les variables latentes. En introduisant les Z_i inconnues, on passe d'un modèle à données manquantes à un modèle à données complètes (X_i, Z_i) ! La vraisemblance est donc différente (et c'est d'ailleurs la base de l'algorithme EM) :

$$f(x_i|\boldsymbol{\theta}, \boldsymbol{\alpha}) \to f(x_i, z_i|\boldsymbol{\theta}, \boldsymbol{\alpha})$$

Nous ne disposons pas d'informations quant aux α_k. Ainsi eux aussi deviennent des paramètres du modèle et il est impératif de prendre cette donnée en compte. On comprend finalement que l'objectif de l'algorithme EM sera donc double : à la fois d'estimer les paramètres du modèle $\boldsymbol{\theta}$ pour expliciter et distinguer les distributions entre elles ; et surtout estimer les proportions du mélange $\boldsymbol{\alpha}$ pour répondre à notre problème de classification !

Question 2 : Quelle est la différence entre des problèmes supervisés et non supervisés ? A quelle catégorie appartient celui que nous venons de présenter ? Quel est le nom exact de ce problème ?

Les problèmes de classification non supervisée ont pour objectif de regrouper en un certain nombre K de classes, un ensemble de n observations à partir de plusieurs caractéristiques. En classification non supervisée, l'appartenance des différentes observations à l'une des K populations n'est pas connue. Les méthodes de classification non supervisée visent donc à retrouver, à partir des caractéristiques relevées sur chaque observation, l'appartenance des observations à telle ou telle classe. En classification supervisée il s'agit du problème inverse : l'appartenance des données aux K classes est connue et doit permettre de construire une règle de classement (ou différentiation) des classes entre elles afin de prédire à quelle classe appartiendra une nouvelle donnée.
Au regard de ce qui vient d'être dit, le problème de ce sujet est un problème de classification non supervisée. Dans ce sujet (cf. partie 3) nous connaissons le nombre de classes K (en pratique ce n'est pas le cas) mais sommes incapables à priori de relier les observations à leurs classes d'appartenance respectives. Plus spécifiquement ici, notre problème est un problème de classification automatique paramétrique (non supervisée). La classification automatique paramétrique désigne toute une catégorie d'algorithme consistant à attribuer une classe à chaque objet/observation, en se basant sur des données statistiques (à la différence des modèles de regroupement hierarchique qui introduisent une mesure de dissimilarité entre les objets). L'algorithme EM, comme nous allons le voir, a pour application majeure l'estimation des paramètres d'un modèle de mélange, en particulier les modèles de mélange gaussiens.

2 Mélange de gaussiennes et algorithme EM

Nous allons désormais nous attarder sur l'algorithme EM et son application à un mélange de gaussiennes. Nous considérons à partir de maintenant un mélange de gaussiennes, c'est-à-dire que les paramètres θ_k pour $k = 1, ..., K$ sont sous la forme $(\mu_k, \Sigma_k) \in \mathbb{R}^p \prod \mathbb{R}^p \prod p$ et que les densités s'écrivent : ;

$$f_{(\mu_k,\Sigma_k)}(x) = \frac{1}{(2\pi)^{p/2}\det(\Sigma)^{1/2}}\exp\left(-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\right) \qquad \forall\, x \in \mathbb{R}^p$$

Nous nous plaçons dans les cas où p = 1 et p=2. Traiter ces deux cas dans chacune des questions de cette partie.

Question 3 : Expliciter la vraisemblance du modèle.

De part le caractère i.i.d des X_i, nous pouvons écrire la vraisemblance comme suit :

$$\mathcal{L}(x_1, ..., x_n|\boldsymbol{\phi}) = \prod_{i=1}^{n} f(x_i|\boldsymbol{\phi}) = \prod_{i=1}^{n}\sum_{k=1}^{K}\alpha_k f_{\theta_k}(x_i)$$

Et donc la log-vraisemblance suivante :

$$\ell(x_1, ..., x_n|\boldsymbol{\phi}) = \sum_{i=1}^{n}\ln\left\{\sum_{k=1}^{K}\alpha_k f_{\theta_k}(x_i)\right\} \qquad (*)$$

avec $\boldsymbol{\phi} = (\boldsymbol{\theta}, \boldsymbol{\alpha})$.

Dans le cas univarié ($p = 1$), il suffit d'écrire que pour tout $k \in [\![1; K]\!]$:

$$f_{(\mu_k,\sigma_k^2)}(x_i) = \frac{1}{\sqrt{2\pi}\sigma_k}\exp\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma_k^2}\right) \;\forall\, i \in [\![1; n]\!]$$

et dans le cas général mutlivarié (p>1) ; pour tout k, $\theta_k = (\mu_k, \Sigma_k) \in \mathbb{R}^p \times \mathbb{R}^{p\times p}$:

$$f_{(\mu_k,\Sigma_k)}(x_i) = \frac{1}{(2\pi)^{p/2}\det(\Sigma)^{1/2}}\exp\left(-\frac{1}{2}(x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k)\right) \;\forall x_i \in \mathbb{R}^p \;,\; i \in [\![1; n]\!]$$

et remplacer dans l'expression $(*)$ pour conclure.

Question 4 : Pourquoi l'estimateur du maximum de vraisemblance n'est-il pas adapté ? Expliquer l'intérêt de l'algorithme EM dans le cas présent.

Pour rappel, la méthode du maximum de vraisemblance vise à annuler la dérivée par rapport aux paramètres θ_k et α_k de la log-vraisemblance précédemment explicitée. Le problème est que la forme complexe de cette dernière rend impossible le calcul explicite des expressions des estimateurs de maximum de vraisemblance $\hat{\theta}_k$ et $\hat{\alpha}_k$... (ceci est lié au logarithme de la somme). Ainsi la méthode du maximum de vraisemblance n'est analytiquement pas réalisable dans le cadre du modèle à données manquantes. C'est là que l'algorithme EM intervient. Nous l'avons dit, le modèle de mélange peut être mis en relation avec des variables latentes Z_i pour ainsi passer d'un modèle à données manquantes à un modèle à données complètes. Ainsi la distribution globale des (X_i, Z_i) est la suivante :

$$f(x_i, z_i|\boldsymbol{\phi}) = \sum_{k=1}^{K} \mathbb{1}_{Z_i=k} \alpha_k f(x_i|\theta_k)$$

En utilisant le fait que :

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{Z}|\phi) = \mathcal{L}(\boldsymbol{X}|\boldsymbol{Z}, \phi) \times \mathcal{L}(\boldsymbol{Z}|\phi)$$

on obtient alors :

$$\begin{aligned}
\mathcal{L}(\mathbf{X}, \mathbf{Z}|\theta) &= \prod_{i=1}^{n} f(x_i, z_i|\theta) \\
&= \prod_{i=1}^{n} \sum_{k=1}^{K} \mathbb{1}_{Z_i=k} f(x_i|Z_i = k, \theta) \mathbb{P}(Z_i = k|\theta) \\
&= \prod_{i=1}^{n} \sum_{k=1}^{K} \mathbb{1}_{Z_i=k} \alpha_k f_{\theta_k}(x_i)
\end{aligned}$$

Ceci peut se réécrire afin de faire apparaître un double produit :

$$\mathcal{L}(\mathbf{X}, \mathbf{Z}|\theta) = \prod_{i=1}^{n} \prod_{k=1}^{K} \left(\alpha_k f_{\theta_k}(x_i)\right)^{\mathbb{1}_{Z_i=k}}$$

Le calcul de la log-vraisemblance dans le modèle à données complètes s'en déduit facilement :

$$\ell(\mathbf{X}, \mathbf{Z}|\theta) = \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{1}_{Z_i=k} \left\{\ln(\alpha_k) + \ln\left(f_{(\theta_k)}(x_i)\right)\right\}$$

avec $\theta_k = (\mu_k, \sigma_k^2) \in \mathbb{R} \times \mathbb{R}^+$ dans le cas où $p = 1$ et $\theta_k = (\mu_k, \Sigma_k) \in \mathbb{R}^p \times \mathbb{R}^{p \times p}$ dans le cas mutlivarié $p > 1$.

Or la maximisation de l'espérance de cette quantité est non-seulement analytiquement réalisable, mais en plus, elle permet d'atteindre l'objectif initial. À savoir, maximiser la log-vraisemblance des données manquantes !

Dans la suite de ce sujet nous noterons $\boldsymbol{\theta} = (\boldsymbol{\alpha_1}, ..., \boldsymbol{\alpha_K}, \boldsymbol{\mu_1}, ..., \boldsymbol{\mu_K}, \boldsymbol{\Sigma_1}, ..., \boldsymbol{\Sigma_K})$.
De plus, les observations $\mathbf{X_i}$ sont des vecteurs colonnes.

Question 5 : Présenter l'algorithme EM : expliciter le principe de la méthode et l'accompagner des notations correspondantes afin d'illustrer la procédure, en particulier pour la vraisemblance complète et les E- et M-steps.

L'algorithme EM - E pour Expectation, M pour Maximization - est un algorithme itératif du à Dempster, Laird et Rubin (1977). L'algorithme EM est utilisé lorsque les seules données dont on dispose (ici les X_i) ne permettent pas l'estimation des paramètres inconnus du modèle car l'expression de la vraisemblance est analytiquement impossible à maximiser. L'introduction de données "cachées" ou manquantes, que l'on appelle variables aléatoires latentes (ici les Z_i), permet de lever cette difficulté. En effet, la connaissance de ces données rendrait possible l'estimation des paramètres du modèle. L'algorithme EM tire son nom du fait qu'à chaque itération il opère deux étapes distinctes :

— la phase « Expectation » (« étape E ») procède à l'estimation des variables latentes sachant les données observées et les valeurs des paramètres déterminées à l'itération précédente ;
— la phase « Maximisation » (« étape M ») procède ensuite à la maximisation de la vraisemblance. Ceci est rendu possible par l'estimation des données inconnues (étape E) ! L'étape M met donc à jour la valeur des paramètres du modèle pour la prochaine itération.

L'algorithme EM permet donc de faire sauter l'obstacle rendant impossible l'application de la méthode du maximum de vraisemblance (MV). De plus, chaque itération de l'algorithme augmente la vraisemblance du modèle à données manquantes !

Formalisons ce qui précède au travers d'une itération $t \to t+1$.

— Nous disposons d'observation i.i.d $\mathbf{X} = (X_1, ..., X_n)$ avec $X_i \in \mathbb{R}^p,\ p \geq 1$, de vraisemblance notée $\mathcal{L}(\mathbf{X}|\theta)$. Comme nous l'avons vu précédemment, maximiser la log-vraisemblance du modèle à données manquantes, $\ln(\mathcal{L}(\mathbf{X}|\theta))$, n'est pas analytiquement possible.
— On considère alors des variables latentes (dites données cachées) $\mathbf{Z} = (Z_1, ..., Z_n)$ avec $Z_i \in [\![1; K]\!]$ dont la connaissance rend possible la maximisation de la log-vraisemblance du modèle à données complètes, $\ln(\mathcal{L}(\mathbf{X}, \mathbf{Z}|\theta))$.
— Les données $\mathbf{Z}$ étant inconnues, on estime la log-vraisemblance du modèle à données complètes en calculant la quantité $\mathcal{Q}^{(t)}\left(\theta|\tilde{\theta}\right) = \mathbb{E}_{\mathbf{Z}|\mathbf{X}=x,\theta^{(t)}}\left[\ln(\mathcal{L}(x, \mathbf{Z}|\tilde{\theta}))\right]$ (étape E). $\theta^{(t)}$ désigne l'ensemble des paramètres courants estimés à l'étape précédente t.
— On maximise enfin $\mathcal{Q}^{(t)}\left(\theta|\tilde{\theta}\right)$ par rapport aux paramètres $\tilde{\theta}$ afin de déterminer les nouvelles valeurs des paramètres $\theta^{(t+1)}$ (étape M).

Ainsi une itération vise à calculer :

$$\boxed{\theta^{(t+1)} = \text{argmax}_{\tilde{\theta}}\left[\mathcal{Q}^{(t)}\left(\theta|\tilde{\theta}\right)\right]}$$

Le double avantage de l'algorithme EM est qu'il améliore conjointement, et ce à chaque itération, l'estimation des données manquantes (les $\mathbb{P}(Z_i = k) = \alpha_k$) et celle des paramètres du modèles.

Cas $\mathbf{p = 1}$:

Dans cette section, nous nous plaçons dans le cas particulier d'un mélange de gaussiennes univariées. Ainsi, pour tout $k \in [\![1; K]\!]$:

$$f_{(\mu_k, \sigma_k^2)}(x_i) = \frac{1}{\sqrt{2\pi}\sigma_k}\exp\left(-\frac{1}{2}\frac{(x_i - \mu_k)^2}{\sigma_k^2}\right)\ \forall\, i \in [\![1; n]\!]$$

Comme détaillé dans la question précédente, la log-vraisemblance du modèle à valeurs complètes s'écrit :

$$\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{1}_{Z_i=k}\left\{\ln(\alpha_k) + \ln\left(f_{(\theta_k)}(x_i)\right)\right\}$$

Dans le cas univarié, on obtient :

$$\ell(\mathbf{X}, \mathbf{Z}|\theta) = \sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{1}_{Z_i=k}\left\{\ln(\alpha_k) - \frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln(\sigma_k^2) - \frac{1}{2}\frac{(x_i - \mu_k)^2}{\sigma_k^2}\right\}$$

Maintenant que la log-vraisemblance du modèle à données complètes est explicitée, nous pouvons passer à l'implémentation des différentes étapes de l'algorithme EM.

Initialisation

La première étape de l'algorithme consiste à initialiser les paramètres :

$$\theta^{(0)} = \left(\alpha_1^{(0)}, ..., \alpha_K^{(0)}, \mu_1^{(0)}, ..., \mu_K^{(0)}, \sigma_1^{2(0)}, ..., \sigma_K^{2(0)}\right)$$

Cette initialisation s'effectue souvent grâce à l'algorithme K-means (classification initiale des X_i en K clusters qui permet de déduire les proportions initiales du mélange $(\alpha_1^{(0)}, ..., \alpha_K^{(0)})$) puis l'application des formules d'espérance et variance empiriques pour estimer $(\mu_1^{(0)}, ..., \mu_K^{(0)}, \sigma_1^{2(0)}, ..., \sigma_K^{2(0)})$. Nous y reviendrons en temps voulu.

À chaque itération t

— Étape E : On cherche à calculer $\mathcal{Q}^{(t)}\left(\theta|\tilde{\theta}\right) = \mathbb{E}_{\mathbf{Z}|\mathbf{X}=x,\theta^{(t)}}\left[\ln(\mathcal{L}(x, \mathbf{Z}|\tilde{\theta}))\right]$

Par linéarité de l'espérance, on obtient que :

$$\mathcal{Q}^{(t)}\left(\theta|\tilde{\theta}\right) = \sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{E}\left[\mathbb{1}_{Z_i=k}|X_i = x_i, \theta^{(t)}\right]\left\{\ln(\tilde{\alpha}_k) - \frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln(\tilde{\sigma}_k^2) - \frac{1}{2}\frac{(x_i - \tilde{\mu}_k)^2}{\tilde{\sigma}_k^2}\right\}$$

Nous pouvons alors introduire les probabilités d'appartenance de l'observation X_i à la classe C_k : $p_{ik}^{(t)} = p_{ik}(\theta^{(t)})$! Ceci permet également de simplifier l'écriture.

$$p_{ik}^{(t)} = \mathbb{E}\left[\mathbb{1}_{Z_i=k}|X_i = x_i, \theta^{(t)}\right] = \mathbb{P}\left(Z_i = k|X_i = x_i, \theta^{(t)}\right)$$

Appliquons maintenant la formule de Bayes pour expliciter les $p_{ik}^{(t)}$.

$$p_{ik}^{(t)} = \frac{\mathbb{P}\left(Z_i = k|X_i = x_i, \theta^{(t)}\right)}{\mathbb{P}\left(X_i = x_i|\theta^{(t)}\right)} = \frac{\alpha_k^{(t)} f_{(\mu_k^{(t)}, \sigma_k^{2(t)})}(x_i)}{\sum_{l=1}^{K}\alpha_l^{(t)} f_{(\mu_l^{(t)}, \sigma_l^{2(t)})}(x_i)} \qquad (1)$$

Finalement :

$$\mathcal{Q}^{(t)}\left(\theta|\tilde{\theta}\right) = \sum_{i=1}^{n}\sum_{k=1}^{K}p_{ik}^{(t)}\left\{\ln(\tilde{\alpha}_k) - \frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln(\tilde{\sigma}_k^2) - \frac{1}{2}\frac{(x_i - \tilde{\mu}_k)^2}{\tilde{\sigma}_k^2}\right\}$$

Ces probabilités $p_{ik}^{(t)}$ sont des probabilités à posteriori et dépendent donc des paramètres du modèle (α_k, μ_k et σ_k^2). Ainsi l'estimation de ces paramètres permet de calculer les $p_{ik}^{(t)}$. Cette estimation se fait à l'étape M.

— Étape M : On cherche désormais à maximiser en $\tilde{\theta}$ la quantité $\mathcal{Q}^{(t)}\left(\theta|\tilde{\theta}\right)$. Cette maximisation s'opère par rapport aux trois catégories de paramètres, les $\tilde{\alpha_k}$, $\tilde{\mu_k}$ et $\tilde{\sigma}_k^2$ pour tout k. Commençons par la maximisation en $\tilde{\alpha}$.

- **Maximisation en $(\tilde{\boldsymbol{\alpha}}_1, \tilde{\boldsymbol{\alpha}}_2, ..., , \tilde{\boldsymbol{\alpha}}_K)$**

Il s'agit ici d'un problème d'optimisation sous la contrainte $\sum_{k=1}^{K}\tilde{\alpha}_k = 1$. On utilise donc le Lagrangien

$$\mathcal{L}(\tilde{\alpha}_1, ..., \tilde{\alpha}_k; \lambda) = \mathcal{Q}^{(t)}\left(\theta|\tilde{\theta}\right) + \lambda\left(\sum_{k=1}^{K}\tilde{\alpha}_k - 1\right)$$

Pour tout k fixé $\in [\![1; K]\!]$, on a donc :

$$\frac{\partial\mathcal{L}(\tilde{\alpha}_1, ..., \tilde{\alpha}_k; \lambda)}{\partial\tilde{\alpha}_k} = \sum_{i=1}^{n}\frac{p_{ik}^{(t)}}{\tilde{\alpha}_k} + \lambda = 0 \implies -\lambda\tilde{\alpha}_k = \sum_{i=1}^{n}p_{ik}^{(t)} \qquad (*)$$

Ainsi en sommant les k égalités $(*)$ et en intervertissant les deux sommes (ce qui est permis par finitude), on obtient :

$$-\lambda\sum_{k=1}^{K}\tilde{\alpha}_k = \sum_{i=1}^{n}\sum_{k=1}^{K}p_{ik}^{(t)}$$

Seulement, $\sum_{k=1}^{K}\tilde{\alpha}_k = 1$ par définition et en reprenant l'expression des $p_{ik}^{(t)}$ (équation (1)), il est aussi immédiat que $\sum_{k=1}^{K}p_{ik}^{(t)} = 1$. On en déduit donc que $-\lambda = n$. En reprenant l'équation $(*)$, nous pouvons donc conclure que, $\forall k \in [\![1; K]\!]$:

$$\boxed{\alpha_k^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n}p_{ik}^{(t)}}$$

- **Maximisation en $(\tilde{\mu}_1, \tilde{\mu}_2, ..., , \tilde{\mu}_K)$**

Pour tout $k \in [\![1; K]\!]$, on a :

$$\frac{\partial \mathcal{Q}^{(t)}\left(\theta|\tilde{\theta}\right)}{\partial \tilde{\mu}_k} = 0 \implies -\frac{1}{\tilde{\sigma}_k^2}\sum_{i=1}^{n}p_{ik}^{(t)}(x_i - \tilde{\mu}_k) = 0$$

Ce qui mène de manière immédiate à :

$$\boxed{\mu_k^{(t+1)} = \frac{\sum_{i=1}^{n}p_{ik}^{(t)}x_i}{\sum_{i=1}^{n}p_{ik}^{(t)}}}$$

- **Maximisation en $(\tilde{\sigma}_1^2, \tilde{\sigma}_2^2, \dots \tilde{\sigma}_K^2)$**

Pour tout $k \in [\![1; K]\!]$, on a :

$$\frac{\partial \mathcal{Q}^{(t)}\left(\theta|\tilde{\theta}\right)}{\partial \tilde{\sigma}_k^2} = 0 \implies \sum_{i=1}^{n}p_{ik}^{(t)}\left\{-\frac{1}{2\tilde{\sigma}_k^2} + \frac{(x_i - \tilde{\mu}_k)^2}{2\left(\tilde{\sigma_k}^2\right)^2}\right\} = 0$$

Ceci implique :

$$\frac{1}{\tilde{\sigma}_k^2}\sum_{i=1}^{n}p_{ik}^{(t)}(x_i - \tilde{\mu}_k)^2 = \sum_{i=1}^{n}p_{ik}^{(t)}$$

Finalement pour tout $k \in [\![1; K]\!]$, on a :

$$\boxed{\sigma_k^{2(t+1)} = \frac{\sum_{i=1}^{n}p_{ik}^{(t)}(x_i - \mu_k^{(t+1)})^2}{\sum_{i=1}^{n}p_{ik}^{(t)}}}$$

Ainsi, le passage de l'étape t à l'étape $t+1$ se résume par la mise à jour des paramètres de la manière suivante :

$$\forall k \in [\![1;K]\!] : \begin{cases} \alpha_k^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} p_{ik}^{(t)} \\ \mu_k^{(t+1)} = \dfrac{\sum_{i=1}^{n} p_{ik}^{(t)} x_i}{\sum_{i=1}^{n} p_{ik}^{(t)}} \\ \sigma_k^{2(t+1)} = \dfrac{\sum_{i=1}^{n} p_{ik}^{(t)} (x_i - \mu_k^{(t+1)})^2}{\sum_{i=1}^{n} p_{ik}^{(t)}} \end{cases} \quad \text{avec } p_{ik}^{(t)} = \frac{\frac{\alpha_k^{(t)}}{\sqrt{2\pi}\sigma_k^{(t)}} \exp\left(-\frac{1}{2}\frac{\left(x_i - \mu_k^{(t)}\right)^2}{\sigma_k^{2(t)}}\right)}{\sum_{l=1}^{K} \frac{\alpha_l^{(t)}}{\sqrt{2\pi}\sigma_l^{(t)}} \exp\left(-\frac{1}{2}\frac{\left(x_i - \mu_l^{(t)}\right)^2}{\sigma_l^{2(t)}}\right)}$$

À la dernière itération considérée, admettons $t = T$, le résultat de l'algorithme est donc

$$\theta^{(K)} = \left(\alpha_1^{(T)}, ..., \alpha_K^{(T)}, \mu_1^{(T)}, ..., \mu_K^{(T)}, \sigma_1^{2(T)}, ..., \sigma_K^{2(T)}\right)$$

Ainsi, l'utilisation de variables latentes (ou données cachées) permet d'introduire la loi jointe des variables $(\mathbf{X}, \mathbf{Z})$. En plus de résoudre l'obstacle dans la méthode d'estimation MV, cette technique permet également de prouver qu'à chaque itération, l'algorithme EM augmente la vraisemblance de la loi marginale, $\mathcal{L}(\mathbf{X}|\theta)$. On peut ainsi montrer que pour tout $t \leq T$:

$$\Delta\left(\theta^{(t+1)}, \theta^{(t)}\right) := \ell(\mathbf{X}|\theta^{(t+1)}) - \ell(\mathbf{X}|\theta^{(t)}) \geq 0$$

Cas général p > 1 :

Dans le cas de distributions gaussiennes multivariées, la méthode reste également la même. Cependant, les calculs permettant d'exhiber les nouveaux paramètres à l'étape $t+1$ en fonction de ceux à l'étape t se compliquent. Reprenons la structure de la partie consacrée au cas univariée.

Tout d'abord, la densité des X_i s'écrit, pour tout k, $(\mu_k, \Sigma_k) \in \mathbb{R}^p \times \mathbb{R}^{p\times p}$:

$$f_{(\mu_k,\Sigma_k)}(x_i) = \frac{1}{(2\pi)^{p/2}\det(\Sigma)^{1/2}} \exp\left(-\frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1}(x_i - \mu_k)\right) \ \forall x_i \in \mathbb{R}^p \ , \ i \in [\![1;n]\!]$$

On en déduit alors par le même raisonnement la log-vraisemblance du modèle à données complètes :

$$\ell(\mathbf{X}, \mathbf{Z}|\theta) = \sum_{i=1}^{n}\sum_{k=1}^{K} \mathbb{1}_{Z_i=k} \left\{\ln(\alpha_k) - \frac{p}{2}\ln(2\pi) - \frac{1}{2}\ln(\det(\Sigma_k)) - \frac{1}{2}\left((x_i - \mu_k)^T \Sigma_k^{-1}(x_i - \mu_k)\right)\right\}$$

À chaque itération t

— Étape E : On cherche à calculer $\mathcal{Q}^{(t)}\left(\theta|\tilde{\theta}\right) = \mathbb{E}_{\mathbf{Z}|\mathbf{X}=x,\theta^{(t)}}\left[\ln(\mathcal{L}(x, \mathbf{Z}|\tilde{\theta}))\right]$

Par linéarité de l'espérance, on obtient que :

$$\mathcal{Q}^{(t)}\left(\theta|\tilde{\theta}\right) = \sum_{i=1}^{n}\sum_{k=1}^{K} p_{ik}^{(t)} \left\{\ln(\tilde{\alpha}_k) - \frac{p}{2}\ln(2\pi) - \frac{1}{2}\ln(\det(\tilde{\Sigma}_k)) - \frac{1}{2}\left((x_i - \tilde{\mu}_k)^T \tilde{\Sigma}_k^{-1}(x_i - \tilde{\mu}_k)\right)\right\}$$

avec

$$p_{ik}^{(t)} := \frac{\alpha_k^{(t)} f_{(\mu_k^{(t)}, \Sigma_k^{(t)})}(x_i)}{\sum_{l=1}^{K} \alpha_l^{(t)} f_{(\mu_l^{(t)}, \Sigma_l^{(t)})}(x_i)} \qquad (2)$$

— Étape M : On cherche désormais à maximiser en $\tilde{\theta}$ la quantité $\mathcal{Q}^{(t)}\left(\theta|\tilde{\theta}\right)$. Cette maximisation s'opère par rapport aux trois catégories de paramètres, les $\tilde{\alpha_k}$, $\tilde{\mu_k}$ et $\tilde{\Sigma}_k$ pour tout k. La maximisation en $\tilde{\alpha}$ est exactement la même que dans le cas univarié (la dimension $p > 1$ n'a pas d'influence sur la dérivation par rapport aux $\tilde{\alpha}_k$). Seule l'expression des $p_{ik}^{(t)}$ change selon l'équation (2).

- **Maximisation en $(\tilde{\boldsymbol{\alpha}}_1, \tilde{\boldsymbol{\alpha}}_2, ..., , \tilde{\boldsymbol{\alpha}}_K)$**

On retrouve que $\forall\ k \in [\![1; K]\!]$:

$$\boxed{\alpha_k^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} p_{ik}^{(t)}}$$

- **Maximisation en $(\tilde{\boldsymbol{\mu}}_1, \tilde{\boldsymbol{\mu}}_2, ..., , \tilde{\boldsymbol{\mu}}_K)$**

Pour tout $k \in [\![1; K]\!]$, on a :

$$\frac{\partial \mathcal{Q}^{(t)}\left(\theta|\tilde{\theta}\right)}{\partial \tilde{\mu}_k} = 0 \implies -1 \times \sum_{i=1}^{n} p_{ik}^{(t)} \tilde{\Sigma}_k^{-1}(x_i - \tilde{\mu}_k) = 0$$

En multipliant cette inégalité par $\tilde{\Sigma}_k$, on obtient que

$$\sum_{i=1}^{n} p_{ik}^{(t)}(x_i - \tilde{\mu}_k) = 0$$

ce qui mène finalement au "même" résultat que dans le cas univarié, à savoir :

$$\boxed{\mu_k^{(t+1)} = \frac{\sum_{i=1}^{n} p_{ik}^{(t)} x_i}{\sum_{i=1}^{n} p_{ik}^{(t)}}}$$

- **Maximisation en $(\tilde{\boldsymbol{\Sigma}}_1, \tilde{\boldsymbol{\Sigma}}_2, \ldots \tilde{\boldsymbol{\Sigma}}_K)$**

La maximisation de $\mathcal{Q}^{(t)}\left(\theta|\tilde{\theta}\right)$ par rapport aux $\tilde{\Sigma}_k$ est plus complexe que dans le cas univarié. Nous avons donc préféré utiliser la méthode utilisée par Jeff A.Bilmes ([1], pages 5 à 7). Il en découle que :

$$\boxed{\Sigma_k^{(t+1)} = \frac{\sum_{i=1}^{n} p_{ik}^{(t)}(x_i - \mu_k^{(t+1)})(x_i - \mu_k^{(t+1)})^T}{\sum_{i=1}^{n} p_{ik}^{(t)}}}$$

Ainsi dans le cas de distributions gaussiennes multivariées, le passage de l'étape t à l'étape $t+1$ se résume par la mise à jour des paramètres de la manière suivante :

$$\forall k \in [\![1;K]\!] : \begin{cases} \alpha_k^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} p_{ik}^{(t)} \\ \mu_k^{(t+1)} = \frac{\sum_{i=1}^{n} p_{ik}^{(t)} x_i}{\sum_{i=1}^{n} p_{ik}^{(t)}} \\ \Sigma_k^{(t+1)} = \frac{\sum_{i=1}^{n} p_{ik}^{(t)} (x_i - \mu_k^{(t+1)})(x_i - \mu_k^{(t+1)})^T}{\sum_{i=1}^{n} p_{ik}^{(t)}} \end{cases},$$

avec

$$\text{avec } p_{ik}^{(t)} = \frac{\frac{\alpha_k^{(t)}}{(2\pi)^{p/2}\det\left(\Sigma_k^{(t)}\right)^{1/2}} \exp\left(-\frac{1}{2}(x_i - \mu_k^{(t)})^T \Sigma_k^{(t)-1}(x_i - \mu_k^{(t)})\right)}{\sum_{l=1}^{K} \frac{\alpha_l^{(t)}}{(2\pi)^{p/2}\det\left(\Sigma_l^{(t)}\right)^{1/2}} \exp\left(-\frac{1}{2}(x_i - \mu_l^{(t)})^T \Sigma_l^{(t)-1}(x_i - \mu_l^{(t)})\right)}$$

Une question que nous pourrions nous poser serait de savoir à quel moment arrêter l'algorithme ? Pour cela, on peut utiliser soit un critère d'arrêt temporel (nombre maximum d'itérations), soit utiliser un critère de précision. À savoir, lorsque, d'une itération à l'autre, la différence :

$$\Delta\left(\theta^{(t+1)}, \theta^{(t)}\right) := \ell(\mathbf{X}|\theta^{(t+1)}) - \ell(\mathbf{X}|\theta^{(t)})$$

passe sous un certain seuil ϵ.

Question 6 : Il est courant d'initialiser l'algorithme EM avec un algorithme K-Means. Présenter simplement les grandes lignes de ce dernier.

L'algorithme K-means est un algorithme de partitionnement de données utilisé principalement pour des problèmes de classification non-supervisée, ce qui est notre cas ici. Étant donné la connaissance de données $(X_i)_{1 \leqslant i \leqslant n} \in (\mathbb{R}^p)^n$ avec p la dimension des données ($p = 1, 2$ ici) ; et d'un entier K, l'algorithme K-means permet de diviser les observations en K groupes, dits clusters, de façon à minimiser un critère appelé *inertie* ou *variance intra-classe*. Ce critère n'est autre que la somme, pour tous les clusters $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_K$, des carrés des distances des points à leurs moyennes respectives (barycentres) :

$$I_W = \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_K} d^2(x_i, \mu_k) = \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_K} \|x_i - \mu_k\|^2$$

L'avantage du K-means est qu'il reste l'un des plus simples algorithmes de classification automatique des données. L'idée principale et de choisir aléatoirement un ensemble de barycentres fixés a priori (les μ_k) et de chercher itérativement la partition optimale qui minimise I_W :

$$\mathcal{C}_K^* = \underset{\mathcal{C} \in \mathcal{C}_K}{\operatorname{argmin}}\, I_W$$

Voici ci-dessous le principe de l'algorithme écrit en pseudo-code :

Data: Observations $(X_i)_{1\leqslant i\leqslant n} \in (\mathbb{R}^p)^n$ et nombre K de clusters souhaités.
Result: Partition optimale des clusters $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_K$.
Initialisation aléatoire des baycentres μ_k;
while *Critère d'arrêt non-vérifié* **do**

1 Affectation de chaque observation au cluster le plus proche (partition de Voronoï) :

$$X_i \in \mathcal{C}_k \text{ si } \forall\, l \in [\![1; K]\!]\ , \|x_i - \mu_k\| \leqslant \|x_i - \mu_l\|$$

2 Mise à jour des barycentres $\forall\, l \in [\![1; K]\!]$:

$$\mu_k = \frac{1}{|\mathcal{C}_k|} \sum_{i \in \mathcal{C}_k} x_i$$

end

Algorithm 1: Algorithme K-means

Il existe différents critères d'arrêts pour l'algorithme K-means : un nombre maximal d'itérations à ne pas dépasser ou un critère de précision. Ce critère est satisfait (i.e l'algorithme est dit convergeant) si d'une itération à l'autre, l'étape d'affectation des observations aux différents clusters ne modifie pas la composition des groupes.

Question 7 : Les deux algorithmes précédents convergent-ils nécessairement ? Si oui, a-t-on l'assurance d'avoir un optimum global ?

Les algorithmes K-means et EM convergent nécessairement. Cependant, la convergence vers un optimum global n'est pas systématique. En effet ces deux algorithmes sont très sensibles aux valeurs d'initialisation des paramètres. L'algorithme EM peut parfois converger vers un point-selle / maximum local de la vraisemblance. Pour certaines "mauvaises" valeurs d'initialisation, l'algorithme peut rester "bloqué" alors qu'il convergerait vers le maximum global pour d'autres valeurs initiales plus proches de la réalité. Il faut donc réaliser plusieurs initialisations différentes ! Quant à l'algorithme K-means, celui-ci converge systématiquement vers un minimum local de la fonction d'inertie, ce qui pose encore une fois le problème de l'initialisation. De plus la sensibilité de l'algorithme à l'initialisation est d'autant plus grande que la dimension des observations est grande (i.e p). Enfin, il faut noter que dans la majeure partie des cas, les algorithmes K-means et EM convergent relativement rapidement. Néanmoins pour certaines valeurs d'initialisation, il est possible que le critère d'arrêt de précision du K-means (introduit à la question précédente) soit vérifié au bout d'un temps très long. En effet l'algorithme K-means ne converge pas en général en un temps de calcul polynomial. Dans ce cas il est préférable d'imposer à la fois le critère d'arrêt et le critère du maximum d'itérations.

Initialisation de l'algorithme EM

Une fois que l'algorithme K-means a permis de donner une première classification des observations, la deuxième étape de l'initialisation consiste à estimer les différents paramètres du modèle. Pour cela nous utilisons une approche purement empirique. $\forall\, k \in [\![1; K]\!]$, on a :

$$\alpha_k^{(0)} = \frac{|\mathcal{C}_k|}{n} \qquad\qquad \mu_k^{(0)} = \frac{1}{|\mathcal{C}_k|} \sum_{i \in \mathcal{C}_k} x_i$$

$$\Sigma_k^{(0)} = \frac{1}{|\mathcal{C}_k|} \sum_{i \in \mathcal{C}_k} (x_i - \mu_k^{(0)})(x_i - \mu_k^{(0)})^T \text{ dans le cas général}$$

$$\sigma_k^{2(0)} = \frac{1}{|\mathcal{C}_k|} \sum_{i \in \mathcal{C}_k} (x_i - \mu_k^{(0)})^T (x_i - \mu_k^{(0)}) \text{ dans le cas où } \Sigma_k = \sigma_k^2 I_2$$

avec x_i vecteur colonne $\in \mathbb{R}^2$, et $|\mathcal{C}_k|$ =card$\{i \ /\ x_i \in \mathcal{C}_k\}$.

Implémentation de l'algorithme EM

Nous avons choisi de programmer les différentes questions et algorithmes sous le langage R, ici $p = 1$ et $K = 4$. Le dataset est dans le repo.

Question 8 : Implémenter l'algorithme K-Means décrit question 6 et l'appliquer au jeu de données. Joindre un graphique présentant les données, les clusters ainsi que leurs différents centroïdes (utiliser différentes couleurs pour les différents clusters). Essayer plusieurs fois l'algorithme avec plusieurs initialisations aléatoires et comparer les résultats. Commenter.

Voici ci-dessous (figure 2) un résultat obtenu par l'algorithme K-means appliqué à notre jeu de données - les points en gras correspondent aux centroïdes des clusters.

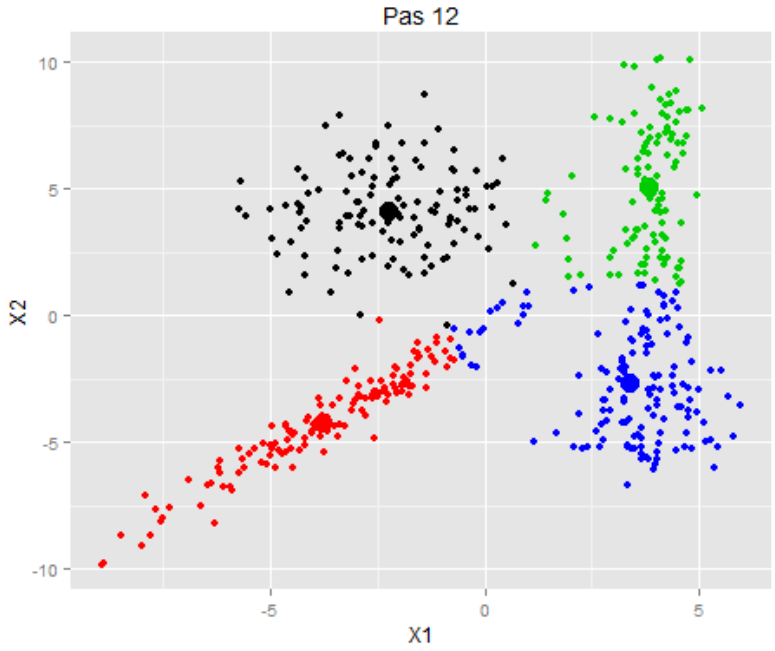

FIGURE 2 – *Exécution de l'algorithme K-means*

On distingue donc bien 4 clusters. L'algorithme s'est arrêté à la 12^{ime} itération puisque qu'aucun cluster n'a été modifié (il s'agit du critère de précision mentionné plus haut). Nous allons maintenant apporter une preuve plus tangible de la sensibilité de la convergence de l'algorithme vis-à-vis de l'initialisation. Cette initialisation se fait par la commande suivante :

```
> Indices_centres = sample.int(nrow(data), size = K)
> centres = (data[I_centres, ])
```

Cette initialisation est plutôt naïve (comparé à K-means ++) puisqu'elle sélectionne aléatoirement les 4 centres parmi les données. Nous avons affiché page 12 les 6 étapes de l'algorithme K-means jusqu'à convergence ; autrement dit, une convergence rapide.

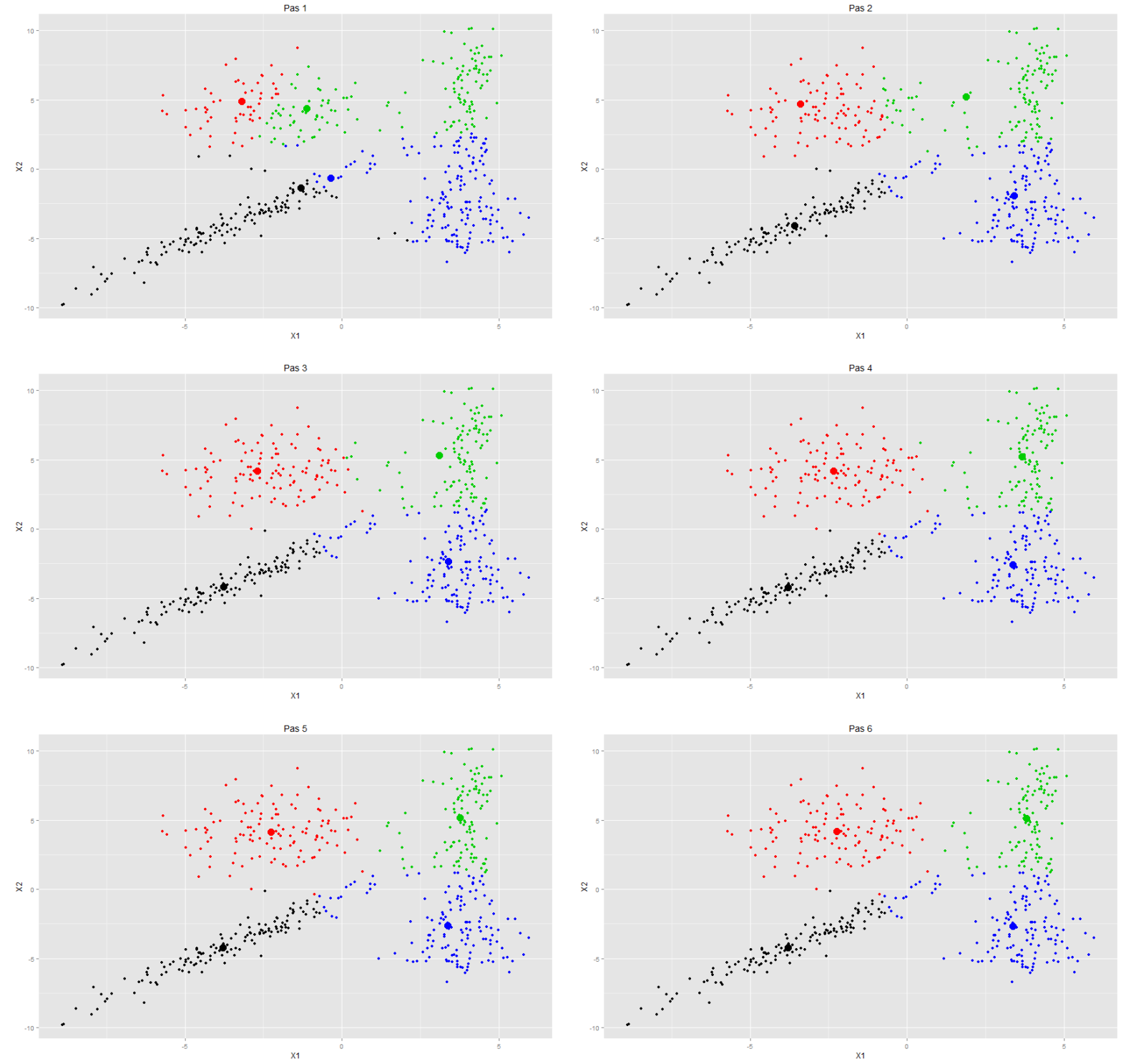

FIGURE 3 – *Convergence de l'algorithme K-means en 6 étapes seulement*

Maintenant, pour une autre initialisation, nous obtenons une convergence tout à fait différente puisqu'elle a nécessité 24 itérations. Cette différence s'explique en s'appuyant sur le choix des centres à la première étape (initialisation). Pour le cas ci-dessus, la désignation aléatoire des centres était plutôt favorable à une convergence rapide : centres éloignés et proches en moyenne des résultats finaux. Dans l'exemple ci-dessous, ces conditions ne sont pas vérifiées. En effet, les centres rouge, bleu et vert sont très proches et donc éloignés de leur position finale. On voit d'ailleurs que la répartition (au pas 1) des clusters est très différente comparé au cas précédent.

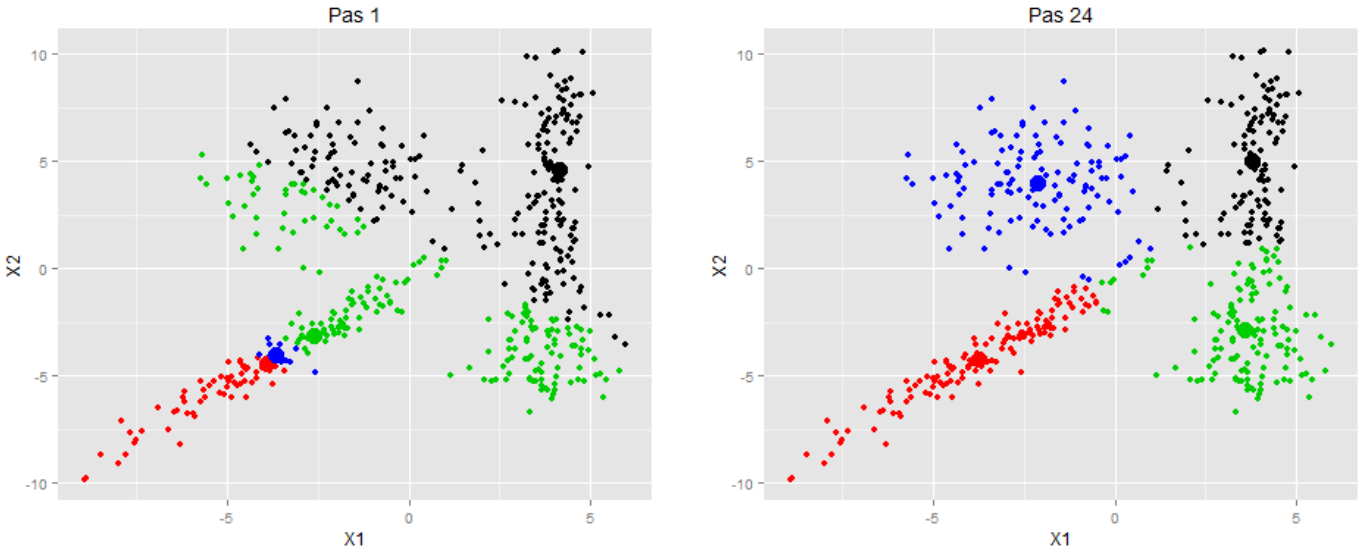

FIGURE 4 – *Convergence de l'algorithme K-means en 24 étapes*

Au-delà de ces 2 exemples, nous avons souhaité tracer la distribution du nombre de pas nécessaires à la convergence de l'algorithme K-means, ce pour un grand nombre d'initialisations différentes (ici, 5000).

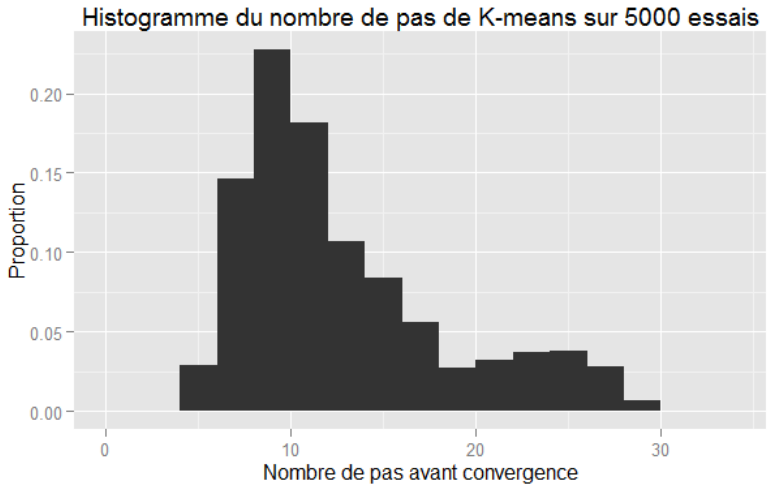

FIGURE 5 – *Distribution du nombre de pas nécessaires à la convergence de l'algorithme K-means*

Dans environ 25% des cas, l'algorithme K-means convergeait en 10 pas maximum. Dans plus de 75% des cas, en moins de 20 pas. En revanche, il existe bien des initialisations pour lesquelles ce nombre se situe entre 20 et 30, soit 2 à 3 fois plus que la valeur moyenne (autour de 10). Cette queue de distribution illustre dans quelle mesure l'algorithme est sensible aux différentes initialisations aléatoires de ses 4 centres.

Question 9 : Nous considérons dans cette question des gaussiennes dont les matrices de covariance sont proportionnelles à l'identité, i.e. telles que $\Sigma_k = \sigma_k^2 I_2$ où I_2 est la matrice identité de taille (2,2) et σ_k^2 un réel strictement positif. Réécrire la forme de l'équation associée à la M-Step. Implémenter l'algorithme EM correspondant en initialisant à l'aide d'un K-Means et l'appliquer au jeu de données. Joindre à nouveau un graphique présentant les données, les moyennes des différentes gaussiennes et les clusters. Expliquer comment ces clusters sont obtenus.

Pour rappel, l'étape E vise à calculer la quantité $\mathcal{Q}^{(t)}\left(\theta|\tilde{\theta}\right) = \mathbb{E}_{\mathbf{Z}|\mathbf{X}=x,\theta^{(t)}}\left[\ln(\mathcal{L}(x,\mathbf{Z}|\tilde{\theta}))\right]$
Dans le cas des matrices de variance-covariance proportionnelles à l'identité, on a :

$$\Sigma_k = \sigma_k^2 I_2, \quad \det(\Sigma_k) = \left(\sigma_k^2\right)^2, \quad \text{et} \quad \Sigma_k^{-1} = \frac{1}{\sigma_k^2} I_2$$

Ainsi

$$\mathcal{Q}^{(t)}\left(\theta|\tilde{\theta}\right) = \sum_{i=1}^{n}\sum_{k=1}^{K} p_{ik}^{(t)} \left\{ \ln(\tilde{\alpha}_k) - \frac{p}{2}\ln(2\pi) - \frac{1}{2}\ln(\det(\tilde{\Sigma}_k)) - \frac{1}{2}\left((x_i-\tilde{\mu}_k)^T\tilde{\Sigma}_k^{-1}(x_i-\tilde{\mu}_k)\right)\right\}$$

se réecrit pour $p = 2$:

$$\mathcal{Q}^{(t)}\left(\theta|\tilde{\theta}\right) = \sum_{i=1}^{n}\sum_{k=1}^{K} p_{ik}^{(t)} \left\{ \ln(\tilde{\alpha}_k) - \ln(2\pi) - \ln(\tilde{\sigma}_k^2) - \frac{1}{2\tilde{\sigma}_k^2}\left((x_i-\tilde{\mu}_k)^T(x_i-\tilde{\mu}_k)\right)\right\}$$

avec

$$p_{ik}^{(t)} = \frac{\frac{\alpha_k^{(t)}}{2\pi\sigma_k^{2(t)}}\exp\left(-\frac{1}{2\sigma_k^{2(t)}}(x_i-\mu_k^{(t)})^T(x_i-\mu_k^{(t)})\right)}{\sum_{l=1}^{K}\frac{\alpha_l^{(t)}}{2\pi\sigma_l^{2(t)}}\exp\left(-\frac{1}{2\sigma_l^{2(t)}}(x_i-\mu_l^{(t)})^T(x_i-\mu_l^{(t)})\right)}$$

La maximisation en les paramètres $\tilde{\mu}_k$ et $\tilde{\alpha}_k$ est inchangée. En revanche, lorsque l'on regarde la nouvelle expression de $\mathcal{Q}^{(t)}\left(\theta|\tilde{\theta}\right)$, on voit bien que dans le cas particulier des matrices de variance-covariance proportionnelles à l'identité, la maximisation en $(\tilde{\Sigma}_1, \tilde{\Sigma}_2, \dots \tilde{\Sigma}_K)$ devient une maximisation en $\boldsymbol{(\tilde{\sigma}_1^2, \tilde{\sigma}_2^2, \dots \tilde{\sigma}_K^2)}$. Reprenons donc la méthode de maximisation de l'étape M pour ces paramètres.

- **Maximisation en $\boldsymbol{(\tilde{\sigma}_1^2, \tilde{\sigma}_2^2, \dots \tilde{\sigma}_K^2)}$**

Pour tout $k \in [\![1; K]\!]$, on a :

$$\frac{\partial\mathcal{Q}^{(t)}\left(\theta|\tilde{\theta}\right)}{\partial\tilde{\sigma}_k^2} = 0 \implies \sum_{i=1}^{n} p_{ik}^{(t)}\left\{-\frac{1}{\tilde{\sigma}_k^2} + \frac{\|x_i-\tilde{\mu}_k\|^2}{2\left(\tilde{\sigma_k}^2\right)^2}\right\} = 0$$

Ceci implique :

$$\frac{1}{2\tilde{\sigma}_k^2}\sum_{i=1}^{n} p_{ik}^{(t)}\|x_i-\tilde{\mu}_k\|^2 = \sum_{i=1}^{n} p_{ik}^{(t)}$$

Finalement pour tout $k \in [\![1; K]\!]$, on a :

$$\boxed{\sigma_k^{2(t+1)} = \frac{\sum_{i=1}^{n} p_{ik}^{(t)}(x_i-\tilde{\mu}_k)^T(x_i-\tilde{\mu}_k)}{2\sum_{i=1}^{n} p_{ik}^{(t)}}}$$

Ainsi dans le cas particulier des matrices de variance-covariance proportionnelles à l'identité, l'équation associée à l'étape M de l'algorithme EM s'écrit :

$$\forall k \in [\![1;K]\!] : \begin{cases} \alpha_k^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} p_{ik}^{(t)} \\ \mu_k^{(t+1)} = \frac{\sum_{i=1}^{n} p_{ik}^{(t)} x_i}{\sum_{i=1}^{n} p_{ik}^{(t)}}, \\ \sigma_k^{2(t+1)} = \frac{\sum_{i=1}^{n} p_{ik}^{(t)} (x_i - \tilde{\mu}_k)^T (x_i - \tilde{\mu}_k)}{2\sum_{i=1}^{n} p_{ik}^{(t)}} \end{cases}$$

avec

$$\text{avec } p_{ik}^{(t)} = \frac{\frac{\alpha_k^{(t)}}{2\pi\sigma_k^{2(t)}} \exp\left(-\frac{1}{2\sigma_k^{2(t)}}(x_i - \mu_k^{(t)})^T (x_i - \mu_k^{(t)})\right)}{\sum_{l=1}^{K} \frac{\alpha_l^{(t)}}{2\pi\sigma_l^{2(t)}} \exp\left(-\frac{1}{2\sigma_l^{2(t)}}(x_i - \mu_l^{(t)})^T (x_i - \mu_l^{(t)})\right)}$$

La différence avec la dimension 1 vient de l'apparition du facteur 2 au dénominateur de l'expression de $\sigma_k^{2(t+1)}$ (en plus du fait que l'on ne considère plus des observations scalaires mais bien des vecteurs de dimension 2). Une fois les $\sigma_k^{2(t+1)}$ estimés, on peut donc retrouver immédiatement les matrices $\Sigma_k^{(t+1)}$.

Voici donc le résultat graphique de l'exécution de l'algorithme EM dans le cas des matrices proportionnelles. Le seuil de précision (confère fin de la question 5 est fixé à $\epsilon = 10^{-3}$).

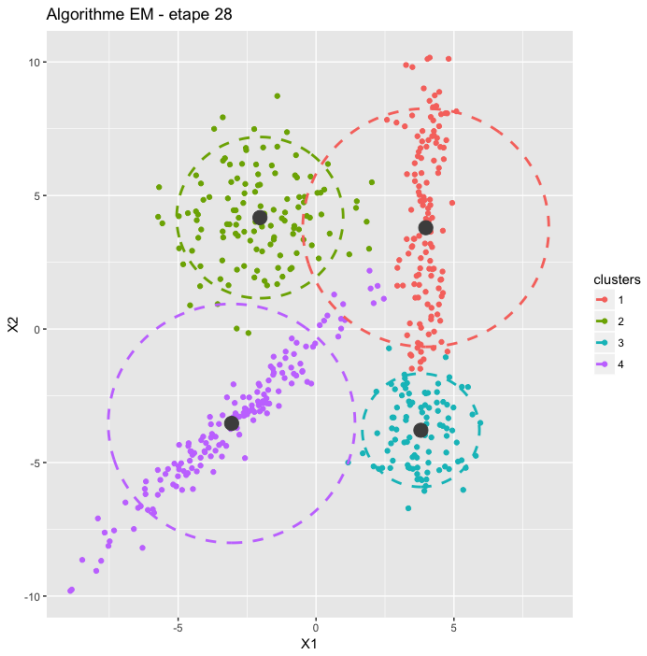

FIGURE 6 – *Convergence de l'algorithme EM en 28 étapes pour des matrices de variance-covariance proportionnelles à l'identité* - $\Delta\left(\theta^{(t+1)}, \theta^{(t)}\right) < 10^{-3}$

En prenant un peu d'avance sur le sujet, nous constatons que les ellipses contenant 80% de la masse de chaque gaussienne sont en réalité des cercles. Ceci est bien sûr lié à l'hypothèse selon laquelle les matrices de variance-covariance sont proportionnelles à l'identité.

Maximum À Posteriori

Comment avons nous obtenu les clusters précédents ? En utilisant les $p_{ik}^{(t)}$. Pour rappel, à i et k fixés, p_{ik} désigne la probabilité que l'observation X_i appartienne à la classe C_k. Une fois l'algorithme EM arrêté, leur estimation permet d'affecter chaque individu à une classe. Pour cela, on utilise la règle du Maximum À Posteriori (MAP) :

$$\boxed{X_i \in G_k \text{ si } p_{ik}^{(t)} > p_{il}^{(t)} \quad \forall\, l \neq k}$$

Ceci est d'autant plus efficace et pertinent lorsque ces probabilités sont très dispersées. Auquel cas la règle du MAP est plus significative.

Question 10 : Traiter la question précédente mais dans le cas général (où les matrices de covariances ne sont plus nécessairement proportionnelles à l'identité).

Pour rappel, dans le cas des matrices de variance-covariance non proportionnelles à l'identité, le passage d'une étape t à une étape $t+1$ dans l'algorithme EM se résume par la mise à jour des paramètres de la manière suivante :

$$\boxed{\forall k \in [\![1;K]\!] : \begin{cases} \alpha_k^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} p_{ik}^{(t)} \\ \mu_k^{(t+1)} = \dfrac{\sum_{i=1}^{n} p_{ik}^{(t)} x_i}{\sum_{i=1}^{n} p_{ik}^{(t)}}, \\ \Sigma_k^{(t+1)} = \dfrac{\sum_{i=1}^{n} p_{ik}^{(t)} (x_i - \mu_k^{(t+1)})(x_i - \mu_k^{(t+1)})^T}{\sum_{i=1}^{n} p_{ik}^{(t)}} \end{cases}}$$

avec

$$\boxed{\text{avec } p_{ik}^{(t)} = \frac{\frac{\alpha_k^{(t)}}{\det\left(\Sigma_k^{(t)}\right)^{1/2}} \exp\left(-\frac{1}{2}(x_i - \mu_k^{(t)})^T \Sigma_k^{(t)-1} (x_i - \mu_k^{(t)})\right)}{\sum_{l=1}^{K} \frac{\alpha_l^{(t)}}{\det\left(\Sigma_l^{(t)}\right)^{1/2}} \exp\left(-\frac{1}{2}(x_i - \mu_l^{(t)})^T \Sigma_l^{(t)-1} (x_i - \mu_l^{(t)})\right)}}$$

après simplification des termes en 2π.

Veuillez-donc trouver à la page 17 le résultat de l'algorithme EM dans le cas général bivarié.

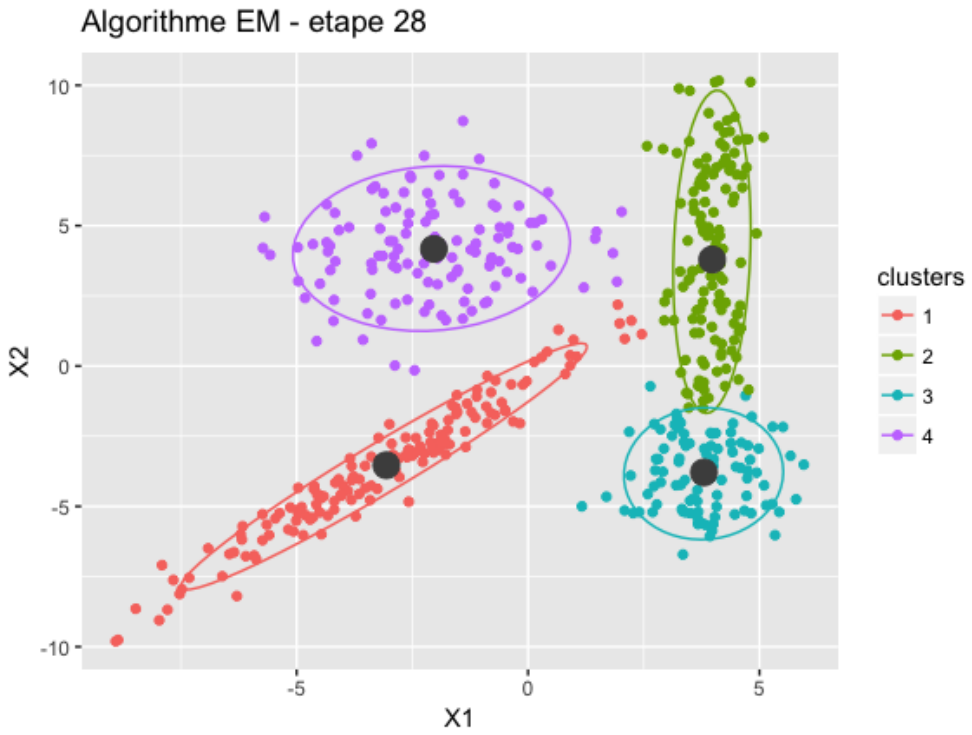

FIGURE 7 – *Convergence de l'algorithme EM en 28 étapes dans le cas général* - $\Delta < 10^{-3}$

Question 11 : Pour les deux questions précédentes, représenter sur le graphique présentant les données les matrices de covariance obtenues. Pour cela, tracer les ellipses qui contiennent 80% de la masse de chaque gaussienne. Comparer les ellipses pour le cas des matrices proportionnelles à l'identité et le cas général.

Avant de passer à la comparaison graphique, nous avons souhaité ajouter quelques éléments théoriques relatifs aux différentes formes des ellipses de covariance. Pour cela, nous nous sommes référés à la thèse d'Alexandre Lourme publiée en 2011 ([2], pages 25 à 26).

Toute matrice symétrique, définie et positive est diagonalisable dans une base orthonormée de vecteurs propres. En particulier, chaque matrice de variance-covariance $\mathbf{\Sigma}_k$ peut se décomposer de la sorte :

$$\mathbf{\Sigma}_k = \lambda_k \boldsymbol{S}_k \mathbf{\Lambda}_k \boldsymbol{S}_k^T$$

où $\lambda_k = |\Sigma_k|^{1/2}$ dans le cas bivarié, $\boldsymbol{S}_k$ une matrice orthogonale des vecteurs propres de $\mathbf{\Sigma}_k$, $\mathbf{\Lambda}_k$ est la matrice des valeurs propres de $\mathbf{\Sigma}_k$ divisées par λ_k (normalisation). Ces paramètres influent respectivement sur le volume (homothétie de rapport $\lambda_k^{1/2}$), l'orientation (rotation de matrice $\boldsymbol{S}_k$), la forme (via la matrice des valeurs propres) et la position (translation de vecteur $\boldsymbol{\mu}_k$) du cercle unité centré en 0 (cas de la dimension 2). Ainsi pour des matrices $\mathbf{\Sigma}_k$ proportionnelles à l'identité, $\lambda_k^{1/2} \times \boldsymbol{S}_k = \lambda_k^{1/2} \times \boldsymbol{S}_k^T = I_2$, avec $\lambda_k = \sigma_k^2$. Autrement dit, la seule transformation opérée sur le cercle unité est une transformation volumique via σ_k^2 et bien sûr une translation selon μ_k ; on conserve donc bien des cercles (déplacés dans le plan) dont le rayon est à chaque fois propotionnel à $\sigma_k^{2(t)}$. Dans le cas général, on obtient des ellipses.

Pour pouvoir comparer les ellipses aux cercles, nous avons choisi de les superposer. L'initialisation de l'algorithme est différente de celles réalisées dans les deux figures précédentes (5 et 6). Le seuil est toujours de $\epsilon = 10^{-3}$. Cercles comme ellipses contiennent bien 80% de la masse de chaque gaussienne.

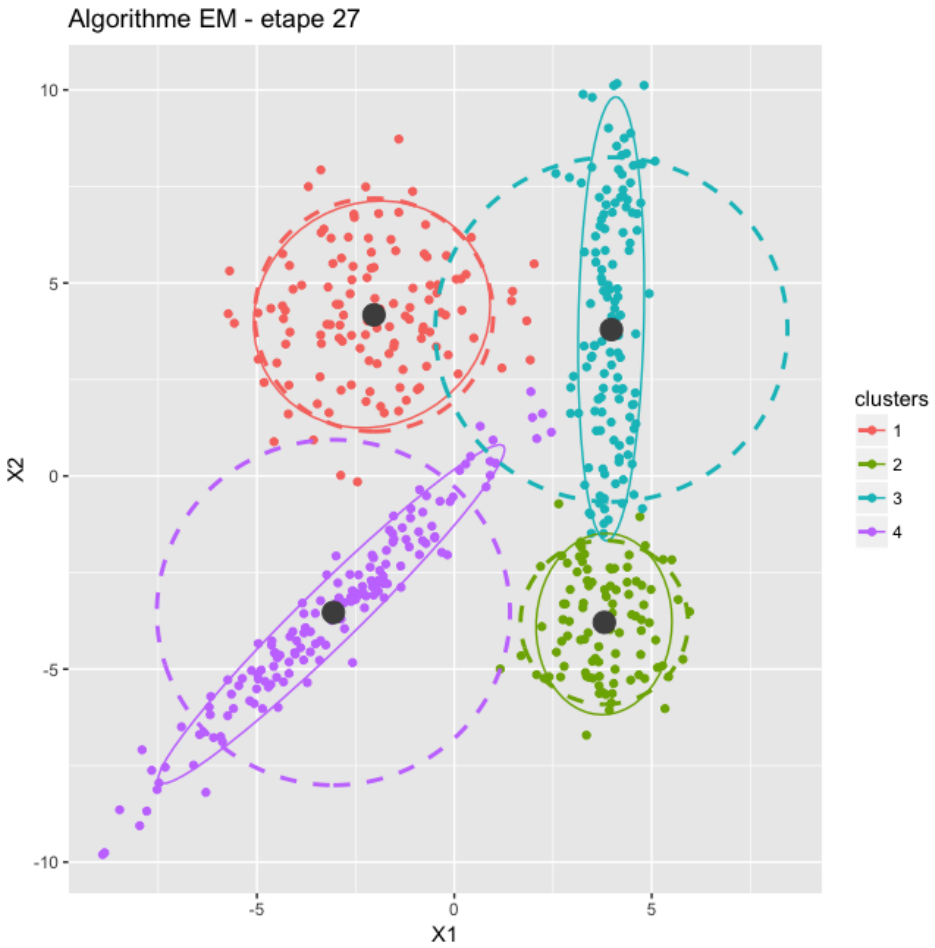

FIGURE 8 – *Comparaison des ellipses de covariance au seuil 80%* - $\Delta < 10^{-3}$

Question 12 : Comparer les résultats des deux questions précédentes et expliquer en quoi le modèle avec des matrices de covariance proportionnelles à l'identité n'est pas adapté à notre exemple.

L'image précédente est assez claire : l'hypothèse des matrices de variance-covariance proportionnelles à l'identité n'est pas adaptée à notre jeu de données, du moins pour 2 clusters (le violet et le cyan). Pour ces clusters, on se rend compte que d'une part les cercles contiennent des observations appartenant aux autres clusters, et d'autre part ne contiennent pas les observations aux extrêmités des ellipses. Ces ellipses de covariance sont aussi appelées ellipses de confiance car elles généralisent la notion d'intervalle de confiance au cas 2D. Imaginons que nous voulions utiliser l'algorithme EM dans un but prédictif. Alors des observations très éloignées des ellipses de confiance mais appartenant aux cercles seraient associées au mauvais cluster (au seuil 80%), ce qui n'est pas souhaité pour un algorithme de classification. De plus ici il s'agit d'ellipses de confiance à 80%, alors que classiquement nous utilisons plutôt des seuils égaux à 95% ou 97.5%. On comprend bien que pour de tels seuils, les cercles ci-dessus vont être nettement plus grands et donc d'autant moins pertinents. C'est ce qu'illustre la figure 9 page 19.

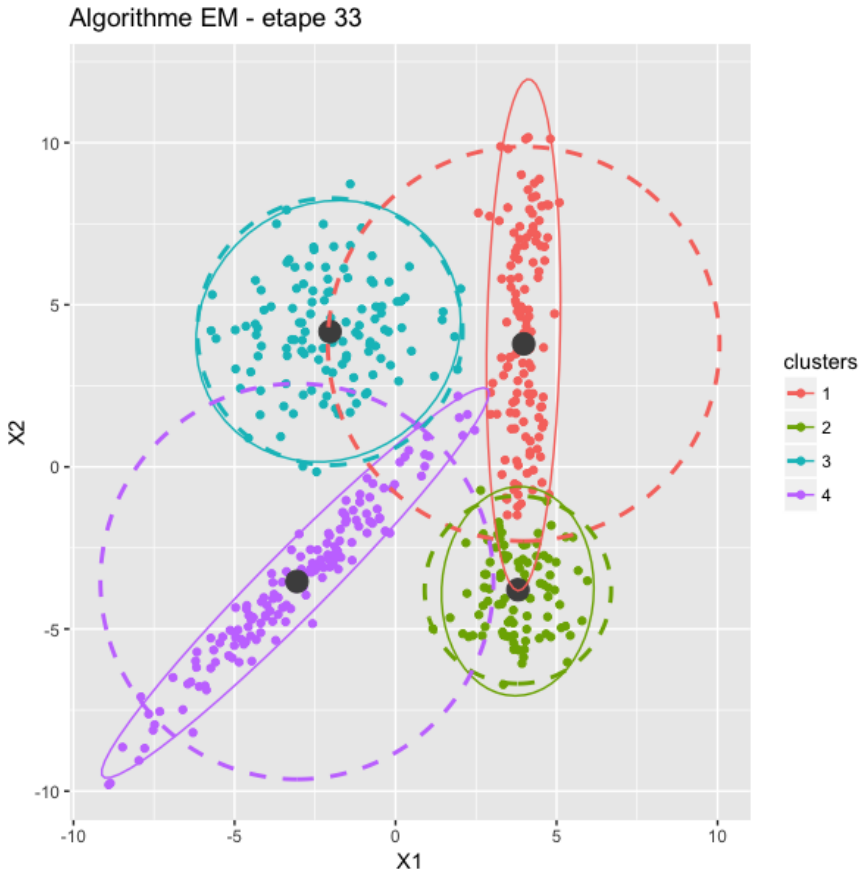

FIGURE 9 – *Comparaison des ellipses de covariance au seuil 95%* - $\Delta < 10^{-3}$

Références

[1] Jeff A. Bilmes. A Gentle Tutorial of the EM Algorithm and its Applications to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. Berkeley's International Computer Science Institute, 2012.

[2] Alexandre Lourme. Contribution à la Classification par Modèles de Mélange et Classification Simultanée d'Echantillons d'Origine Multiples. Université Lille 1, 2011.

[3] Frédéric Santos. L'algorithme EM : une courte présentation. CNRS, 2015.

[4] T. Mary-Huard E. Lebarbier. Classification non supervisée. AgroParisTech, 2015.

[5] Christophe Biernacki. Pourquoi les modèles de mélange pour la classification ? CNRS & Université Lille 1. *Revue Modulad*, (40), 2009.