Dauphine | PSL
UNIVERSITÉ PARIS

Mémoire Master 1

Quelles variables sont importantes pour le clustering ?

Auteurs :
Sixtine Sphabmixay
Josue Kamdem Kamwa
Harivola Rajaonah

Encadrante :
Madalina Olteanu

Table des matières

1 Introduction

L'apprentissage non-supervisé est une méthode d'analyse de données qui vise à découvrir des structures cachées dans un ensemble de données sans l'aide de labels ou de catégories prédéfinies. Contrairement à l'apprentissage supervisé, où le modèle est entraîné à prédire des étiquettes à partir d'un ensemble de données étiquetées, l'apprentissage non supervisé ne dispose pas d'informations de catégorisation préalables.

Le clustering est une méthode d'apprentissage non supervisé couramment utilisée pour découvrir des structures dans les données. Elle permet de regrouper des données similaires ou proches les unes des autres en un ensemble de groupes ou clusters. Cela peut avoir de nombreuses applications, telles que le traitement d'image, où l'on peut chercher à distinguer le même motif dans une galerie de photos, le marketing où les entreprises cherchent à établir un profil de leurs clients afin de leur proposer les produits ou services les plus adéquats, ou bien dans la biologie où elle peut être utile pour retraçer l'évolution dans le temps des populations.

L'un des modèles les plus couramment utilisés dans le clustering est le modèle de mélange gaussien. Ce modèle suppose que les données dans chaque cluster suivent une distribution normale, ce qui permet de déterminer les paramètres de la distribution pour chaque cluster. Cependant, l'analyse de données volumineuses et hétérogènes peut poser des problèmes pour l'estimation des paramètres, en particulier lorsque le nombre de variables est très élevé par rapport au nombre d'observations. D'autant plus que toutes les variables ne seront pas pertinentes pour la création de clusters, et certaines pourraient même fausser notre modèle. Dans ce contexte, la sélection de variables est un enjeu crucial pour améliorer la qualité des clusters et faciliter leur interprétation.

Dans ce mémoire, nous nous concentrerons sur la problématique de la sélection de variables dans le contexte du modèle de mélange gaussien. Notre objectif est de fournir des outils pratiques et efficaces pour identifier les variables les plus importantes pour la construction de clusters basés sur ce modèle.

En se basant sur l'article de **Wei Pan and Xiaotong Shen** intitulé **"Penalized Model-Based Clustering with Application to Variable Selection"**, nous présenterons tout d'abord l'algorithme EM (Estimation-Maximisation) sans pénalité afin de déterminer les paramètres de notre modèle et nous parlerons également de l'algorithme EM pénalisé, qui est une première méthode de sélection de variables. Ensuite, nous examinerons la construction du Bayesian Information Criterion (BIC), un critère d'évaluation largement utilisé dans la sélection de variables pour le modèle de mélange gaussien. Enfin, en s'appuyant sur l'aricle de **Adrian E Raftery and Nema Dean** intitulé **"Variable Selection for Model-Based Clustering"**, nous présenterons une deuxième méthode de sélection de variables basée sur le facteur de Bayes.

2 Présentation du modèle de mélange

On dispose d'un n-échantillon $\mathbf{x} := (x_1, ..., x_n)$ caractérisé par p variables continues, c'est-à-dire que $\forall i = \{1, ..., n\}$, $x_i \in \mathbb{R}^p$.
Pour un entier K fixé, on suppose que nos données sont issues de K vecteurs aléatoires gaussiens de paramètres (μ_k, Σ_k) pour tout $k \in \{1, ..., K\}$. On note $(\pi_1, ..., \pi_K)$ les proportions des différents groupes. Ici K est un hyperparamètre du modèle qui représente aussi le nombre de clusters considérés.
On suppose également que les matrices de variance-covariance sont diagonales et identiques pour chaque cluster. Afin de faciliter les notations par la suite, posons :

$$V := \Sigma_1 = ... = \Sigma_K = diag(\sigma_1^2, ..., \sigma_p^2)$$

Posons $\Phi := (\pi_1, ..., \pi_K, \mu_1, ...\mu_K, \sigma_1, ..., \sigma_p)$ les paramètres du mélange inconnus, où $\pi_k \in (0, 1)$, $\mu_k \in \mathbb{R}^p$, $\sigma_j \in \mathbb{R}^+$, $\forall k \in \{1, ..., K\}$, $\forall j \in \{1, ..., p\}$. On cherche les valeurs de Φ qui représentent au mieux notre mélange gaussien. Pour cela, on détermine l'estimateur du maximum de vraisemblance du mélange gaussien. La probabilité d'obtenir la donnée x dans notre mélange de K gaussiennes est la suivante :

$$f(x) = \sum_{k=1}^{K} \pi_k f_k(x)$$

où π_k est la proportion de la population dans le k-ème groupe et f_k la densité gaussienne de paramètre (μ_k, V) pour le groupe k.

On cherche ensuite à déterminer l'estimateur du maximum de vraisemblance de notre mélange gaussien, c'est-à-dire à déterminer Φ qui maximise la log-vraisemblance suivante :

$$l(\mathbf{x}, \Phi) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k f_k(x_i, \mu_k, V) \right)$$

Or cette quantité est difficile à maximiser.
Considérons un exemple simple, où l'on possède trois groupes et trois gaussiennes de dimension 1. On cherche alors à optimiser une fonction non linéaire de 6 paramètres, ce qui semble fastidieux à résoudre si l'on utilise la méthode habituelle consistant à déterminer les dérivées partielles la log-vraisemblance et trouver les points pour lesquelles elles s'annulent. Pour contourner cette difficulté, nous présenterons dans la suite l'algorithme Espérance-Maximisation.

3 Algorithme Espérance-Maximisation

Il existe plusieurs méthodes pour déterminer les paramètres d'un modèle gaussien. Dans notre mémoire, nous présenterons l'algorithme EM (Expectation-Maximization) afin de déterminer l'estimateur du maximum de vraisemblance d'un mélange gaussien.
L'algorithme EM est une méthode d'optimisation largement utilisée pour estimer les paramètres de modèles probabilistes ayant des variables latentes (cachées). L'idée principale de cet algorithme est de trouver les paramètres du modèle qui maximisent la vraisemblance des données observées en utilisant les données complètes, c'est-à-dire en tenant compte des variables latentes. Le principe de l'algorithme EM repose sur deux étapes principales :

Etape E (Expectation) : Dans cette étape, on calcule l'espérance de la log-vraisemblance des données complètes étant donné les données observées et les estimations actuelles des paramètres du modèle. Cette espérance est calculée en utilisant la distribution a posteriori des variables latentes étant donné les données observées et les estimations actuelles des paramètres.

Etape M (Maximization) : Dans cette étape, on met à jour les estimations des paramètres du modèle en maximisant l'espérance de la log-vraisemblance des données complètes calculée lors de l'étape E. Les nouvelles estimations des paramètres sont obtenues en résolvant les équations de maximisation de la log-vraisemblance.

L'algorithme EM est itératif et alterne entre les étapes E et M jusqu'à convergence, c'est-à-dire jusqu'à ce que les changements dans les estimations des paramètres soient suffisamment petits.

L'utilisation des données complètes permet de simplifier considérablement le calcul de la vraisemblance et la mise à jour des paramètres du modèle. Dans de nombreux cas, les étapes E et M deviennent beaucoup plus faciles à résoudre en utilisant les données complètes plutôt que la vraisemblance marginale des données observées.
Considérons donc $Z_i \sim \mathcal{M}(K, \pi_1, ..., \pi_K)$. L'introduction de cette suite de variables aléatoires va nous permettre de déterminer la provenance des données.

Définissons un résultat préalable :
Montrons que si $X|Z = k$ est une variable aléatoire de densité $f_{(\mu_k, V)}$ alors X suit la loi du mélange. En notant F la fonction de répartition de $X|Z = k$ on a,

pour tout $x \in \mathbb{R}$:

$$\begin{aligned}\mathbb{P}(X \leq x) &= \sum_{k=1}^{K} \mathbb{P}(X \leq x, Z = k) \\ &= \sum_{k=1}^{K} \mathbb{P}(X \leq x | Z = k)\mathbb{P}(Z = k) \\ &= \sum_{k=1}^{K} \pi_k F_{(\mu_k, V)}(x)\end{aligned}$$

Donc X est bien une variable aléatoire de densité f.
Considérons $\mathbf{z} := (z_1, ..., z_n)$ un n-échantillon où z_i est une réalisation de Z_i, $\forall i \in \{1, ..., n\}$. On peut alors écrire la vraisemblance des données complètes :

$$L(\mathbf{x}, \mathbf{z}|\Phi) = \prod_{i=1}^{n} f(x_i, z_i|\Phi) = \prod_{i=1}^{n} f(x_i|z_i, \Phi) f(z_i|\Phi) = \prod_{i=1}^{n} \prod_{k=1}^{K} (\pi_k f_k(x_i, \mu_k, V))^{\mathbb{1}_{z_i=k}}$$

Comme les proportions du mélange sont comprises entre 0 et 1 et que la densité d'une loi normale de paramètre (μ_k, V) est positive, on peut passer à la log-vraisemblance :

$$l(\mathbf{x}, \mathbf{z}|\Phi) = \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{1}_{z_i=k} \log(\pi_k f_k(x_i, \mu_k, V))$$

3.1 Méthode

Détaillons un peu plus les étapes de cet algorithme :

Etape 0 : Initialisation

Tout d'abord, nous définissons les paramètres initiaux : $\begin{cases} \pi_1^0, ..., \pi_K^0 \in (0,1) \\ \mu_1^0, ..., \mu_K^0 \in \mathbb{R}^p \\ V^0 \in \mathbb{R}^{p \times p} \end{cases}$

où $\sum_{k=1}^K \pi_k^0 = 1$

Etape E : Dans l'étape E (Expectation), on calcule la probabilité a posteriori des variables latentes Z étant donné les données observées $\mathbf{x}$ et les estimations actuelles (à l'étape m-1) des paramètres $\Phi^{(m-1)}$. Cette probabilité a posteriori est utilisée pour déterminer le degré d'appartenance de chaque point de données x_i aux différents clusters k. Le but de cette étape est de remplacer les valeurs des $z_1, ..., z_n$ inconnues par leur espérance, ce qui facilitera l'estimation des paramètres $\Phi^{(m)}$ restant.

Comme $\mathbb{1}_{z_i=k}$ est bornée, on peut considérer l'espérance de la vraisemblance des données complètes étant donné les données observées et les estimations actuelles des paramètres du modèle.

$$Q(\Phi|\Phi^{(m-1)}) := \mathbb{E}_{\Phi^{(m-1)}}(l(\mathbf{x}, \mathbf{z}|\mathbf{x}, \Phi)) = \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik}^{(m)} \log\left(\pi_k f_k(x_i, \mu_k, V)\right)$$

où $\forall i \in \{1, ..., n\}, \forall k \in \{1, ..., K\}$

$$\tau_{ik}^{(m)} := \mathbb{E}_{\Phi^{(m-1)}}(\mathbb{1}_{z_i=k}|x_i, \Phi)$$

Estimons τ_{ik} grâce à l'étape (E). D'après la formule de Bayes on a à l'itération m,

$$\begin{aligned} \tau_{ik}^{(m)} &= \mathbb{E}(\mathbb{1}_{z_i=k}|x_i) \\ &= \mathbb{P}(z_i = k|x_i) \\ &= \frac{\mathbb{P}(z_i = k, X = x_i)}{\mathbb{P}(X = x_i)} \\ &= \frac{f(x_i|z_i = k)\mathbb{P}(z_i = k)}{\mathbb{P}(X = x_i)} \\ &= \frac{\pi_k^{(m-1)} f_k(x_i, \mu_k^{(m-1)}, V^{(m-1)})}{\sum_{k=1}^K \pi_k^{(m-1)} f_k(x_i, \mu_k^{(m-1)}, V^{(m-1)})} \end{aligned}$$

Etape M : L'étape M (Maximization) de l'algorithme EM pour un mélange gaussien vise à mettre à jour les estimations des paramètres du modèle en maximisant l'espérance de la log-vraisemblance des données complètes, calculée lors de l'étape E. On cherche ainsi à estimer $\Phi^{(m)}$, les paramètres à l'étape m. Pour cela on détermine les points pour lesquels le gradient de Q est nul.

Rappelons que $\forall k \in \{1, ..., K\}$,

$$f_k(\mathbf{x}, \Phi) = \frac{e^{(-\frac{(x-\mu_k)^T V^{-1}(x-\mu_k)}{2})}}{(2\pi)^{p/2} \prod_{j=1}^{p} \sigma_j}$$

Réécrivons Q :

$$\begin{aligned} Q(\Phi|\Phi^{(m-1)}) &= \sum_{i=1}^{n} \sum_{j=1}^{K-1} \tau_{ij}^{(m)} (\log(\pi_j f_j(x_i, \mu_j, V))) + \sum_{i=1}^{n} (\tau_{iK}^{(m)} (\log(1 - \sum_{j=1}^{K-1} \pi_j)) + \log(f_K(x_i, \mu_K, V))) \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik}^{(m)} (\log(\pi_k) - \frac{\log((2\pi)^p det(V))}{2} - \frac{(x_i - \mu_k)^T V^{-1} (x_i - \mu_k)}{2}) \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik}^{(m)} (\log(\pi_k) - \frac{\log((2\pi)^p)}{2} - \frac{\sum_{j=1}^{p} \log(\sigma_j^2)}{2} - \frac{\sum_{j=1}^{p} \frac{(x_{ij} - \mu_{kj})^2}{\sigma_j^2}}{2}) \end{aligned}$$

Soit $k \in \{1, ..., K\}$, Q est de classe C^1 en chacunes de ses composantes comme somme, produit et composée de fonctions qui le sont. En utilisant directement les expressions de Q ci-dessus, on obtient :

$$\frac{\partial Q(\Phi|\Phi^{(m-1)})}{\partial \pi_k} = \sum_{i=1}^{n} (\frac{\tau_{ik}^{(m)}}{\pi_k} - \frac{\tau_{iK}^{(m)}}{\pi_K})$$

D'où,

$$\frac{\partial Q(\Phi|\Phi^{(m-1)})}{\partial \pi_k} = 0 \Leftrightarrow \pi_k = \frac{\pi_K \sum_{i=1}^{n} \tau_{ik}^{(m)}}{\sum_{i=1}^{n} \tau_{iK}^{(m)}}$$

Or,

$$\sum_{k=1}^{K} \pi_k = 1$$

et

$$\sum_{k=1}^{K} \sum_{i=1}^{n} \tau_{ik}^{(m)} = \sum_{i=1}^{n} \sum_{k=1}^{K} \frac{\pi_k^{(m-1)} f_k(x_i, \mu_k^{(m-1)}, V^{(m-1)})}{\sum_{k=1}^{K} \pi_k^{(m-1)} f_k(x_i, \mu_k^{(m-1)}, V^{(m-1)})} = n$$

Donc,

$$1 = \pi_K \frac{\sum_{k=1}^{K} \sum_{i=1}^{n} \tau_{ik}^{(m)}}{\sum_{i=1}^{n} \tau_{iK}^{(m)}} \Leftrightarrow \pi_K = \frac{\sum_{i=1}^{n} \tau_{iK}^{(m)}}{n}$$

Et ainsi,

$$\pi_k^{(m)} = \frac{\sum_{i=1}^n \tau_{ik}^{(m)}}{n}$$

D'autre part,

$$\frac{\partial Q(\Phi|\Phi^{(m-1)})}{\partial \mu_k} = \sum_{i=1}^n \tau_{ik}^{(m)} V^{-1}(x_i - \mu_k)$$

En effet, V est symétrique et son inverse l'est également et on obtient :

$$\frac{\partial (x_i - \mu_k)^T V^{-1}(x_i - \mu_k)}{\partial \mu_k} = -2V^{-1}(x_i - \mu_k)$$

En annulant la dérivée partielle de Q par rapport à μ_k on obtient :

$$\mu_k^{(m)} = \frac{\sum_{i=1}^n \tau_{ik}^{(m)} x_i}{\sum_{i=1}^n \tau_{ik}^{(m)}}$$

Déterminons enfin V :

Toujours d'après l'expression de Q donnée précédemment on obtient, pour tout $j \in \{1, ..., p\}$:

$$\frac{\partial Q(\Phi|\Phi^{(m-1)})}{\partial \sigma_j^2} = \sum_{i=1}^n \sum_{k=1}^K \tau_{ik}^{(m)} (-\frac{1}{2\sigma_j^2} + \frac{(x_{ij} - \mu_{kj})^2}{2\sigma_j^4})$$

Or,

$$\frac{\partial Q(\Phi|\Phi^{(m-1)})}{\partial \sigma_j^2} = 0 \Leftrightarrow \sum_{i=1}^n \sum_{k=1}^K \tau_{ik}^{(m)} = \sum_{i=1}^n \sum_{k=1}^K \tau_{ik}^{(m)} \frac{(x_{ij} - \mu_{kj})^2}{\sigma_j^2}$$

soit,

$$\sigma_j^{2(m)} = \sum_{i=1}^n \sum_{k=1}^K \tau_{ik}^{(m)} \frac{(x_{ij} - \mu_{kj})^2}{n}$$

Récapitulons les résultats obtenus :

$$\pi_k^{(m)} = \frac{\sum_{i=1}^n \tau_{ik}^{(m)}}{n}$$

$$\mu_k^{(m)} = \frac{\sum_{i=1}^n \tau_{ik}^{(m)} x_i}{\sum_{i=1}^n \tau_{ik}^{(m)}}$$

$$\sigma_j^{2(m)} = \sum_{i=1}^n \sum_{k=1}^K \tau_{ik}^{(m)} \frac{(x_{ij} - \mu_{kj}^{(m)})^2}{n}$$

Après avoir mis à jour les paramètres, l'algorithme EM retourne à l'étape E pour recalculer les probabilités a posteriori des variables latentes Z_i en utilisant les nouvelles estimations des paramètres. L'algorithme EM continue d'alterner entre les étapes E et M jusqu'à ce que les changements dans les estimations des paramètres soient suffisamment petits ou qu'un critère d'arrêt spécifique soit atteint.

3.2 Convergence de l'algorithme EM

Montrons à présent que la vraisemblance est bien améliorée à chaque étape de l'algorithme.

Supposons que l'on se trouve à la fin d'une itération (m) de l'algorithme EM. On cherche Φ telle que,

$$\Delta(\Phi, \Phi^{(m)}) = \log L(\mathbf{x}|\Phi) - \log L(\mathbf{x}|\Phi^{(m)}) \geq 0$$

Tout d'abord,

$$L(\mathbf{x}|\Phi) = \sum_{\mathbf{z}\in\{1,\dots,K\}^n} L(\mathbf{x},\mathbf{z}|\Phi) = \sum_{\mathbf{z}\in\{1,\dots,K\}^n} L(\mathbf{x}|\mathbf{z},\Phi)L(\mathbf{z}|\Phi)$$

On a ensuite,

$$\begin{aligned}\Delta(\Phi, \Phi^{(m)}) &= \ln(\sum_{\mathbf{z}\in\{1,\dots,K\}^n} L(\mathbf{x}|\mathbf{z},\Phi)L(\mathbf{z}|\Phi)) - \sum_{\mathbf{z}\in\{1,\dots,K\}^n} L(\mathbf{z}|\mathbf{x},\Phi^{(m)}) \ln L(\mathbf{x}|\Phi^{(m)}) \\ &= \ln(\sum_{\mathbf{z}\in\{1,\dots,K\}^n} \frac{L(\mathbf{x}|\mathbf{z},\Phi)L(\mathbf{z}|\Phi)}{L(\mathbf{z}|\mathbf{x},\Phi^{(m)})} L(\mathbf{z}|\mathbf{x},\Phi^{(m)})) - \sum_{\mathbf{z}\in\{1,\dots,K\}^n} L(\mathbf{z}|\mathbf{x},\Phi^{(m)}) \ln L(\mathbf{x}|\Phi^{(m)})\end{aligned}$$

Or $x \mapsto \log(x)$ est concave et $\sum_{\mathbf{z}\in\{1,\dots,K\}^n} L(\mathbf{z}|\mathbf{x},\Phi^{(m)}) = 1$. D'après l'inégalité de Jensen, on a donc :

$$\ln(\sum_{\mathbf{z}\in\{1,\dots,K\}^n} \frac{L(\mathbf{x}|\mathbf{z},\Phi)L(\mathbf{z}|\Phi)}{L(\mathbf{z}|\mathbf{x},\Phi^{(m)})} L(\mathbf{z}|\mathbf{x},\Phi^{(m)})) \geq \sum_{\mathbf{z}\in\{1,\dots,K\}^n} \ln(\frac{L(\mathbf{x}|\mathbf{z},\Phi)L(\mathbf{z}|\Phi)}{L(\mathbf{z}|\mathbf{x},\Phi^{(m)})}) L(\mathbf{z}|\mathbf{x},\Phi^{(m)})$$

Ainsi,

$$\begin{aligned}\Delta(\Phi, \Phi^{(m)}) &\geq \sum_{\mathbf{z}\in\{1,\dots,K\}^n} \ln(\frac{L(\mathbf{x}|\mathbf{z},\Phi)L(\mathbf{z}|\Phi)}{L(\mathbf{z}|\mathbf{x},\Phi^{(m)})L(\mathbf{x}|\Phi^{(m)})}) L(\mathbf{z}|\mathbf{x},\Phi^{(m)}) \\ &= \sum_{\mathbf{z}\in\{1,\dots,K\}^n} L(\mathbf{z}|\mathbf{x},\Phi^{(m)}) \ln(\frac{L(\mathbf{x},\mathbf{z}|\Phi)}{L(\mathbf{z},\mathbf{x}|\Phi^{(m)})})\end{aligned}$$

Or $\sum_{\mathbf{z}\in\{1,\dots,K\}^n} L(\mathbf{z}|\mathbf{x},\Phi^{(m)}) \ln(L(\mathbf{x},\mathbf{z}|\Phi)) = \mathbb{E}_{\Phi^{(m)}}(\ln L(\mathbf{x},\mathbf{z}|\Phi)|\mathbf{x})$

Donc,

$$\Delta(\Phi, \Phi^{(m)}) \geq \mathbb{E}_{\Phi^{(m)}}(\ln L(\mathbf{x}, \mathbf{z}|\Phi)|\mathbf{x}) - \sum_{\mathbf{z}\in\{1,\dots,K\}^n} L(\mathbf{z}|\mathbf{x}, \Phi^{(m)}) \ln\left(L(\mathbf{z}, \mathbf{x}|\Phi^{(m)})\right)$$

Notons $\delta(\Phi, \Phi^{(m)}) := \mathbb{E}_{\Phi^{(m)}}(\ln L(\mathbf{x}, \mathbf{z}|\Phi)|\mathbf{x}) - \sum_{\mathbf{z}\in\{1,\dots,K\}^n} L(\mathbf{z}|\mathbf{x}, \Phi^{(m)}) \ln\left(L(\mathbf{z}, \mathbf{x}|\Phi^{(m)})\right)$. On a immédiatement que $\delta(\Phi^{(m)}|\Phi^{(m)}) = 0$. D'autre part, comme

$$\Phi^{(m+1)} = \arg\max_{\Phi} \delta(\Phi|\Phi^{(m)}) = \arg\max_{\Phi} \mathbb{E}_{\Phi^{(m)}}(\ln L(\mathbf{x}, \mathbf{z}|\Phi)|\mathbf{x})$$

On obtient que $\Delta(\Phi^{(m+1)}, \Phi^{(m)}) \geq \delta(\Phi^{(m+1)}|\Phi^{(m)}) \geq \delta(\Phi^{(m)}|\Phi^{(m)}) = 0$.

Ainsi, à chaque étape on trouve de nouvelles valeurs de Φ qui augmentent la valeur de la vraisemblance.

3.3 Illustration de l'algorithme EM

3.3.1 Mélange gaussien en dimension 1

Dans le but de tester l'efficacité de notre programme, nous mélangeons deux échantillons gaussiens de dimension 1 :

— Un échantillon de taille 400, de moyenne 2, et de variance 1

— Un échantillon de taille 100, de moyenne 6, et de variance 1

On obtient alors une matrice 500x1. Nous savons qu'il y a exactement deux clusters au sein de l'objet J.

En paramètres initiaux, on avons entré les valeurs suivantes :

	π^0	μ^0	σ^0
cluster 1	0,5	3,91	3,49
cluster 2	0,5	8,32	3,49

Après avoir fait tourner l'algorithme, on retrouve les paramètres suivants : :

	$\hat{\pi}$	$\hat{\mu}$	$\hat{\sigma}$
cluster 1	0,81	2,07	1,05
cluster 2	0,19	6,06	1,05

Grâce à la distance courante ci-dessus, on a une idée de la vitesse de convergence.

$$dist_c = l(\mathbf{x}, \Phi^{(m-1)}) - l(\mathbf{x}, \Phi^{(m)})$$

On remarque en effet que la courbe décroît rapidement, cela nous conforte dans l'idée que l'algorithme converge vers un minimum local.

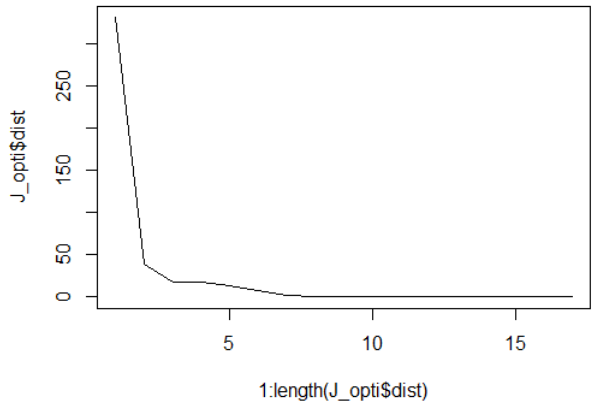


FIGURE 1 – Vitesse de convergence de l'algorithme EM

3.3.2 Mélange gaussien en dimension 2

Regardons ce qui se passe en dimension 2. On va considérer un échantillon de taille n=500 et de 8 variables.

Notons :
— $A_1 \sim \mathcal{N}((-4,0),\, I_2)$
— $A_2 \sim \mathcal{N}((0,0),\, I_2)$
— $A_3 \sim \mathcal{N}((4,0),\, I_2)$
— $A_4 \sim \mathcal{N}((0,-4),\, I_2)$
— $A_5 \sim \mathcal{N}((0,4),\, I_2)$

On simule 100 échantillons pour chaque variable ci-dessus et on crée une matrice de taille 500x2 comme suit : $\begin{pmatrix} A^1 \\ A^2 \\ A^3 \\ A^4 \\ A^5 \end{pmatrix}$

Notons X^1 la première colonne de cette matrice et X^2 la deuxième colonne. On définit alors nos 8 variables comme suit :
— X^1
— X^2
— $X^3, X^4, X^5 \sim X^1 + \mathcal{U}([-1;1])$
— $X^6, X^7, X^8 \sim X^2 + \mathcal{U}([-1;1])$

On a finalement une matrice $X = \left(X^1, X^2, X^3, X^4, X^5, X^6, X^7, X^8\right) \in \mathbb{R}^{500\times 8}$.

On a construit nos données de telle sorte qu'on ait 5 clusters :

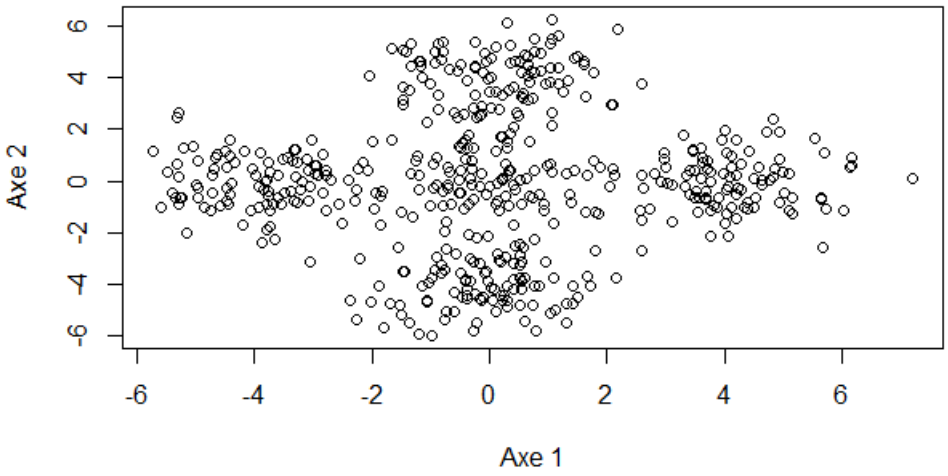


FIGURE 2 – Mélange gaussien en dimension 2

Après avoir fait tourner notre algorithme avec ce jeu de données, on retrouve les paramètres suivants :

Ce tableau contient $\hat{\mu}_{k,j}$ pour $k = 1, ..., 5$ et $j = 1, ..., 8$:

j,k	1	2	3	4	5
1	$\hat{\mu}_{1,1}$ =-0,13	-3,86	4,11	-0,02	0,15
2	-4,04	-0,07	-0,07	-0,05	4,00
3	-0,21	-3,75	3,98	0,02	0,21
4	-0,12	-3,90	4,24	0,02	0,24
5	-0,09	-3,92	4,12	-0,02	0,18
6	-4,13	-0,08	-0,09	-0,03	3,99
7	-4,04	-0,11	-0,09	-0,03	4,01
8	-4,14	-0,08	-0,01	-0,09	4,03

Ce tableau ci-dessous donne $\hat{\sigma}^2_{i,j}$ pour $i, j = 1, ..., 8$:

	1	2	3	4	5	6	7	8
1	0,926	0,000	0,000	0,000	0,000	0,000	0,000	0,000
2	0,000	0,957	0,000	0,000	0,000	0,000	0,000	0,000
3	0,000	0,000	1,190	0,000	0,000	0,000	0,000	0,000
4	0,000	0,000	0,000	1,248	0,000	0,000	0,000	0,000
5	0,000	0,000	0,000	0,000	1,195	0,000	0,000	0,000
6	0,000	0,000	0,000	0,000	0,000	1,315	0,000	0,000
7	0,000	0,000	0,000	0,000	0,000	0,000	1,277	0,000
8	0,000	0,000	0,000	0,000	0,000	0,000	0,000	1,165

Les proportions $\hat{\pi}_k$ pour $k = 1, ..., 5$ sont données par :

k	Proportions
1	0,199
2	0,205
3	0,197
4	0,190
5	0,208

Une fois ces résultats obtenus, nous avons calculés les τ_{ik} $\forall i \in \{1, ..., n\}$, $\forall k \in \{1, ..., K\}$ afin de déterminer la provenance de chaque donnée.
Plus précisément, nous avons attribué l'observation x_i au cluster k lorsque $k = \arg\max_{j \in \{1,...,K\}} \tau_{ij}$.
Afin de vérifier le bon fonctionnement de notre algorithme et d'évaluer ses performances, nous avons établis la matrice de confusion suivante :

	1	2	3	4	5
1	98	0	0	0	0
2	0	98	0	0	2
3	0	0	96	0	8
4	0	1	0	96	5
5	2	1	4	4	85

Sur l'axe verticale nous avons les classes prédites et sur l'autre les vraies classes.

Nous obtenons un taux d'erreur relativement faible de 5.7% et une précision de 98% pour les classe 1 et 2, 96% pour les classes 3 et 4, et 85% pour la classe 5. Notre algorithme EM sans pénalité semble donc plutôt efficace.

3.3.3 Jeu de donées Old Faithful

On va maintenant tester l'lagorithme EM sur des données réelles, où l'hypothèse de modèle de mélange gaussien n'estpas forcément vérifiée.
On utilise le jeu de données Old Faithful Geyser composé de 272 observations et 2 variables. Les graphes ci-dessous illustrent le déroulement de l'algorithme EM : à l'étape a) on initialise nos clusters. A l'étape b) on détermine les probabilités d'appartenance aux groupes, soit les τ_{ik}. A l'étape c) on détermine une première estimation de Φ. Comme décrit plus haut, on répète ce procédé jusqu'à

convergence et l'on obtient alors l'étape f).

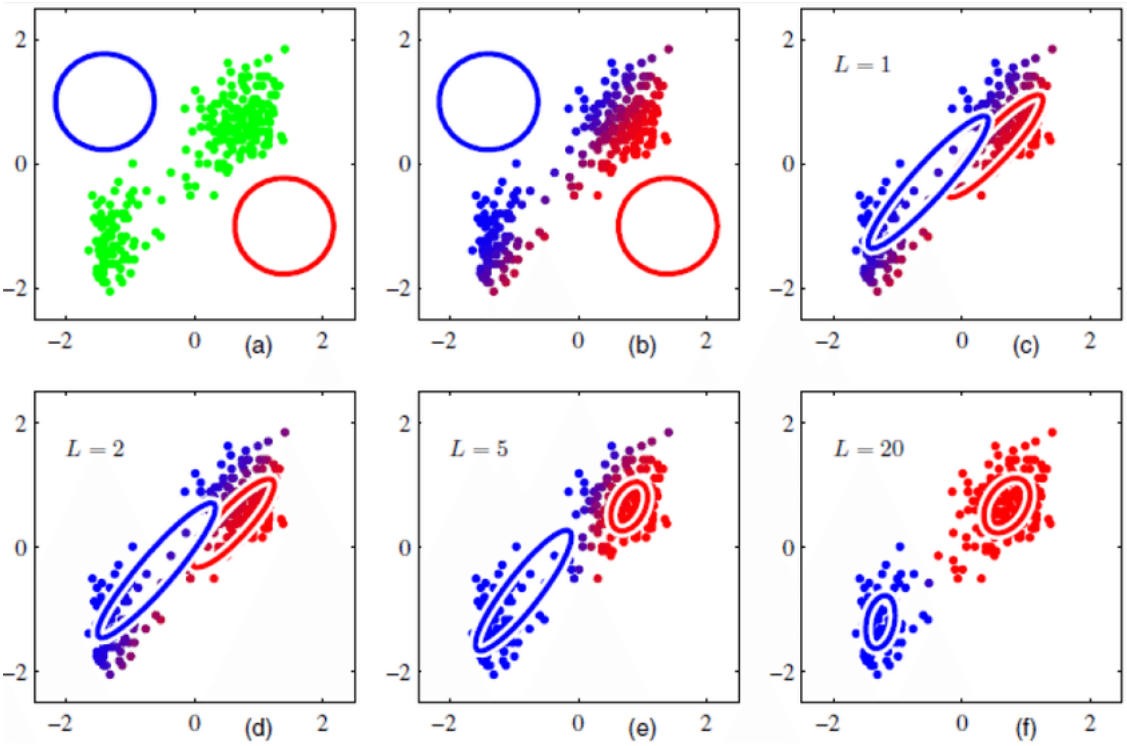


FIGURE 3 – Clustering des données Faithful

Ces résultats reviennent à Christopher M.Bishop publiés en 2006 dans son livre "Pattern Recognition And Machine Learning".

Grâce aux travaux de Monsieur Bishop, nous savons déjà à l'avance que le nombre de clusters est 2. Revenons une fois de plus à la description de notre dataframe R faithful. Le dataset faithful comprend 272 observations (lignes) et 2 variables (colonnes). Les noms de ces deux variables sont "eruptions" et "waiting". Ce jeu décrit le temps d'attente entre les éruptions et la durée de l'éruption pour le geyser Old Faithful dans le parc national de Yellowstone, Wyoming, USA. Ainsi la première variable "eruptions" représente le temps d'éruption en minutes. Et la seconde variable "waiting" représente le temps d'attente en minutes avant la prochaine éruption.

Pour ce jeu de données à deux variables, nous allons représenter chaque ligne (individu) comme un point de $\mathbb{R}^2$. Ensuite nous allons dessiner sur le plot des points des ellipses dont les centres seront les moyennes calculées par notre algorithme EM .Les rayons des ellipses quant à eux seront proportionnels aux variances calculées par l'algorithme.

Grâce au graphique, on voit clairement que notre jeu de données faithful a bel

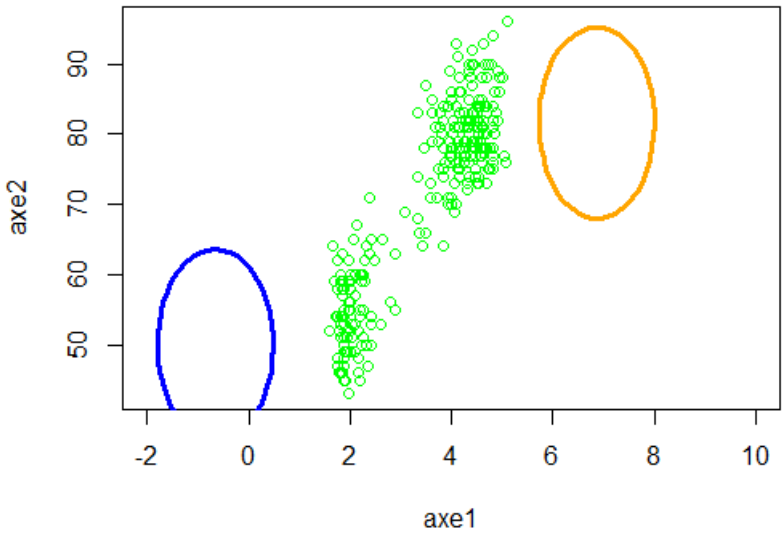


FIGURE 4 – Illustration des données avec les paramètres initiaux

et bien deux principaux clusters. On voit également que les ellipses n'entourent pas vraiment les données.

On observe très bien que le centre de chaque ellipse a été déplacé et nos ellipses "entourent" chacune un sous-groupe de notre dataset. Le déplacement de nos ellipses s'expliquent par le fait que les centres de nos ellipses sont désormais les moyennes calculées par notre programme d'EM sans pénalité. En effet, notre programme a pour but de calculer (ou à défaut d'approcher un maximum) les véritables moyennes (proportions et variances également) de chaque cluster présent dans le jeu de données. Par conséquent le fait que chaque ellipse ait migré vers un sous-groupe justifie que notre programme d'EM sans pénalité est bon pour cette simulation.

Ce test avec le jeu de données faithful nous renforce dans l'idée que notre algorithme EM sans pénalité fonctionne bien.

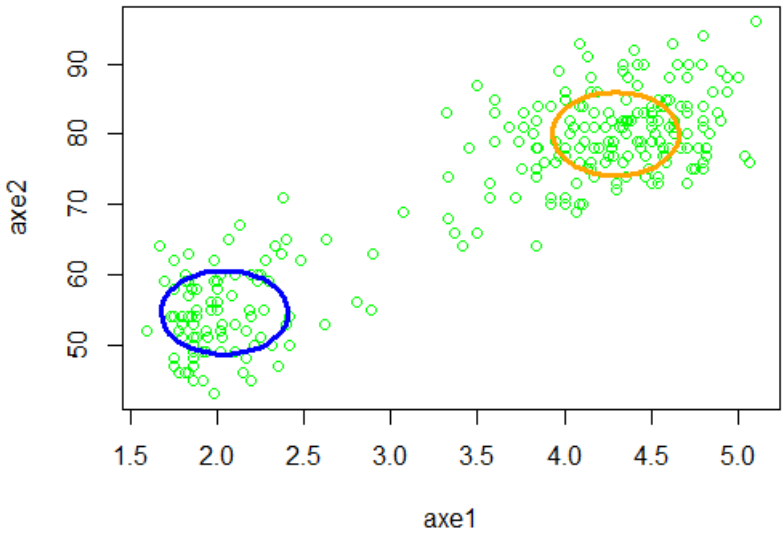


FIGURE 5 – Illustration des données après l'algorithme EM

4 Sélection de modèles

En apprentissage non supervisé, le choix du nombre de clusters est crucial car il détermine la complexité de la structure de clustering. Un modèle trop simple pourrait manquer des structures subtiles dans les données, tandis qu'un modèle trop complexe pourrait surajuster les données et ne pas généraliser correctement.

Le Bayesian Information Criterion (BIC) est une méthode couramment utilisée pour déterminer le nombre optimal de clusters dans l'analyse de clustering. Le BIC est un critère d'évaluation qui cherche à trouver un compromis entre l'ajustement et la complexité du modèle.

Pour utiliser le BIC pour déterminer le nombre optimal de clusters, il faut faire varier le nombre de clusters K et calculer le BIC correspondant à chaque K. Le nombre optimal de clusters K correspond alors au K qui minimise le BIC. En d'autres termes, le BIC mesure la qualité de l'ajustement du modèle aux données tout en prenant en compte sa complexité. Plus le BIC est petit, meilleur est le modèle.

Il est intéressant de voir tout d'abord en détails comment se fait la construction de ce critère.

4.1 Construction du critère BIC

On considère un vecteur aléatoire $X = (X_1, ..., X_p)$ dont les coordonnées sont deux-à-deux indépendants de densité inconnue f que l'on souhaite estimer. On se donne une collection finie de modèles $M_1, ..., M_m$ où un modèle M_i correspond à une famille de densités g_{M_i} de paramètre θ_i de dimension K_i et Θ_i l'espace de dimension K_i auquel appartient θ_i.

θ_i et M_i sont des variables aléatoires munies d'une distribution **a priori**. La distribution a priori sur M_i est notée $P(M_i)$ et pour un M_i donné, la distribution a priori de θ_i est notée $P(\theta_i|M_i)$.

L'idée du **BIC** est de sélectionner le modèle le plus vraisemblable au vu des données X qui sont iid, c'est-à-dire qu'on cherche le modèle M_i qui maximise la probabilité **a posteriori** $P(M_i|X)$: $M_{BIC} = \underset{M_i}{\text{argmax}} P(M_i|X)$.

D'après la formule de Bayes :

$$P(M_i|X) = \frac{P(X|M_i)P(M_i)}{P(X)}$$

Nous supposons que la loi à priori des M_i est non informative donc

$$P(M_1) = ... = P(M_m).$$

Comme il n'y a pas de raison de préférer un modèle à un autre, il s'ensuit que :

$$P(M_i|X) \approx P(X|M_i)$$

Ainsi,

$$P(X|M_i) = \int_{\Theta_i} P(X, \theta_i|M_i)d\theta_i = \int_{\Theta_i} g_{M_i}(X, \theta_i)P(\theta_i|M_i)d\theta_i \qquad (1)$$

Par la formule de Bayes et où $g_{M_i} = P(X|\theta_i, M_i)$ la vraisemblance correspondant au modèle M_i de paramètre θ_i

$$P(X|M_i) = \int_{\Theta_i} e^{g(\theta_i)}d\theta_i$$

où $g(\theta_i) = log(g_{M_i}(X, \theta_i)P(\theta_i|M_i))$

Or il est difficile de calculer la valeur exacte de cette intégrale. Pour remédier à ce problème, appliquons l'approximation de Laplace dont nous rappelons la définition :

Soit $L : \mathbb{R}^d \to \mathbb{R}$ telle que L est deux fois différentiable sur $\mathbb{R}^d$ et atteint un unique maximum sur $\mathbb{R}^d$ en u^*. Alors :

$$\int_{\mathbb{R}_d} e^{nL(u)}du = e^{nL(u^*)}(\frac{2\pi}{n})^{\frac{d}{2}}|-L''(u^*)|^{\frac{-1}{2}} + \mathcal{O}\left(\frac{1}{n}\right)$$

Dans notre cas : $L_n(\theta_i) = \frac{g(\theta_i)}{n} = \frac{1}{n}\sum_{k=1}^{n} log(g_{M_i}(X_k, \theta_i)) + \frac{log(P(\theta_i|M_i))}{n}$

Notons :

— $\theta_i^* = \underset{\theta_i \in \Theta_i}{\text{argmax}} L_n(\theta_i)$

— $A_{\theta_i^*} = -[\frac{\partial^2 L_n(\theta_i^*)}{\partial\theta_i^j \partial\theta_i^l}]_{j,l}$ où θ_i^j est la j-ième composante du vecteur des paramètres θ_i et $A_{\theta_i^*}$ est l'opposé de la matrice Hessienne des dérivées secondes partielles de la fonction $L_n(\theta_i)$ en θ_i^*.

On obtient alors,

$$P(X|M_i) = e^{\frac{g_n(\theta_i^*)}{n}(\frac{2\pi}{n})^{\frac{K_i}{2}}|A_{\theta_i^*}|^{\frac{-1}{2}} + \mathcal{O}\left(\frac{1}{n}\right)}$$

ou encore,

$$log(P(X|M_i)) = log(g_{M_i}(X, \theta_i^*)) + log(P(\theta_i^*|M_i)) + \frac{K_i}{2}log(\frac{2\pi}{n}) - \frac{1}{2}log(|A_{\theta_i^*}|) + \mathcal{O}\left(\frac{1}{n}\right)$$

La difficulté est maintenant d'évaluer θ_i^* et $A_{\theta_i^*}$.
On a montré en Annexe A que l'on peut remplacer :

— θ_i^* par l'EMV $\hat{\theta}_i$

$$\hat{\theta}_i = \underset{\theta_i \in \Theta_i}{\text{argmax}} \frac{1}{n} g_{M_i}(X, \theta_i)$$

— et $A_{\theta_i^*}$ par $I_{\hat{\theta}_i}$ la matrice d'information de Fisher pour une observation qui est définie par

$$I_{\hat{\theta}_i} = -\mathbb{E}([\frac{\partial^2 log(g_{M_i}(X_i, \hat{\theta}_i))}{\partial\theta_i^j \partial\theta_i^l}])$$

Lorsque n est grand $log(g_{M_i}(X, \theta_i)P(\theta_i|M_i))$ se comporte comme $log(g_{M_i}(X, \theta_i))$ qui croît lorsque n croît tandis que $log(P(\theta_i|M_i))$ reste constant.

$$log(P(X|M_i)) \approx log(g_{M_i}(X, \hat{\theta}_i)) - \frac{K_i}{2}log(n)$$

C'est de cette approximation que le critère BIC est issu. Plus précisement pour le modèle M_i il correspond à l'approximation de $-2\log(P(X|M_i))$ et est donc défini par

$$BIC_i = -2log(g_{M_i}(X, \hat{\theta}_i)) + K_i log(n)$$

Et le modèle sélectionné par le critère est

$$M_{BIC} = \underset{M_i}{\text{argmin}} BIC_i = \underset{M_i}{\text{argmax}} - BIC_i.$$

4.2 Exemple de sélection de modèle

On a montré que pour le jeu de données Faithful, le K optimal est bien égal à 2. En effet, on obtient les BIC suivants :

K	BIC
2	$-2636,52$
3	$-2767,33$
4	$-2955,36$
5	$-3089,35$

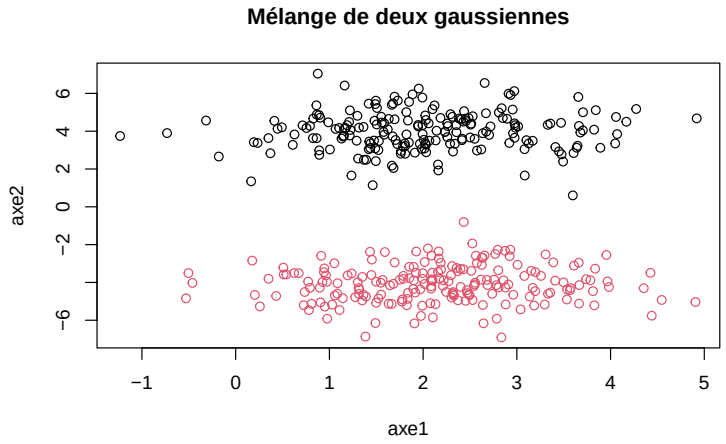
Comme dit ci-dessus on cherche à maximiser -BIC d'où $K = 2$.

5 Première méthode de sélection de variable : Algorithme Espérance-Maximisation avec pénalité

5.1 Motivation

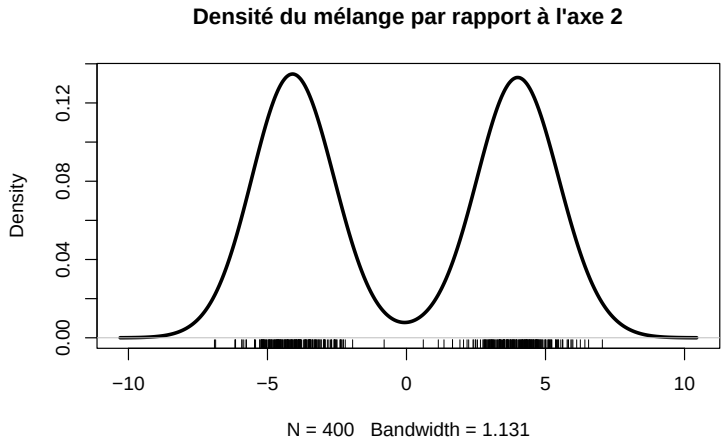
Comme dit dans l'introduction, lorsque le nombre de variables explicatives dans les données est élevé, l'estimation des paramètres peut devenir difficile mais surtout cela introduit du bruit et les clusters peuvent être mal estimés. Dans ce cas, une pénalisation peut être appliquée. L'algorithme EM avec pénalité est une extension de l'algorithme EM qui inclut une pénalisation pour contrôler la complexité du modèle de mélange gaussien.

Prenons l'exemple suivant où l'on considère un mélange de deux gaussiennes $\mathcal{N}((2,4),I_2)$ et $\mathcal{N}((2.2,-4),I_2)$:

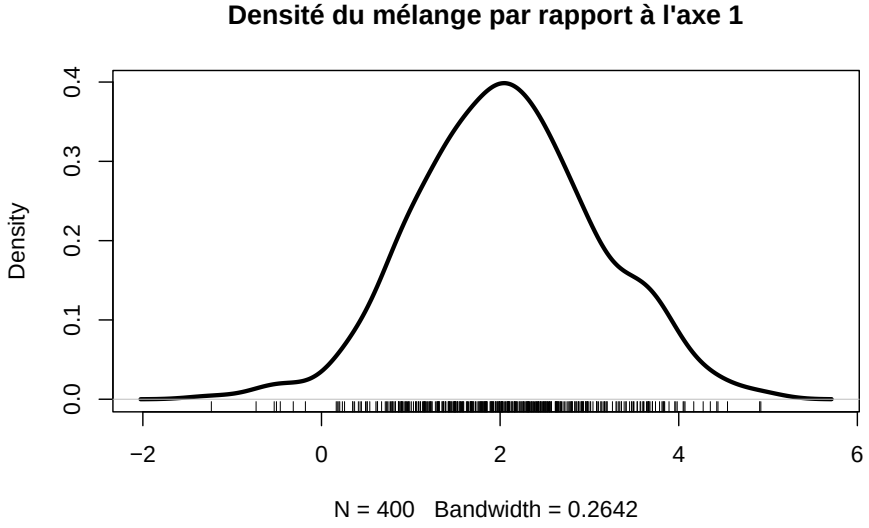


où les données représentées en rouge correspondent à celles issues de la gaussienne $\mathcal{N}((2.2,-4),I_2)$ tandis que les données en noir proviennent de la gaussienne $\mathcal{N}((2,4),I_2)$.

Considérons nos données par rapport à l'axe 2 :



On se trouve dans un cas où l'algorithme sans pénalité devrait bien fonctionner comme nos données proviennent de deux gaussiennes bien séparées. En revanche, si l'on considère nos données par rapport à l'axe 1, on obtient le graphe suivant :



Il semble difficile de distinguer la provenance de nos données comme les moyennes des deux gaussiennes sont très proche (2 et 2.2 dans notre cas). On voudrait "éliminer" cette dimension associée à une variable 1 comme elle ne nous donne pas d'information pour former les clusters. L'algorithme EM pénalisé proposé par Pan et Shen nous donne une première manière de faire une sélection de variables informatives.

Pour ce faire nous allons donc introduire une pénalité $h_\lambda(.)$ de type L^1. Ce type de pénalité a pour effet de forcer certains coefficients estimés contribuant peu à notre modèle d'être égaux à zéro.

Considérons $\mu_k = \mu + \delta_k$ où μ est la moyenne globale, μ_k la moyenne du cluster k. Si $\delta_{kp} = 0$ pour certains p alors les p variables en questions n'apportent pas d'informations pour le regroupement de nos données en terme de moyenne pour le cluster k. Dans la suite on suppose que nos données sont centrées donc $\mu = 0$. Le but de cette partie est alors de réduire à zéro les moyennes μ_{kp} où la p-ième composante n'est pas pertinente pour notre modèle. Cela peut être interprété comme une sélection de variables, car les variables correspondant aux éléments nuls sont considérées comme moins importantes ou non pertinentes pour le modèle.

5.2 Méthode

On considère la pénalité suivante :

$$h_\lambda(\Phi) = \lambda \sum_{k=1}^{K} \sum_{p=1}^{P} |\mu_{kp}|$$

On met ensuite en place un algorithme EM avec notre log-vraisemblance pénalisée. En gardant les hypothèses de la partie précédente, on peut l'écrire sous la forme suivante :

$$l(\mathbf{x}|\Phi) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k f_k(x_i|\mu_k, V)\right) - h_\lambda(\Phi)$$

et la log-vraisemblance des données complètes est

$$l(\mathbf{x}, \mathbf{z}|\Phi) = \sum_{k=1}^{K} \sum_{i=1}^{n} \mathbb{1}_{z_i=k} (\log (\pi_k f_k(x_i|\mu_k, V)) - h_\lambda(\Phi)$$

Le principe de l'algorithme EM ayant été décrit à la partie précédente, nous ne détaillerons pas son procédé et certains calculs.

Après l'initialisation de nos paramètres, l'étape d'estimation (E) donne le résultat suivant :

$$\begin{aligned} Q(\Phi|\Phi^{(m-1)}) : &= \mathbb{E}_{\Phi^{(m-1)}}(l(\mathbf{x}, \mathbf{z}|\Phi)|x_1, ..., x_n) \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik}^{(m)} (\log (\pi_k f_k(x_i, \mu_k, V))) - h_\lambda(\Phi) \end{aligned}$$

où $\forall i \in \{1, ..., n\}, \forall k \in \{1, ..., K\}$

$$\tau_{ik}^{(m)} = \frac{\pi_k^{(m-1)} f_k(x_i, \mu_k^{(m-1)}, V^{(m-1)})}{\sum_{k=1}^{K} \pi_k^{(m-1)} f_k(x_i, \mu_k^{(m-1)}, V^{(m-1)})}$$

L'étape de maximisation de Q (M) permet de mettre à jour les paramètres inconnus et nous obtenons les paramètres suivants :

$$\pi_k^{(m)} = \frac{\sum_{i=1}^{n} \tau_{ik}^{(m)}}{n}$$

$$\sigma_j^{2(m)} = \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik}^{(m)} \frac{(x_{ij} - \mu_{kj}^{(m)}))^2}{n}$$

A la différence de la partie précédente, la dérivée partielle de Q par rapport à μ_k n'est pas immédiate car la fonction $x \mapsto |x|$ n'est pas dérivable en 0.
Nous allons déterminer dans un premier temps le sous-différentiel de la norme 1, ce qui nous permettra de calculer l'estimateur de μ_k.

Rappelons les définitions des sous-gradient et du sous-différentiel :
Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction continue, pas nécessairement différentiable. $g \in \mathbb{R}^n$ est le sous-gradient de f au point $\mathbf{x} \in \mathbb{R}^n$ si et seulement si

$$f(y) - f(\mathbf{x}) \geq g^T(y - \mathbf{x}), y \in \mathbb{R}^n$$

Le sous-différentiel de f en x est l'ensemble

$$\partial f(\mathbf{x}) = \{g \in \mathbb{R}^n; \text{g est un sous-gradient de f en } \mathbf{x}\}$$

Déterminons tout d'abord le sous-différentiel de la fonction

$$\begin{array}{rl} |.| & : \mathbb{R} \to \mathbb{R} \\ & x \mapsto |x| \end{array}$$

On sait que cette fonction est dérivable pour tout $x \neq 0$ et sa dérivée vaut 1 si $x \geq 0$, -1 si $x \leq 0$. Déterminons à présent le sous-différentiel de la valeur absolue en 0.
En ce point les sous-gradients sont caractérisés par $|x| \geq gx$. Cela implique que $g \in [-1, 1]$. Ainsi, pour tout $x \in \mathbb{R}$:

$$\partial|x| = \begin{cases} 1, & x > 0 \\ [-1, 1], & x = 0 \\ -1, & x < 0 \end{cases}$$

Comme la fonction

$$\begin{array}{rl} ||.||_1 & : \mathbb{R}^n \to \mathbb{R} \\ & x \mapsto \sum_{i=1}^{n} |x_i| \end{array} .$$

est une somme fine de fonctions sous-différentiables, son sous-différentiel est la somme des sous-différentiels de ces fonctions. Pour déterminer le sous-gradient de la norme L1, nous examinons chaque élément de x individuellement. Ainsi, $\forall i \in \{1, ..., n\}$, le sous-gradient de la norme L1 au point x_i est défini comme suit :

$$\partial|x_i| = \begin{cases} 1, & x_i > 0 \\ [-1,1], & x_i = 0 \\ -1, & x_i < 0 \end{cases}$$

Ainsi on peut définir le sous-différentiel de la norme 1 :

$$\partial||x||_1 = \left\{ g \in \mathbb{R}^n : g_i = \begin{cases} 1, & x_i > 0 \\ [-1,1], & x_i = 0 \\ -1, & x_i < 0 \end{cases} , i \in \{1,...,n\} \right\}$$

Nous pouvons alors déterminer la dérivée partielle de Q par rapport à μ_k. Grâce au calculs précédent, nous obtenons aisément que

$$\frac{\partial Q(\Phi|\Phi^{(m-1)})}{\partial \mu_k} = \sum_j \tau_{kj}^{(m)} V^{-1}(x_j - \mu_k) - \lambda \sum_{j=1}^{K} \partial||\mu_j||_1$$

En annulant cette dérivée, on obtient :

$$\hat{\mu}_k^{(m)} = sign(\tilde{\mu}_k^{(m)})(|\tilde{\mu}_k^{(m)}| - \frac{\lambda}{\sum_{i=1}^n \tau_{ik}^{(m)}} V^{(m)} 1)_+$$

où $\tilde{\mu}_k^{(m)} = \frac{\sum_{i=1}^n \tau_{ik}^{(m)} x_i}{\sum_{i=1}^n \tau_{ik}^{(m)}}$ trouvé pour dans l'EM sans pénalité.

Tout comme pour l'algorithme EM sans pénalité, on alterne les étapes (E) et (M) jusqu'à convergence.

5.3 Illustration de l'algorithme EM pénalisé

5.3.1 Mélange gaussien dont les paramètres sont connus

On se place en dimension 2 et on considère le même mélange gaussien qu'on a décrit dans la partie EM non pénalisé. On rapple qu'on a un 500-échantillon avec 8 variables.On a construit nos donées de sorte qu'on ait $K = 5$. On a fait tourner l'algorithme EM pénalisé avec $\lambda = 1$ et on retrouve les résultats suivants :

Ce tableau donne les moyennes $\hat{\mu}_{kj}$ pour $j = 1, ..., 8$ et $l = 1, ..., 5$:

k,j	1	2	3	4	5
1	1,509	-1,432	-0,013	-0,053	0,046
2	-0,020	-0,020	-0,013	-1,465	1,456
3	1,470	-1,402	0,000	-0,083	0,068
4	1,480	-1,406	-0,015	-0,064	0,061
5	1,474	-1,418	-0,014	-0,041	0,055
6	-0,019	-0,015	0,000	-1,443	1,415
7	-0,024	-0,032	-0,004	-1,429	1,431
8	0,003	-0,021	-0,022	-1,456	1,432

Ce tableau contient les variances $\hat{\sigma}^2_{ij}$ où $i, j = 1, ..., 8$:

i,j	1	2	3	4	5	6	7	8
1	0,13	0,00	0,00	0,00	0,00	0,00	0,00	0,00
2	0,00	0,13	0,00	0,00	0,00	0,00	0,00	0,00
3	0,00	0,00	0,16	0,00	0,00	0,00	0,00	0,00
4	0,00	0,00	0,00	0,16	0,00	0,00	0,00	0,00
5	0,00	0,00	0,00	0,00	0,15	0,00	0,00	0,00
6	0,00	0,00	0,00	0,00	0,00	0,16	0,00	0,00
7	0,00	0,00	0,00	0,00	0,00	0,00	0,16	0,00
8	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,15

On retrouve ici les proportions $\hat{\pi}_k$ pour $k = 1, ..., 5$:

k	$\hat{\pi}_k$
1	0,197
2	0,205
3	0,190
4	0,199
5	0,208

Comme pour la partie EM non pénalisé, on a calculé la matrice de confusion suivante :

	1	2	3	4	5
1	98	0	0	0	0
2	0	96	6	1	0
3	2	4	84	1	0
4	0	0	2	98	0
5	0	0	8	0	96

Nous obtenons cette fois-ci un taux d'erreur de 5.9%, une précision de 98% pour les classes 1 et 4, de 96% pour les classes 2 et 5 et de 84% pour la classe 3.

5.3.2 Jeu de données DataMice

On introduit le jeu de données sur les souris, des données qui reviennent à Clara Higuera, au Department of Software Engineering and Artificial Intelligence.
Notre jeu de donées comporte 72 observations (lignes) et 905 variables (colonnes). Ainsi, l'ensemble de données se compose des niveaux d'expression de 68 protéines qui ont produit un signal détectable dans la fraction nucléaire du cortex pour un échantillon de 72 souris. Il y a 38 souris témoins et 34 souris trisomiques. Plusieurs mesures ont été enregistrées pour chaque protéine et pour chaque souris. Les mesures contenant des observations manquantes dans les données d'origine ont été supprimées, de sorte que l'on a entre 12 et 15 mesures par protéine et par souris.

Les 900 premières variables ont des noms de la forme Protein-X-Meas-Y où X est un entier entre 1 et 68 et Y un entier entre 12 et 15. Chacune de ces 900 variables représente le niveau d'expression de la protéine X à la mesure Y. S'en suivent les 5 dernières variables :

-"Genotype" : qui prend deux valeurs, soit "Control" (pour indiquer que c'est une souris témoin) soit "Ts65Dn" (pour indiquer qu'il s'agit d'une souris trisomique)

-"Treatment" : qui prend elle aussi deux valeurs, soit "Memantine" soit "Saline". Elle renseigne sur le type de traitement reçu par les souris.

-"Behaviour" : une fois de plus cette variable prend deux valeurs, soit "C/S" (si la souris est stimulée à apprendre) soit "S/C" (si la souris n'est pas stimulée à apprendre)

-"Class.mouse" : Elle peut prendre 8 valeurs de la forme **Genotype-Behaviour-Treatment** où Genotype vaut soit "c" (pour "control"), soit "t" (pour "Ts65Dn"), Behaviour vaut soit "CS" soit "SC", et Treatment vaut soit "m" (pour "Memantine") soit "s" (pour "saline")

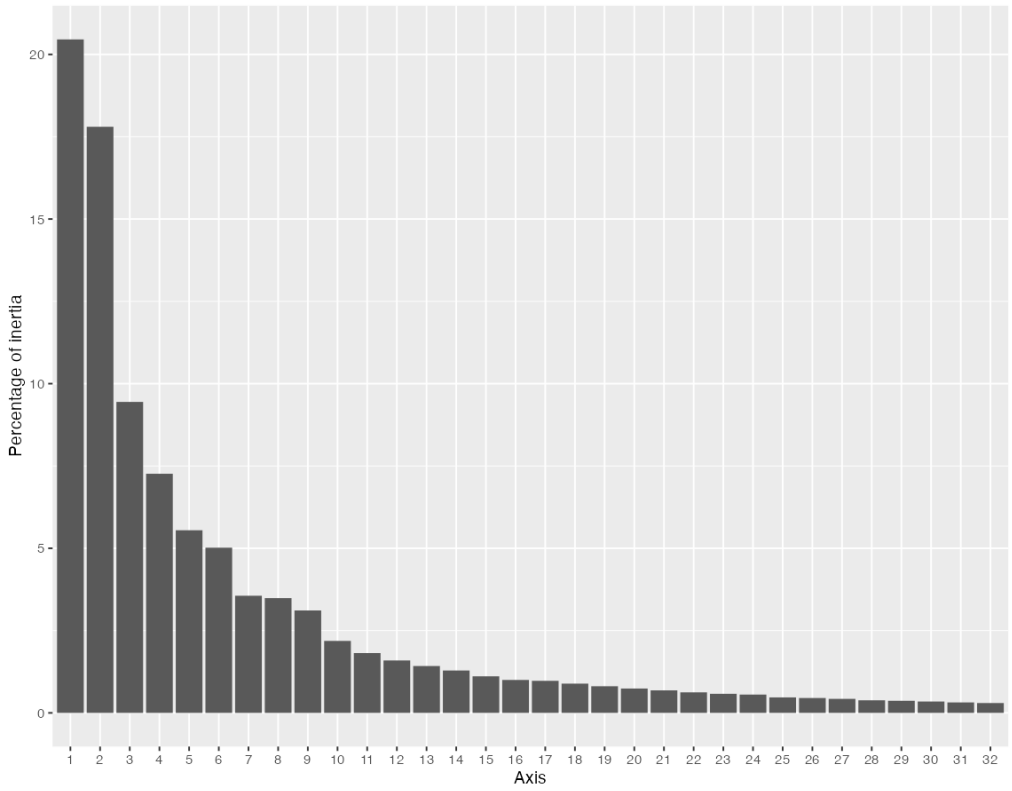
-"MouseID" : Cette variable joue le rôle d'identifiant pour chaque souris présente dans le dataframe.

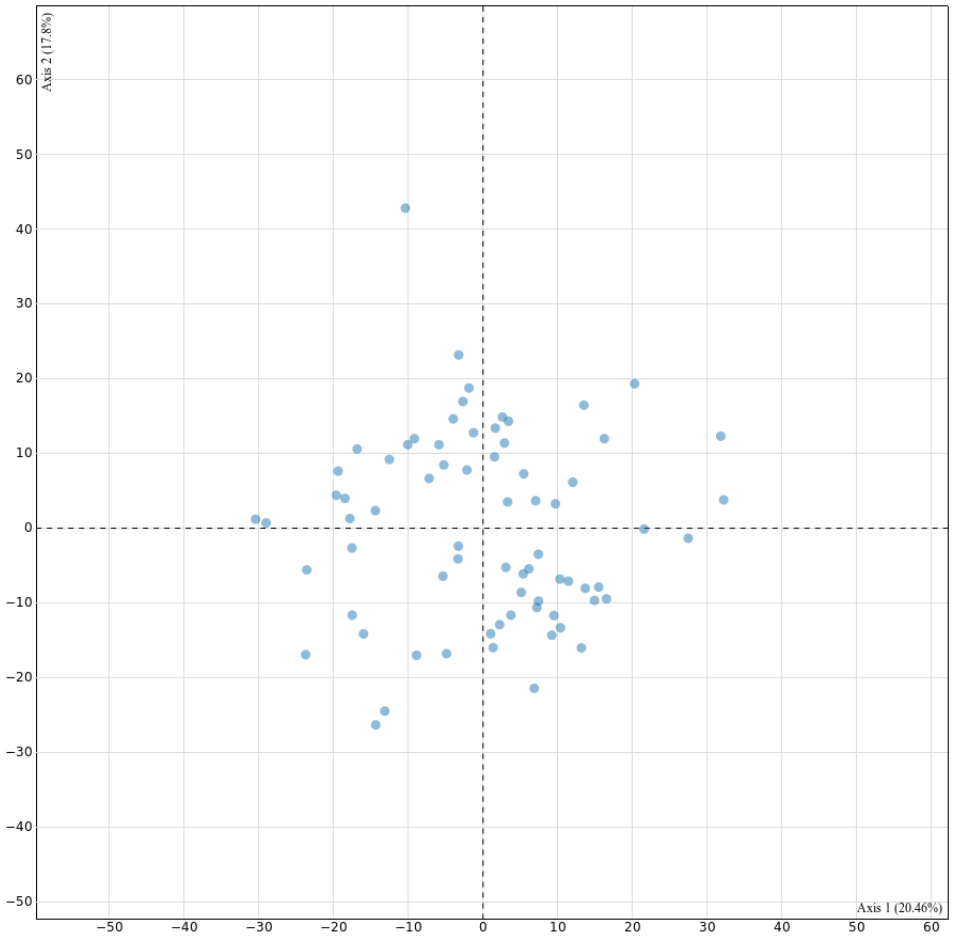
Pour les calculs suivants, nous décidons de ne charger que les variales de type numérique de notre jeu de données, c'est-à-dire les 900 premières.

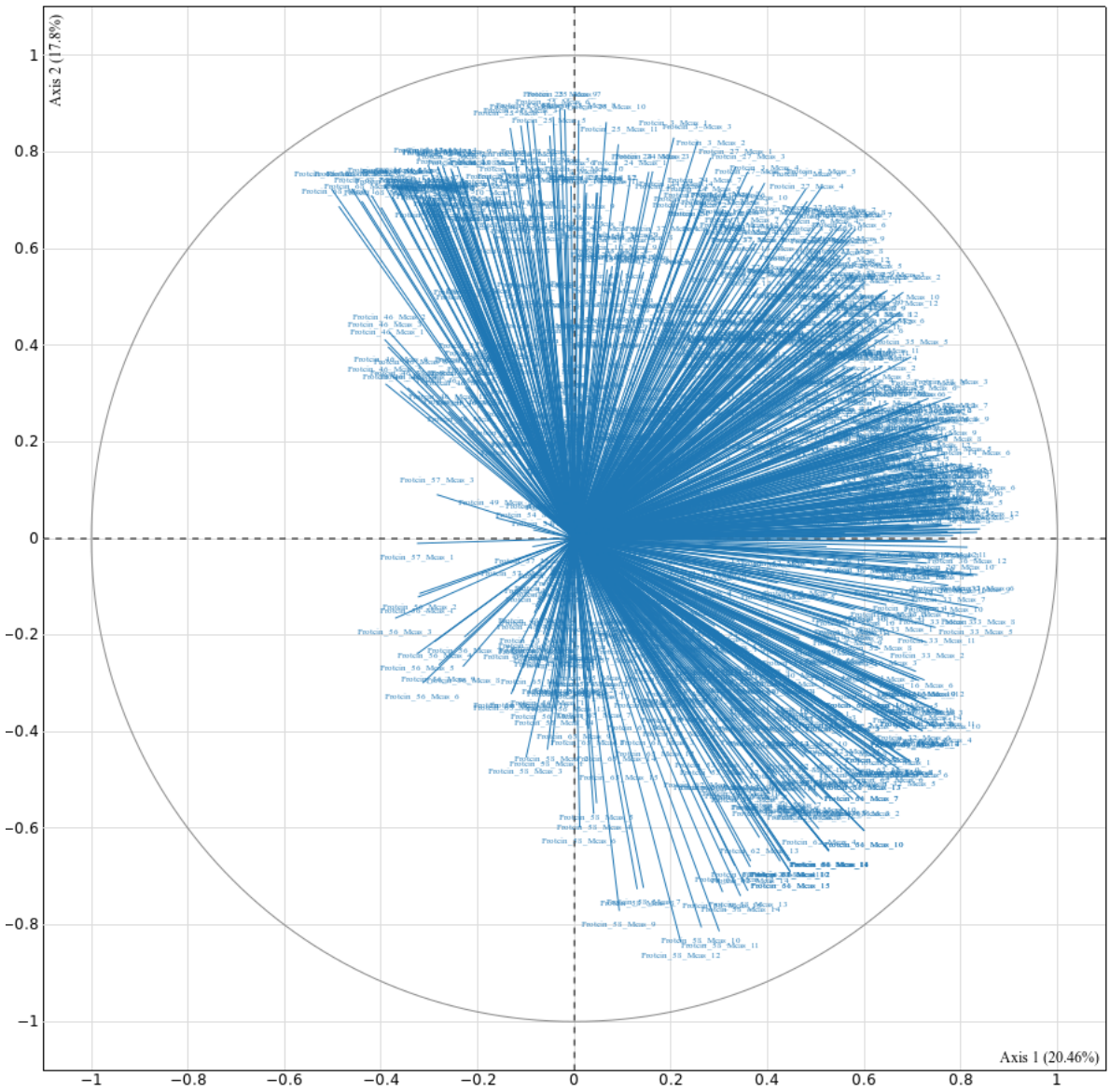
Nous avons désormais un jeu de données avec 72 observations et 900 variables. Cependant nous connaissons les limites des programmes que nous avons codé et $p = 900$ est un paramètre trop grand pour espérer avoir des résultats.

Pour ce faire, on fait une Analyse en Composantes Principales(ACP) sur nos données. On rappelle que l'ACP permet de transformer un ensemble de variables corrélées en un ensemble de variables non corrélées, appelées composantes principales. Elle vise à réduire la dimensionnalité des données en conservant le maximum d'informations possible. Et donc faire tourner l'EM pénalisé sur cette nouvelle matrice où les composantes principales sont les nouvelles variables.

L'ACP nous a fourni les graphiques suivants :


Percentage of inertia
0
5
10
15
20
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
Axis


Axis 2 (17.8%)
Axis 1 (20.46%)
60
50
40
30
20
10
0
−10
−20
−30
−40
−50
−50
−40
−30
−20
−10
0
10
20
30
40
50
60



Intéressons nous à présent uniquement au taux cumulé de variance expliquée par chaque variable : L'idée ici est de ne garder que le nombre p* de variables pour lequel au moins 95% de la variance est expliquée.

Nous observons que pour les 32 premières variables de notre jeu de données, plus de 95% de la variance est expliquée. Nous passons donc de 900 à 32 variables. Nous allons passer comme argument dans notre algorithme un dataframe de dimension (72, 32) constitué des 32 premières composantes principales du jeu

DataMice.

Fixons λ à 1. On peut voir d'après les tableaux ci-dessous que la structuration de nos clusters change selon la valeur K :

	Nombre de moyenne =0	Variables non informatives	Proportion
Cluster 1	11	1-9-11-16-20-21-23-24-27-28-32	0,93
Cluster 2	11	1-9-11-16-20-21-23-24-27-28-32	0,069

	Nombre de moyenne =0	Variables non informatives	Proportion
Cluster 1	7	4-5-6-10-12-23-31	0,86
Cluster 2	10	4-14-20-23-24-25-26-28-31-32	0,06
Cluster 3	12	2-4-5-7-10-15-18-19-23-24-29-31	0,08

	Nombre de moyenne =0	Variables non informatives	Proportion
Cluster 1	12	1-2-5-6-10-12-14-16-19-21-23-28	0,4
Cluster 2	19	1-2-3-4-6-7-8-10-12-14-15-16-17-21-22-23-26-27-28	0,03
Cluster 3	15	2-3-5-6-7-8-9-11-16-20-24-29-30-31-32	0,01
Cluster 4	20	2-4-5-6-7-10-11-13-15-16-17-18-19-23-24-26-27-28-30-32	0,56

	Nombre de moyenne =0	Variables non informatives	Proportion
Cluster 1	11	4-8-10-12-18-19-20-21-23-25-31	0,08
Cluster 2	13	2-8-9-10-14-15-17-20-21-23-24-27-28	0,10
Cluster 3	8	4-7-8-12-16-18-19-31	0,37
Cluster 4	13	6-7-8-12-15-16-21-22-23-24-29-30-31	0,39
Cluster 5	11	3-6-8-14-15-18-24-27-29-30-32	0,06

On regarde grâce au BIC quel est le K optimal. On a les valeurs suivantes :

K	BIC
2	-2337,65
3	-2470,49
4	-2471,94
5	-2661,07

Maintenant pour $K = 2$ fixé, on fait varier λ

λ		Nombre de moyennes = 0	Variables non informatives	Proportion
1	cluster 1	11	1-9-11-16-20-21-23-24-27-28-32	0,93
	cluster 2	11	1-9-11-16-20-21-23-24-27-28-32	0,07
2,5	cluster 1	27	1-2-4-6-8-9-10-11-12-13-14-15-16- 18-19-20-21-22-23-25-26-27-28-29-30-31-32	0,10
	cluster 2	27	1-2-4-6-8-9-10-11-12-13-14-15-16- 18-19-20-21-22-23-25-26-27-28-29-30-31-32	0,90
3,5	cluster 1	32	1-2-3-4-5-6-7-8-9-10-11-12-13-14 -15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32	0,72
	cluster 2	32	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32	0,28
3	cluster 1	32	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32	0,47
	cluster 2	32	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32	0,53
0,5	cluster 1	5	6-7-20-24-31	0,94
	cluster 2	5	6-7-20-24-31	0,05
1,5	cluster 1	13	5-6-7-11-16-18-20-21-22-24-30-31-32	0,94
	cluster 2	13	5-6-7-11-16-18-20-21-22-24-30-31-32	0,056
2	cluster 1	23	1-2-4-6-7-8-9-10-11-13-17-18-19-20-21-22-23-25-26-27-28-31-32	0,87
	cluster 2	23	1-2-4-6-7-8-9-10-11-13-17-18-19-20-21-22-23-25-26-27-28-31-32	0,12
6	cluster 1	32	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32	0,72
	cluster 2	32	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32	0,27

Regardons quel λ maximise le BIC :

λ	BIC
0.5	-2402,76
1	-2337,65
1.5	-2321,93
2	-2224,24
2.5	-2190,06
3	-2141,13
3.5	-2141,13
6	-2141,13

6 Deuxième méthode de sélection de variables

Dans cette partie, nous présentons une deuxième méthode de sélection de variables pour le clustering. Cette méthode a été établie par A.Raftery et N.Dean dans leur article "Variable Selection for Model Based Clustering"

Pour effectuer la sélection de variables, Raftery et Dean proposent un algorithme de recherche itératif reposant sur deux étapes que nous détaillerons dans la suite et qui teste à chaque itération si l'on gagne en rajoutant des variables ou en supprimant.

A la différence de la méthode de sélection de variables reposant sur l'algorithme EM pénalisé, celle-ci détermine automatiquement le nombre de clusters. Un deuxième avantage à cette méthode est que l'utilisateur de l'algorithme que nous présenterons aura seulement besoin de choisir le nombre de cluster maximal. En revanche, elle présente un coût algorithmique plus élevé.

6.1 Cadre d'étude

Pour résoudre le problème de sélection de variable, on introduit l'ensemble de données X et au cours de la sélection, on partitionne les données dans 3 ensembles définis comme suit :

- $X^{(1)}$ l'ensemble des variables déjà sélectionnées
- $X^{(2)}$ les variables en cours d'inclusion ou exclusion de l'ensemble des variables de regroupement
- $X^{(3)}$ l'ensemble des variables restantes.

La décision d'inclure ou d'exclure $X^{(2)}$ de l'ensemble des variables de clustering est reformulée selon la comparaison de deux modèles suivants pour l'ensemble des données :

$$\begin{aligned} M_1 : \mathbb{P}(X|z) &= \mathbb{P}(X^{(1)}, X^{(2)}, X^{(3)}|z) \\ &= \mathbb{P}(X^{(3)}|X^{(2)}, X^{(1)})\mathbb{P}(X^{(2)}|X^{(1)})\mathbb{P}(X^{(1)}|z) \end{aligned}$$

$$\begin{aligned} M_2 : \mathbb{P}(X|z) &= \mathbb{P}(X^{(1)}, X^{(2)}, X^{(3)}|z) \\ &= \mathbb{P}(X^{(3)}|X^{(2)}, X^{(1)})\mathbb{P}(X^{(2)}, X^{(1)}|z). \end{aligned}$$

où z correspond aux variables latentes introduites dans la section "Algorithme Espérance-Maximisation". Le modèle M_1 dit que $X^{(2)}$ ne donne aucune information supplémentaire pour la création des clusters tandis que le modèle M_2 dit que $X^{(2)}$ fournit des informations supplémentaires après que $X^{(1)}$ a été observé.

On compare les modèles M_1 et M_2 à l'aide d' une approximation du facteur de Bayes B_{12} pour M_1 contre M_2 :

$$B_{12} = \frac{\mathbb{P}(X|M_1)}{\mathbb{P}(X|M_2)}.$$

Si le facteur de Bayes est supérieur à 1, alors le modèle M_1 est plus significatif que M_2. Si le Bayes factor est inférieur à 1, le modèle M_2 est plus évident pour l'ensemble des données observées.

Il est important de noter que l'interprétation du facteur de Bayes dépend des probabilités à priori des modèles. En effet on a :

$$\mathbb{P}(X|M_k) = \int \mathbb{P}(X|\theta_k, M_k)\mathbb{P}(\theta_k|M_k)d\theta_k.$$

où θ_k correspond au vecteur de paramètres et $\mathbb{P}(\theta_k|M_k)$ la probabiblité à priori du modèle, et $k = 1, 2$.

Pour le modèle M1, on a :

$$\begin{aligned}\mathbb{P}(X|M_1) &= \int \mathbb{P}(X|\theta_1, M_1)\mathbb{P}(\theta_1|M_1)d\theta_1 \\ &= \int \mathbb{P}(X^{(3)}|X^{(2)}, X^{(1)}, \theta_1, M_1)\mathbb{P}(X^{(2)}|X^{(1)}, \theta_1, M_1)\mathbb{P}(X^{(1)}|\theta_1, M_1)\mathbb{P}(\theta_1|M_1)d\theta_1\end{aligned}$$

On note $\theta_{13}, \theta_{12}, \theta_{11}$ les paramètres respectifs des distributations respectifs $(X^{(3)}|X^{(2)}, X^{(1)}, \theta_1, M_1)$, $(X^{(2)}|X^{(1)}, \theta_1, M_1)$, $(X^{(1)}|\theta_1, M_1)$. Et on admet que leurs distributions à priori sont indépendantes. Autrement dit :

$$\begin{aligned}\mathbb{P}(\theta_1|M_1) &= \mathbb{P}(\theta_{11}, \theta_{12}, \theta_{13}|M_1) \\ &= \mathbb{P}(\theta_{11}|M_1)\mathbb{P}(\theta_{12}|M_1)\mathbb{P}(\theta_{13}|M_1)\end{aligned}$$

D'où :

$$\begin{aligned}\mathbb{P}(X|M_1) = \int \mathbb{P}(X^{(3)}|X^{(2)}, X^{(1)}, \theta_{13}, M_1)\mathbb{P}(\theta_{13}|M_1)\mathbb{P}(X^{(2)}|X^{(1)}, \theta_{12}, M_1)\mathbb{P}(\theta_{12}|M_1) \\ \mathbb{P}(X^{(1)}|\theta_{11}, M_1)\mathbb{P}(\theta_{11}|M_1)d\theta_{13}d\theta_{12}d\theta_{11} \\ = \mathbb{P}(X^{(3)}|X^{(2)}, X^{(1)}, M_1)\mathbb{P}(X^{(2)}|X^{(1)}, M_1)\mathbb{P}(X^{(1)}|M_1)\end{aligned}$$

De la même manière on obtient :

$$\mathbb{P}(X|M_2) = \mathbb{P}(X^{(3)}|X^{(2)}, X^{(1)}, M_2)\mathbb{P}(X^{(2)}, X^{(1)}|M_2)$$

La distribution à priori du paramètre θ_{13} est supposé être le même sous M_1 et M_2 donc $\mathbb{P}(X^{(3)}|X^{(2)}, X^{(1)}, M_1) = \mathbb{P}(X^{(3)}|X^{(2)}, X^{(1)}, M_2)$, il s'ensuit alors que :

$$B_{12} = \frac{\mathbb{P}(X^{(2)}|X^{(1)}, M_1)\mathbb{P}(X^{(1)}|M_1)}{\mathbb{P}(X^{(2)}, X^{(1)}|M_2)}.$$

Cependant cette quantité est difficile à calculer, on cherche alors à l'approximer en utilisant le BIC.

$$\log(B_{12}) \approx \log(\mathbb{P}(X^{(2)}|X^{(1)}, M_1)) + \log\left(\mathbb{P}(X^{(1)}|M_1)\right) - \log(\mathbb{P}(X^{(2)}, X^{(1)}|M_2))$$

Notations :

— $-2\log(B_{12}) = BIC_{diff}$

— $2\log(\mathbb{P}(X^{(2)}, X^{(1)}|M_2)) = BIC_{clust}$

— $2\log(\mathbb{P}(X^{(2)}|X^{(1)}, M_1)) + 2\log(\mathbb{P}(X^{(1)}|M_1)) = BIC_{notclust}$

Lorsque $BIC_{diff} > 0$, ie $B_{12} < 1$ alors cela signifie que le modèle M_2 est plus significatif que le modèle M_1, donc les variables de $X^{(2)}$ sont pertinentes donc on les garde.

6.2 Méthode

Dans cette partie nous présentons en détails le fonctionnement de cet algorithme de sélection de variables dans le cas où le mélange considéré est gaussien et satisfait les hypothèses définies précédemment sur la matrice de variance-covariance. Nous utiliserons les conventions employées dans l'article de Adrian E Raftery et Nema Dean, et considèrerons que le BIC correspond au $-BIC$ habituel dont nous avons rappelé la construction dans la partie "Sélection de modèles". Plus précisément, $BIC = 2 \times \log$ (vraisemblance maximisée)-(nombre de paramètres)$\times \log(n)$.

Etape 0 :
Tout d'abord, l'utilisateur fixe K_{max} soit le nombre de clusters maximal considéré pour l'ensemble de données considéré.

Etape 1 : Sélection de la première variable
On sélectionne la première variable dans l'ensemble de données considéré présentant le plus de preuves de clustering univarié.
Plus précisément, on découpe notre ensemble $X^{(3)} = X$ en chacune de ses variables que l'on considère séparément et on sélectionne la variable x^j maximisant BIC_{diff}, soit la plus grande différence entre BIC_{clust} et $BIC_{notclust}$, où

$$BIC_{clust} = \max_{2 \leq K \leq K_{max}} BIC_K(x^{(j)})$$

et,

$BIC_K = 2\times$log(vraisemblance maximisée) $-(K + p - 1 + K \times p) \times \log(n)$
avec K le nombre de clusters, p le nombre de variables, n le nombre de données.
Et $BIC_{notclust} = BIC_{reg}$ où

$$BIC_{reg} = -n\log(2\pi) - n\log(\frac{RSS}{n}) - n - 2 \times \log(n)$$

A cette étape la somme des carrés résiduelles de la régression ($RSS = \sum_{i=1}^{n}(x_i^{(2)} - \hat{a}x_i^{(1)} - \hat{b})^2$) de x^j où $j \in \{1, ..., D_1\}$ avec D_1 le nombre de variables composant $X^{(3)}$ (dans notre cas $D_1 = p$), sur une constante.
Enfin on choisit la meilleure variable x^{j_1} telle que

$$j_1 = \arg\max_{j : x^j \in X} (BIC_{diff}(x^j))$$

On pose alors $X^{(1)} := x^{j_1}$ et $X^{(3)} = X \backslash x^{j_1}$

Etape 2 : Sélection de la deuxième variable

On sélectionne ensuite une seconde variable à ajouter à l'ensemble $X^{(1)}$. Celle-ci est choisie de sorte qu'elle maximise la différence entre le BIC_{clust} pour cette variable et celle sélectionnée à l'étape précéndente, et la somme du BIC_{clust} de la variable choisie à l'étape 1 et le BIC_{reg} pour la régression de x^j sur $X^{(1)}$.

Plus particulièrement, on considère chaque variable composant $X^{(3)}$ séparément et on calcule BIC_{diff} de façon suivante, pour $j \in \{1, ..., D_2\}$ où D_2 correspond au nombre de variables contenues dans $X^{(3)}$ (dans notre cas, $D_2 = p - 1$:

$$BIC_{diff}(x^j) = BIC_{clust}(x^j) - BIC_{notclust}(x^j),$$

où,

$$BIC_{clust}(x^j) = \max_{2 \leq K \leq K_{max}} (BIC_K(X^{(1)}, x^j))$$

et

$$BIC_{notclust} = BIC_{reg} + BIC_{clust}(X^{(1)})$$

avec

$$BIC_{reg} = -n \log(2\pi) - n \log(\frac{RSS}{n}) - n - (\dim(X^{(1)} + 2) \log(n)$$

Dans ce cas RSS correspont à la somme des carrés résiduelles de la régression linéaire de x^j avec la variable sélectionnée à l'étape précédente, et $\dim(X^{(1)}) = 1$. Comme à l'étape 1 on choisit notre deuxième variable x^{j_2} telle que

$$x_2 = \arg \max_{j : x^j \in X^{(3)}} (BIC_{diff}(x^j))$$

On ajoute ensuite cette variable à l'ensemble $X^{(1)}$ et on la supprime de l'ensemble $X^{(3)}$. C'est-à-dire,

$$X^{(1)} = X^{(1)} \cup x^{j_2}$$

et

$$X^{(3)} = X^{(3)} \backslash x^{j_2}.$$

Une fois ces deux étapes mises en place, on alterne les deux suivantes qui testent si l'on peut exclure des variables précédemment sélectionnées ou ajouter de nouvelles pertinentes pour le clustering. Ces deux étapes sont itérées jusqu'à ce que les étapes d'inclusion et d'exclusion soient consécutivement rejetées.

Etape 3.1 : Exclusion de variables

Parmi les variables sélectionnées, on peut choisir d'en exclure certaines si elles apportent peu d'informations pour le clustering.

En particulier, à l'étapet t, on considère séparément les variables composant $X^{(1)}$ et on choisit d'exclure celle qui minimise BIC_{diff}, où

$$BIC_{diff}(x^j) = BIC_{clust} - BIC_{notclust}(x^j), \forall j \in 1, ..., D_t$$

avec D_t le nombre de variables composant $X^{(1)}$ et où,

$$BIC_{clust} = \max_{2 \leq K \leq K_{max}} (BIC_K(X^{(1)}))$$

et,

$$BIC_{notclust} = BIC_{reg} + BIC_{clust}(X^{(1)} \backslash x^j)$$

avec,

$$BIC_{reg} = -n \log(2\pi) - n \log(\frac{RSS}{n}) - n - (\dim(X^{(1)}) + 2) \log(n)$$

Dans notre cas, le RSS correspond à la somme des carrés résiduels de la régression linéaire de x^j et de $X^{(1)} \backslash x^j \ \forall j \in \{1, ..., D_t\}$ où D_t correspond au nombre de variables contenues dans $X^{(1)}$, et $\dim(X^{(1)})$ = nombre de variables sélectionnées - 1. D'autre part, $BIC_{clust}(X^{(1)} \backslash x^j)$ est le BIC pour le clustering de toutes les variables sélectionnées privées de la variable x^j considérée.

On choisit d'exclure la variable x^{j_t} lorsque

$$j_t = \arg \min_{j : x^j \in X^{(1)}} (BIC_{diff}(x^j)),$$

et que

$$BIC_{diff}(x^{j_t}) \leq 0.$$

Si cette condition est remplie on supprime cette variable de $X^{(1)}$ et on l'ajoute à $X^{(3)}$. C'est-à-dire :

$$X^{(1)} = X(1) \backslash x^{j_t}$$

et,

$$X^{(3)} = X^{(3)} \cup x^{j_t}$$

Sinon ces deux ensembles restent inchangés.

Etape 3.2 : Inclusion de variables

Parmi les variables restantes dans $X^{(3)}$ on peut décider d'en inclure certaines si elles apportent de l'information sur la formation des groupes.
Pour cela à l'étape t+1, on considère séparément les variables de $X^{(3)}$ et on choisit la variable qui maximise la différence entre le BIC pour le clustering des variables précédement sélectionnées et celle considérée, et la somme du BIC pour le clustering des variables contenues dans $X^{(1)}$ et le BIC pour la régression linéaire de la variable x^j considérée sur les variables contenues dans $X^{(1)}$. Plus précisément on calcule pour chaque $j \in \{1, ..., D_{t+1}\}$ où D_{t+1} correspond à la taille de $X^{(3)}$, la différence suivante :

$$BIC_{diff} = BIC_{clust}(x^j) - BIC_{notclust}(x^j)$$

avec,

$$BIC_{clust}(x^j) = \max_{2 \leq K \leq K_{max}} (BIC_K(X^{(1)}, x^j))$$

et,

$$BIC_{notclust} = BIC_{reg} + BIC_{clust}(X^{(1)})$$

avec,

$$BIC_{reg} = -n \log(2\pi) - n \log(\frac{RSS}{n}) - n - (\dim(X^{(1)}) + 2) \log(n)$$

Dans notre cas, le RSS correspond à la somme des carrés résiduels de la régression linéaire de x^j sur les variables contenues dans $X^{(3)}$ $\forall j \in \{1, ..., D_{t+1}\}$, et $\dim(X^{(1)})$ = nombre de variables composant $X^{(1)}$. D'autre part, $BIC_{clust}(X^{(1)})$ est le BIC por le clustering avec les variables contenues dans $X^{(1)}$.

On choisit d'inclure la variable $x^{j_{t+1}}$ telle que

$$j_{t+1} = \arg \max_{j : x^j \in X^{(3)}} (BIC_{diff}(x^j))$$

et que la condition suivante est vérifiée :

$$BIC_{diff}(x^{j_{t+1}}) > 0$$

Dans ce cas, on ajoute $x^{j_{t+1}}$ à l'ensemble $X^{(1)}$ et on l'enlève de $X^{(3)}$. C'est-à-dire,

$$X^{(1)} = X^{(1)} \cup x^{j_{t+1}}$$

et

$$X^{(3)} = X^{(3)} \backslash x^{j_{t+1}}$$

Si cette condition n'est pas vérifiée, les deux ensembles restent inchangés.

Résultats :

Une fois que les deux étapes précédentes restent consécutivement inchangées, notre algorithme renvoie $X^{(1)}$ l'ensemble des variables pertinentes pour le clustering ainsi que K, le nombre de clusters.

Remarque :

A chaque fois que l'on calcule BIC_{clust} il est nécessaire d'obtenir l'estimateur du maximum de vraisemblance de notre mélange. Dans notre code, on utilise l'algorithme EM que nous avons présenté à la partie précédente pour obtenir ces paramètres.

6.3 Illustration de l'algorithme de sélection de variables

On revient sur notre mélange gaussien où $X \in \mathbb{R}^{500\times 8}$. On rappelle qu'on est sur une dimension 2 avec 5 clusters. Nos données sont représentées par le graphique ci-dessous :

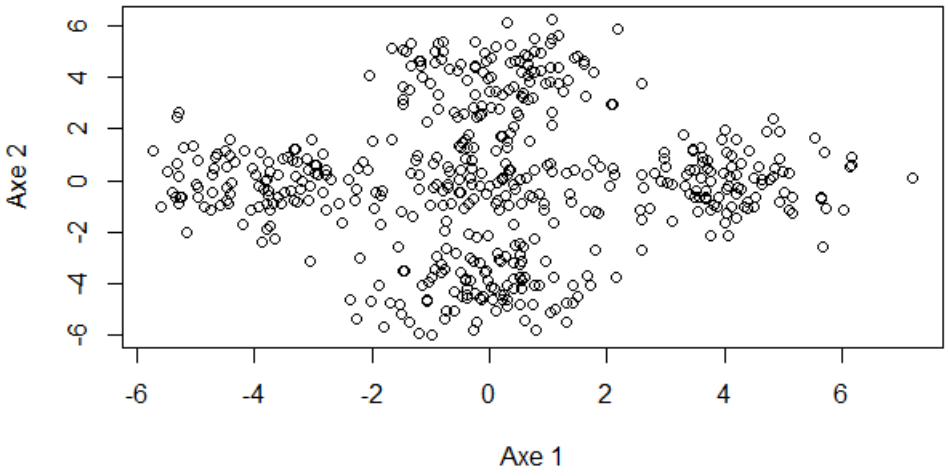


FIGURE 6 – Mélange gaussien en dimension 2

Une seule variable est sélectionée pour le clustering. Nous nous attendions à ce que les variables X^1 et X^2 soient celles sélectionnées (ou alors plus précisément une variable parmi X^1, X^3, X^4, X^5 et une variable parmi X^2, X^6, X^7, X^8) . Ceci nous permet de tirer plusieurs conclusions :
-Notre programme algorithme n'est peut-être pas bien codé
-Notre système de variance (même matrice de variance pour tous les clusters) n'est peut-être pas adaptée pour de la sélection de variable.

7 Annexe

7.1 Annexe A

On a vu que :

— $\theta_i^* = \underset{\theta_i \in \Theta_i}{\text{argmax}} L_n(\theta_i)$

— $A_{\theta_i^*} = -[\frac{\partial^2 L_n(\theta_i^*)}{\partial \theta_i^j \partial \theta_i^l}]_{j,l}$ où θ_i^j est la j-ième composante du vecteur des paramètres θ_i et $A_{\theta_i^*}$ est l'opposé de la matrice Hessienne des dérivées secondes partielles de la fonction $L_n(\theta_i)$ en θ_i^*.

On veut alors montrer qu'on peut remplacer :

— θ_i^* par l'EMV $\hat{\theta}_i$

$$\hat{\theta}_i = \underset{\theta_i \in \Theta_i}{\text{argmax}} \frac{1}{n} g_{M_i}(Y, \theta_i)$$

— et $A_{\theta_i^*}$ par $I_{\hat{\theta}_i}$ la matrice d'information de Fisher pour une observation qui est définie par

$$I_{\hat{\theta}_i} = -\mathbb{E}([\frac{\partial^2 log(g_{M_i}(Y_i, \hat{\theta}_i))}{\partial \theta_i^j \partial \theta_i^l}])$$

Démonstration. On cherche à montrer que

$$\sqrt{n}(\theta^* - \hat{\theta}) = \mathcal{O}(1)$$

On peut écrire : $\sqrt{n}(\hat{\theta} - \theta_0) + \sqrt{n}(\theta_0 - \theta^*)$ où θ_0 est l'unique maximum de $\mathbb{E}[log(g_M(Y_1, \theta))]$.

On sait que sous des conditions de régularités, l'estimateur du maximum de vraisemblance $\hat{\theta}$ vérifie :

$$\sqrt{n}(\hat{\theta} - \theta_0) \xrightarrow[n \to \infty]{} \mathcal{N}(\mu,\, I_{\theta_0}^{-1})$$

donc :

$$\sqrt{n}(\hat{\theta} - \theta_0) = \mathcal{O}(1)$$

Pour le second terme, posons :

$$LG_n(\theta) = \frac{1}{n} \sum_{k=1}^{n} log(g_{M_i}(Y_k, \theta_i))$$

et

$$B_\theta = \frac{log(P(\theta_i | M_i))}{n}$$

Alors $L_n(\theta) = LG_n(\theta) + B_n(\theta)$. On a : $L'_n(\theta) = LG'_n(\theta) + B'_n(\theta)$

$$LG'_n(\theta) = \frac{1}{n} \sum_{k=1}^{n} \frac{\partial log(g_{M_i}(Y_k, \theta_i))}{\partial \theta}$$

Sous la condition que $\mathbb{E}[|\frac{\partial log(g_M(Y_1,\theta))}{\partial\theta}|] \leq \infty$, on a par la loi faible des grands nombres, la convergence en probabilité de LG'_n vers $\mathbb{E}[\frac{\partial log(g_M(Y_1,\theta))}{\partial\theta}]$. De plus, $B'_n(\theta) \xrightarrow[n\to\infty]{} 0$ ps. On obtient alors la convergence en proba de $L'_n(\theta)$ vers $\mathbb{E}[\frac{\partial log(g_M(Y_1,\theta))}{\partial\theta}]$.

D'après le **lemme 5.10 dans Van Der VAART 1998** : Soit f_n une suite de fonctions aléatoires et f une fonction fixe de θ telle que $f_n(\theta) \xrightarrow[n\to\infty]{} f(\theta)\forall\theta$ en probabilité $\forall\theta$. Supposons que chaque $\theta \mapsto f_n$ est continue et a exactement un zéro $\hat{\theta}_n$. Soit θ_0 un point tel que $f(\theta_0-\epsilon) < 0 < f(\theta_0+\epsilon)\ \forall\epsilon$. Alors $\hat{\theta} \xrightarrow[n\to\infty]{} \theta_0$.

Dans notre cas $f_n(\theta) = L'_n(\theta)$ et $f(\theta) = \mathbb{E}[\frac{\partial log(g_M(Y_1,\theta))}{\partial\theta}]$. On sait que L'_n s'annule uniquement en θ^* et $L'_n(\theta^*) = \mathcal{O}(1)$ donc $\hat{\theta}_n = \theta^*$. De plus $f(\theta_0-\epsilon) < 0 < f(\theta_0+\epsilon)$ car $f(\theta_0) = 0$ par définition de θ_0 et c'est d'ailleurs l'unique point par unicité de l'agrmax. Donc par le lemme ci-dessus :

$$\theta^* \xrightarrow[n\to\infty]{} \theta_0.$$

Ce qui conclut qu'on peut alors approximer θ^* par $\hat{\theta}$.

Ensuite, il s'agit de démontrer que $\sqrt{n}(A_{\theta^*} - I_{\hat{\theta}})$ est aussi bornée en probabilité.

On sait que :

$$A_\theta - I_\theta = LG''_n(\theta) - I_\theta + \frac{1}{n}[log(P(\theta))]''$$

par l'expression de A_θ de LG_n et de B_n. Et on sait également que $\frac{1}{n}[log(P(\theta))]'' \xrightarrow[n\to\infty]{}$ 0 ps.

D'un côté on a aussi en notant que $\mathbb{E}[LG''_n(\theta)] = I_\theta$ et sous condition que $\mathbb{E}[||\frac{\partial^2 log(g_M(Y_1,\theta))}{\partial\theta_i^j\partial\theta_i^l}||^2] \leq \infty$, alors par le thèorème central limite, on a la convergence en loi de $\sqrt{n}(LG''_n(\theta) - I_\theta)$:

$$LG''_n(\theta) - I_\theta = \mathcal{O}\Big(n^{-\frac{1}{2}}\Big)$$

Donc on a : $\sqrt{n}(A_\theta - I_\theta) = \mathcal{O}(1)\ \forall\theta(*)$

On a vu ci-dessus que $\theta^* = \hat{\theta} + \mathcal{O}\Big(n^{\frac{-1}{2}}\Big)$ $(**)$

On applique un développement de Taylor A_{θ^*} autour de $A_{\hat{\theta}}$ car $(**)$ dit que θ^* est aussi proche de $\hat{\theta}$ quand n est grand. On a alors

$$\sqrt{n}(A_{\theta^*} - A_{\hat{\theta}}) = \sqrt{n}(\theta^* - \hat{\theta})A'_{\hat{\theta}} + \mathcal{o}\Big(\sqrt{n}(\theta^* - \hat{\theta})^2\Big)$$

$$\sqrt{n}(A_{\theta^*} - I_{\hat{\theta}}) = \sqrt{n}(A_{\hat{\theta}} - I_{\hat{\theta}}) + \sqrt{n}(\theta^* - \hat{\theta})A'_{\hat{\theta}} + o\Big(\sqrt{n}(\theta^* - \hat{\theta})^2\Big)$$

(*) reste vrai pour $\theta = \hat{\theta}$ donc le premier terme est bornée en proba. Ensuite, on remarque que $A'_\theta = L'''_n(\hat{\theta})$ et on rappelle que cette quantité doit être bornée en n afin d'obtenir l'ordre $\frac{1}{n}$ dans l'appoximation de Laplace. Donc le second terme est aussi bornée en proba. On a donc ce qu'on cherchait :

$$\sqrt{n}(A_{\theta^*} - I_{\hat{\theta}}) = \mathcal{O}(1)$$

□

7.2 Annexe B

Il s'agit de montre que :

$$BIC_{reg} = -n\log(2\pi) - n\log(\frac{RSS}{n}) - n - (\dim(Y^{(1)} + 2)\log(n)$$

On est ici dans un cas de mélange gaussien. On considère alors que pour la sélection de variables $Y^{(2)}$ est une régression linéaire de $Y^{(1)}$ avec un intercept. $\forall i = 1, ..., n$

$$Y_i^{(2)}|aY_i^{(1)} + b \sim \mathcal{N}(aY_i^{(1)} + b,\, \sigma^2)$$

où $\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^n (y_i^{(2)} - ay_i^{(1)} - b)$ est un estimateur de σ^2

La vraisemblance s'écrit alors :

$$L_n(Y^{(2)}, a, b) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\hat{\sigma}^2}} \exp\big(\frac{(-(y_i^{(2)} - ay_i^{(1)} - b)^2)}{2\hat{\sigma}^2}\big)$$

$$\begin{aligned} l_n(Y^{(2)}, a, b) &= \log \mathbb{P}(Y^{(2)}|Y^{(1)}, M_1) \\ &= -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\hat{\sigma}^2) - \frac{1}{2\hat{\sigma}^2}\sum_{i=1}^n (y_i^{(2)} - ay_i^{(1)} - b)^2 \\ &= -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\frac{RSS}{n}) - \frac{n}{2} \end{aligned}$$

$$2l_n(Y^{(2)}, a, b) = 2\log \mathbb{P}(Y^{(2)}|Y^{(1)}, M_1) = -n\log(2\pi) - n\log(\frac{RSS}{n}) - n$$

Or on a vu par la définition du BIC que $BIC_{reg} = 2l_n(Y^{(2)}, a, b) - \log(n)(dim(Y^{(1)}) + 2)$

Remarque : $dim(Y^{(1)}) + 2$ représente la dimension du modèle , on ajoute le terme "+2" car on a 2 paramètres a et b à estimer dans le modèle.

D'où :

$$2\log \mathbb{P}(Y^{(2)}|Y^{(1)}, M_1) \approx BIC_{reg} = -n\log(2\pi) - n\log(\frac{RSS}{n}) - n - (dim(Y^{(1)}) + 2)\log(n).$$

8 Réferences

— Journal of Machine Learning Research 8 (2007) 1145-1164 : "Penalized Model-Based Clustering with Application to Variable Selection" by Wei Pan and Xiaotong Shen

— Journal of the American Statistical Association : "Variable Selection for Model-Based Clustering" by Adrian E Raftery Nema Dean

— Emilie Lebarbier, Tristan Mary-Huard. Le critère BIC : fondements théoriques et interprétation. [Rapport Technique] RR-5315, INRIA. 2004, pp.17. inria-00070685

— AM 221 : Advanced Optimization by Prof. Yaron Singer

— Book "Asymptotic Statistics" by Van Der Vaart, 199

— Algorithme EM et modèles de mélange, by Madalina Olteanu (cours de M1 MAEF 2019-2020)