Rapport de projet

Approche bayésienne de clustering sur le jeu de données des galaxies.

Nino Scheele, Bastien Maudry
Tuteur : Martin Metodiev

Département IMDS
Polytech Clermont
France
2024 - 2025

Résumé

Ce projet explore une approche bayésienne du clustering en appliquant un modèle de mélange gaussien. Nous utilisons un échantillonneur de Gibbs pour estimer les paramètres du modèle. Nous abordons aussi le problème du label switching. Enfin, nous déterminons le nombre optimal de clusters en utilisant des critères bayésiens, afin de garantir une classification des données plus rigoureuse et adaptée. Le jeu de données utilisé pour ce projet est celui des galaxies, jeu de données fréquemment utilisé dans la littérature scientifique.

Mots-clés

clustering, approche bayésienne, mélanges gaussiens, échantillonneur de Gibbs, label switching, galaxies

Abstract

This project explores a Bayesian approach to clustering by applying a Gaussian mixture model. We use a Gibbs sampler to estimate the model parameters and address the issue of label switching. Finally, we determine the optimal number of clusters using Bayesian criteria to ensure a more rigorous and adapted data classification. The dataset used for this project is the galaxies dataset, which is frequently referenced in the scientific literature.

Key words

clustering, Bayesian approach, Gaussian mixtures, Gibbs sampler, label switching, galaxies

Remerciements

Nous tenons à remercier Martin Metodiev pour son encadrement, sa bienveillance et sa pédagogie tout au long de ce projet. Cela a été une expérience très formatrice, et nous lui en sommes reconnaissants.

Table des matières

1 Définition des notions **2**
- 1.1 Cadre du projet . . . 2
- 1.2 Explication du sujet . . . 2
  - 1.2.1 Méthode bayésienne . . . 2
  - 1.2.2 Principe du clustering . . . 3
  - 1.2.3 Présentation du jeu de données . . . 4

2 Problème avec un nombre de clusters fixé **5**
- 2.1 Formalisation du problème . . . 5
- 2.2 Les a priori . . . 6
  - 2.2.1 Théorie . . . 6
  - 2.2.2 Pratique . . . 7
- 2.3 Gibbs sampling . . . 10
  - 2.3.1 Théorie . . . 10
  - 2.3.2 Application . . . 12
- 2.4 Label Switching . . . 14
  - 2.4.1 Théorie . . . 14
  - 2.4.2 Pratique . . . 15
- 2.5 Calcul du MAP et résultats finaux . . . 16
  - 2.5.1 MAP . . . 16
  - 2.5.2 Résultats . . . 17

3 Détermination du nombre K de clusters et comparaison des résultats **19**
- 3.1 Estimation de K avec les critères BIC et ICL . . . 19
- 3.2 Comparaisons . . . 23
  - 3.2.1 Algoritme EM . . . 23
  - 3.2.2 Papiers scientifiques . . . 26

Table des figures

2.1 Répartition des galaxies 8
2.2 Répartition des galaxies 8
2.3 Répartition des galaxies 9
2.4 Répartition des données générées suivant nos a priori 10
2.5 Graphe des valeurs des paramètres pour K=3 (Dans l'ordre π, σ^2, μ) 14
2.6 Résultat pour μ à la sortie de l'échantillonneur de Gibbs avec 5 groupes 14
2.7 Résultat pour μ à la sortie du label switching avec 5 groupes 16
2.8 Visualisation du clustering : répartition des galaxies, densités et groupes d'appartenance 17
2.9 Visualisation du clustering : répartition des galaxies, densités et groupes d'appartenance avec le package bayesmix 18

3.1 Tableau récapitulatif de la répartition des individus avec K = 10 21
3.2 Valeurs du BIC et de l'ICL pour les différents nombre de groupes 21
3.3 Visualisation finale avec les groupes d'appartenance et la densité de la mixture gaussienne pour K = 3 21
3.4 Visualisation finale avec les groupes d'appartenance et la densité de la mixture gaussienne pour K = 4 22
3.5 Visualisation finale avec les groupes d'appartenance et la densité de la mixture gaussienne pour K = 5 22
3.6 Clusters assignés aux observations par la méthode mclust sur R 24
3.7 Visualisation des clusters donnés par l'algorithme EM avec leurs densités . . . 25

Introduction

On estime que 2,5 exaoctets de données sont générées chaque jour, un chiffre en constante augmentation[1]. Dans un monde de plus en plus dominé par la collecte massive de données, le besoin d'outils efficaces pour analyser, segmenter et exploiter ces informations n'a jamais été aussi crucial. Le clustering, ou regroupement, est l'une des méthodes les plus puissantes pour extraire des structures cachées au sein de grandes quantités de données de façon automatisée, qu'il s'agisse de données biologiques, sociales, économiques ou encore astronomiques. Cependant, la gestion de l'incertitude des données et des modèles constitue un défi majeur pour obtenir des résultats à la fois précis et pertinents.

L'approche bayésienne, en exploitant les informations a priori et en modélisant les incertitudes, se distingue des méthodes classiques par sa capacité à offrir une vision nuancée des résultats. Plutôt que de se limiter à une partition brute des données, elle permet de considérer chaque regroupement comme une probabilité, ce qui permet de mieux prendre en compte les variations et les ambiguïtés inhérentes aux ensembles complexes. Cette approche est particulièrement pertinente pour l'analyse de données hétérogènes et incertaines, où chaque décision de regroupement comporte une marge d'erreur. L'objectif de notre travail est d'appliquer une méthodologie bayésienne afin d'améliorer le processus de clustering, en l'appliquant à un jeu de données relativement simple, ce qui permettra de mieux comprendre les étapes du processus et de démontrer l'efficacité de cette approche.

Dans ce projet, nous nous intéressons à l'application du clustering bayésien pour analyser son efficacité et ses avantages par rapport aux méthodes classiques. Nous commencerons par introduire les bases théoriques du clustering et de la méthode bayésienne, avant de détailler la mise en œuvre d'un modèle de mélange gaussien. Nous examinerons ensuite l'estimation des paramètres par échantillonnage de Gibbs et aborderons la problématique du label switching. Enfin, nous étudierons la détermination du nombre optimal de clusters et comparerons les résultats obtenus avec ceux des méthodes classiques afin d'évaluer la pertinence de l'approche bayésienne dans ce contexte.

[1] Source : *Informations publiées dans le monde sur le net* - Planetoscope

Chapitre 1

Définition des notions

Ce chapitre a pour objectif de définir les concepts fondamentaux sur lesquels repose notre projet de clustering bayésien du jeu de données des galaxies. Nous commencerons par poser le cadre du projet, puis nous aborderons les principales notions de la méthode bayésienne et du clustering, en lien avec le dataset des galaxies, sur lequel nous allons appliquer ces techniques. Nous expliquerons ainsi le principe de la méthode bayésienne et comment elle est utilisée pour estimer les paramètres du modèle dans un contexte de clustering. Enfin, nous présenterons le jeu de données utilisé dans le projet et son contexte scientifique.

1.1 Cadre du projet

Dans le cadre de notre projet de quatrième année en IMDS (Ingénierie Mathématique et Data Science) à Polytech Clermont pour l'année 2024/2025, nous avons choisi de travailler pendant 5 mois sur un sujet encadré par Monsieur Martin Metodiev : "*Approche bayésienne de clustering sur le jeu de données des galaxies*". Ce projet fait partie intégrante de notre formation et s'inscrit dans l'application de méthodes statistiques avancées à des problématiques concrètes. En effet, les enjeux liés au clustering font partie des algorithmes d'apprentissage non supervisés, largement utilisés dans de nombreux domaines aujourd'hui. Que ce soit pour analyser les relations entre individus ou pour traiter les flux massifs de données dans le cadre du "big data", ces techniques sont omniprésentes.

1.2 Explication du sujet

Décomposons le sujet en sous-parties et comprenons ensemble ce qui est attendu de nous dans ce projet. Nous verrons donc dans un premier temps le principe de la méthode bayésienne ensuite nous discuterons des objectifs du clustering et l'application au cadre bayésien. Enfin nous présenterons le jeu de données sur lequel nous avons travaillé.

1.2.1 Méthode bayésienne

La méthode bayésienne est fondée sur le théorème de Bayes, formulé par Thomas Bayes en 1763. Il stipule que, pour deux événements A et B, de probabilités respectives $P(A)$ et $P(B)$ et tel que $P(A) \neq 0$, on a la relation :

$$P(\mathrm{B} \mid \mathrm{A}) = \frac{P(\mathrm{A} \mid \mathrm{B}) \times P(\mathrm{B})}{P(\mathrm{A})}$$

où $A \mid B$ représente l'événement A sachant B, et $B \mid A$ représente l'événement B sachant A. Dans l'approche bayésienne, l'idée est que A dans la formule ci-dessus représente les paramètres (ou toutes nos lois), que l'on écrira θ, tandis que B que l'on appelera y représente nos observations donc des informations connues. Dans le contexte de ce sujet, y correspond au dataset des galaxies.

On intègre alors trois concepts :

1. La loi a priori : $p(\theta)$ c'est ce que rajoute la méthode bayésienne. Nous considérons nos paramètres comme des variables aléatoires. Ainsi $p(\theta)$ repésente la densité de la loi posée sur nos paramètres.

2. La loi a posteriori : $p(\theta \mid y_1, \ldots, y_N)$ avec $y = (y_1, \ldots, y_N)$. C'est la densité de nos paramètres sachant nos données.

3. La vraissemblance de nos données : $p(y|\theta)$ que l'on connait car on fixe préalablement une loi sur nos données.

La loi a priori $p(\theta)$ représente nos connaissances relatives à θ avant l'observation des données. La loi a posteriori $p(\theta \mid y)$, quant à elle, permet de mettre à jour nos connaissances sur θ une fois que les données Y ont été observées. La loi de Bayes se généralise avec les densités en :

$$p(\theta \mid y_1, \ldots, y_N) = \frac{p(y_1, \ldots, y_N \mid \theta) \times p(\theta)}{p(y_1, ..., y_N)}$$

Ainsi, à partir des observations $y = (y_1, \ldots, y_N)$, l'objectif de la méthode bayésienne est de calculer la loi a posteriori de θ, afin d'obtenir des informations sur ses paramètres. Cette loi est cruciale, car elle permet d'obtenir un estimateur de θ. En effet, une fois la loi a posteriori déterminée, on peut en extraire l'estimateur MAP (Maximum A Posteriori), qui représente la valeur de θ maximisant cette probabilité a posteriori. Il est définit comme suit :

$$\hat{\theta}_{MAP} = \underset{\theta \in \Theta}{\arg\max} \{p(\theta \mid y_1, \ldots, y_N)\} \tag{1.1}$$

Nous appliquons le logarithme car cela facilite souvent la recherche de $\hat{\theta}_{MAP}$.

1.2.2 Principe du clustering

Dans cette section nous souhaitons expliquer en quoi consiste le clustering pour pouvoir ensuite relier le clustering à l'approche bayésienne.

Soit $y = (y_1, \ldots, y_N)$ un ensemble de données à N observations. Le clustering désigne un ensemble d'algorithmes et de méthodes statistiques ou non, permettant de regrouper les données en $K \in \mathbb{N}^*$ clusters (ou groupes) selon leur ressemblance. L'objectif principal de cette approche est de découvrir des structures sous-jacentes dans les données, en identifiant des groupes d'observations qui partagent des caractéristiques similaires.

Cependant, la notion de ressemblance peut rapidement devenir abstraite d'un point de vue humain. Tandis que le regroupement de données dans un espace à deux dimensions peut être intuitif et relativement simple, cette tâche devient beaucoup plus complexe lorsqu'il s'agit d'observations à 7 dimensions, voire plus. Dans ces cas, il devient pratiquement impossible de visualiser ou de comprendre les relations entre les données sans recourir à des outils mathématiques

et algorithmiques adaptés. C'est pourquoi des techniques de clustering ont été développées, permettant de traiter efficacement ces situations.

Ces algorithmes sont essentiels dans de nombreux domaines, car ils automatisent un processus qui serait autrement laborieux et sujet à des biais humains. Par exemple, dans le traitement de grandes quantités de données comme celles générées par les utilisateurs sur les réseaux sociaux, le suivi des cookies sur internet ou encore l'analyse des comportements d'individus dans divers contextes, le clustering permet de dégager des tendances et des liens. Cela permet aux entreprises, chercheurs et analystes de mieux comprendre des ensembles de données complexes, d'en extraire de l'information utile, et de prendre des décisions éclairées.

Notre projet consiste donc à utiliser la méthode bayésienne décrite dans la section précédente pour faire du clustering. Ainsi, trouver des estimations de différents paramètres qui nous permettront de définir à quel groupe appartient chaque individu.

1.2.3 Présentation du jeu de données

Le jeu de données que nous utilisons pour ce projet est celui des galaxies, accessible via la bibliothèque MASS (Venables and Ripley (2002)) en R. Ce jeu de données contient les vitesses en kilomètres par seconde de 82 galaxies. Ces vitesses sont issues d'une étude de galaxies dans la région de Corona Borealis, qui fait partie d'un large ensemble de données issues de relevés astrophysiques. Ce jeu de données permet d'analyser les vitesses des galaxies afin de détecter des structures à grande échelle dans l'univers. Ce jeu de données a beaucoup été utilisé dans la littérature scientifique pour discuter notamment de clustering. Nous pouvons citer par exemple Grün et al. (2022) ou encore Aitkin (2001) mais il en existe bien d'autre. Lors de l'utilisation de ce jeu de données, nous avons divisé la vitesse par 1000 afin que les données soient moins dispersées.

Chapitre 2

Problème avec un nombre de clusters fixé

Dans ce chapitre, nous explorons l'utilisation des mixtures gaussiennes pour effectuer du clustering. L'objectif est d'estimer les paramètres du modèle, à savoir les poids des clusters π_k, leur moyennes μ_k et leur variances σ_k^2, en adoptant une approche bayésienne avec des a priori conjugués, à partir des observations tirées du dataset des galaxies, et pour un nombre fixe de clusters K. Cette approche repose sur le calcul du Maximum A Posteriori (MAP), ce qui permet d'intégrer des connaissances préalables tout en maximisant la vraisemblance des données. Le chapitre aborde également les défis associés au choix des a priori et à l'échantillonnage des paramètres à l'aide de l'algorithme de Gibbs.

2.1 Formalisation du problème

Pour ce projet, nous allons utiliser les mixtures gaussiennes pour effectuer du clustering. Le principe de cette approche est le suivant : soit Y la loi de nos données, et supposons que celles-ci suivent une mixture gaussienne. La densité de cette loi est donnée par :

$$f_Y(y) = \sum_{k=1}^{K} \pi_k \times \mathcal{N}(y|\mu_k, \sigma_k^2)$$

avec K le nombre de groupes et $\mathcal{N}(y|\mu_k, \sigma_k^2)$ la densité d'une loi gaussienne en y, de moyenne μ_k et de variance σ_k^2 et telle que $\sum_{k=1}^{K} \pi_k = 1$.

L'objectif ici est de déterminer les paramètres $(\pi_1, ..., \pi_K, \mu_1, ..., \mu_K, \sigma_1^2, ..., \sigma_K^2) = \theta$ du modèle. Dans notre cas nous utilisons l'approche bayésienne, nous allons fixer des a priori sur nos paramètres et pour trouver une estimation de ceux-ci, nous utiliserons le MAP (Maximum A Posteriori). On veut maximiser :

$$\begin{aligned} \hat{\theta}_{MAP} &= \arg\max_{\theta \in \Theta} \{p(\theta \mid y_1, \ldots, y_N)\} \\ \text{Or, } p(y_1, \ldots, y_N) &\text{ est indépendant de } \theta \text{ donc,} \\ \hat{\theta}_{MAP} &= \arg\max_{\theta \in \Theta} \{p(y_1, \ldots, y_N \mid \theta) \times p(\theta)\} \\ &= \arg\max_{\theta \in \Theta} \{\log(p(y_1, \ldots, y_N \mid \theta)) + \log(p(\theta))\} \end{aligned} \tag{2.1}$$

avec $\log(p(y_1, \ldots, y_N \mid \theta))$ la log-vraissemblance de nos données et $p(\theta)$ la densité sur nos a priori.

De plus,

$$\log(p(y_1, \ldots, y_N \mid \theta)) = \sum_{i=1}^{N} \log(\sum_{k=1}^{K} \pi_k \times \mathcal{N}(y_i|\mu_k, \sigma_k^2))$$

Enfin, si nous connaissons à l'avance le groupe dans lequel appartient notre individu, disons k, on peut écrire que celui-ci suit une loi gausienne de paramètre (μ_k, σ_k^2). Nous introduisons donc aussi les variables latentes Z_i qui correspondent à un vecteur dans lequel toutes les composantes sont nulles sauf une qui correspond au groupe auquel appartient y_i. Ainsi,

$$\forall\, i \in [\![1; N]\!]\ Z_i|\theta\ \sim \mathcal{M}(1, (\pi_1, ..., \pi_K)) \tag{2.2}$$

Et nous construisons donc la matrice $Z \in \mathcal{M}_{N\times K}(\mathbb{R})$ et notons **z** la matrice des observations des Z_i. Dans ce cas,

$$\forall\, i \in [\![1; N]\!],\ y_i|Z_{ik} = 1\ \sim \mathcal{N}(\mu_k, \sigma_k^2) \tag{2.3}$$

Pour la suite de ce chapitre, nous considèrons $\mu = (\mu_1, \ldots, \mu_K),\ \sigma^2 = (\sigma_1^2, \ldots, \sigma_K^2)$ et $\pi = (\pi_1, \ldots, \pi_K)$.

2.2 Les a priori

Dans cette section, nous expliquerons et définirons les a priori pour notre problème. Le choix des a priori est un élément clé, car il influence l'estimation des paramètres finaux. Pour choisir ces lois a priori, nous avons deux options parmi les plus fréquentes : nous pouvons sélectionner des lois qui reflètent nos connaissances préalables sur le problème, ou bien opter pour des a priori conjugués. En effet, un expert dans un domaine spécifique pourra fournir des informations pertinentes, par exemple en précisant que certaines valeurs ne peuvent excéder un certain seuil, ou encore que les valeurs sont discrètes et se limitent à des intervalles précis. Cependant, dans notre cas, nous ne disposons d'aucune connaissance préalable spécifique sur le sujet, ce qui rend l'option des a priori informatifs peu viable. Dès lors, nous nous orientons vers des a priori conjugués. Nous examinerons tout d'abord la théorie sous-jacente aux a priori, puis leur application ainsi que nos résultats.

2.2.1 Théorie

Un a priori est dit conjugué si les lois a priori et a posteriori appartiennent à la même famille. L'avantage principal d'un tel choix réside dans la simplification des calculs : en effet, en optant pour ces a priori, nous connaissons déjà la forme de la distribution a posteriori, ce qui facilite grandement les étapes de calcul. Les a priori conjugués que nous utilisons ont notamment été présentés dans ces deux articles : Aitkin (2001) et Grün et al. (2022). Ces choix nous permettent d'obtenir les formes suivantes pour nos a priori :

$$\pi = (\pi_1, ..., \pi_K) \sim Dir(\alpha_1, \ldots, \alpha_K), \quad \alpha_1, \ldots, \alpha_K > 0 \tag{2.4}$$

$$\sigma_k^2 \sim \mathcal{IG}(\nu_0\ , \Lambda_0), \quad \nu_0, \Lambda_0 > 0 \tag{2.5}$$

$$\mu_k|\sigma_k^2 \sim \mathcal{N}(\mu_0\ , \sigma_k^2/\kappa_0), \quad \kappa_0, \mu_0 > 0 \tag{2.6}$$

$\mathcal{IG}$ représentant la loi inverse-gamma.

Nous devons maintenant déterminer les valeurs des paramètres de nos lois a priori. Pour cela, nous avons utilisé la méthode du *prior predictive check* décrite par Conn et al. (2018). Cette méthode, bien qu'elle soit subjective, repose sur l'idée suivante : générer des données à partir de nos a priori et comparer les distributions obtenues. Si celles-ci sont cohérentes avec les données réelles, cela indique que nos choix d'a priori sont pertinents. Sinon, il sera nécessaire de les ajuster.

Voici un court algorithme qui explique comment générer ces données.

Algorithm 1 Génération de données pour les prior predictive chekcs

Entrées : $(\alpha_1, \ldots, \alpha_K), \nu_0, \Lambda_0, \mu_0, \kappa_0$.
Tirer π, σ^2, μ **:** Le tirage est réalisé grâce aux lois définies précédemment (2.4), (2.5), (2.6)
Tirer Z puis un élément y_i **:**

Pour i **allant de 1 à T :** (T le nombre de tirages)
Tirer Z grâce à $Z_i|\theta \sim \mathcal{M}(1, (\pi_1, ..., \pi_K))$ (2.2)
Tirer les éléments avec $Y_i|Z_{ik} = 1 \sim \mathcal{N}(\mu_k, \sigma_k^2)$ (2.3)
Fin Pour

Sorties : $y = (y_1, \ldots, y_T)$.

2.2.2 Pratique

Dorénavant nous savons comment procéder pour choisir nos a priori. Nous avons essayé différentes valeurs pour les paramètres de nos a priori en se basant sur la moyenne de nos données ou sa variance par exemple. Nous avons aussi fait en sorte que nos a priori ne soient pas trop informatifs. Nous implémentons l'algorithme sur R, nous choisissons de faire des simulations avec 3 clusters. Nous avons commencé par choisir la valeur de $\mu_0 = 20.8$ qui est la moyenne des données. Utiliser les valeurs données directement pour trouver ceux des a priori n'est pas recomandé car justement on cherche cela. Cependant pour la moyenne cela n'est pas gênant puisque celle-ci représente les données plus généralement et non de manière très précise. Parmi les essais que nous avons pu faire la plupart n'étaient pas réussi. Par exemple pour la figure 2.1 nous remarquons que les 3 clusters sont complétement mélangés, ce qui ne correspond pas du tout à ce que nous observons sur la répartition des galaxies 2.3. Concernant la figure 2.2, celle-ci possède bien 3 clusters distincts mais qui sont très éloignés les uns des autres et, un des clusters se trouve dans le négatif alors que les données représentent des vitesses.

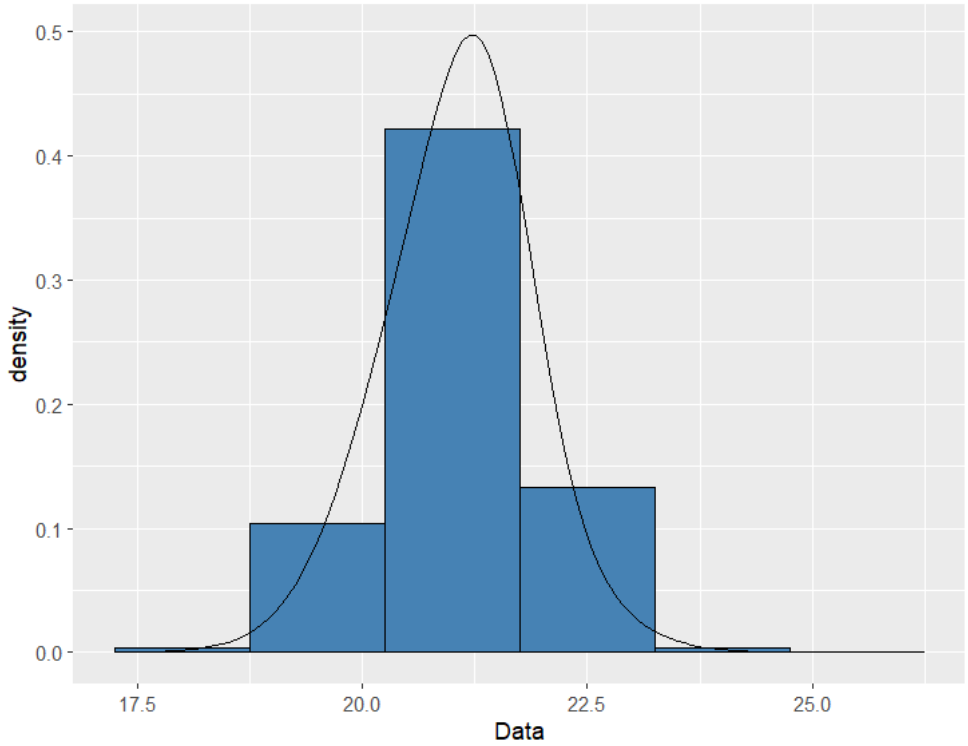

FIGURE 2.1 : Répartition des galaxies

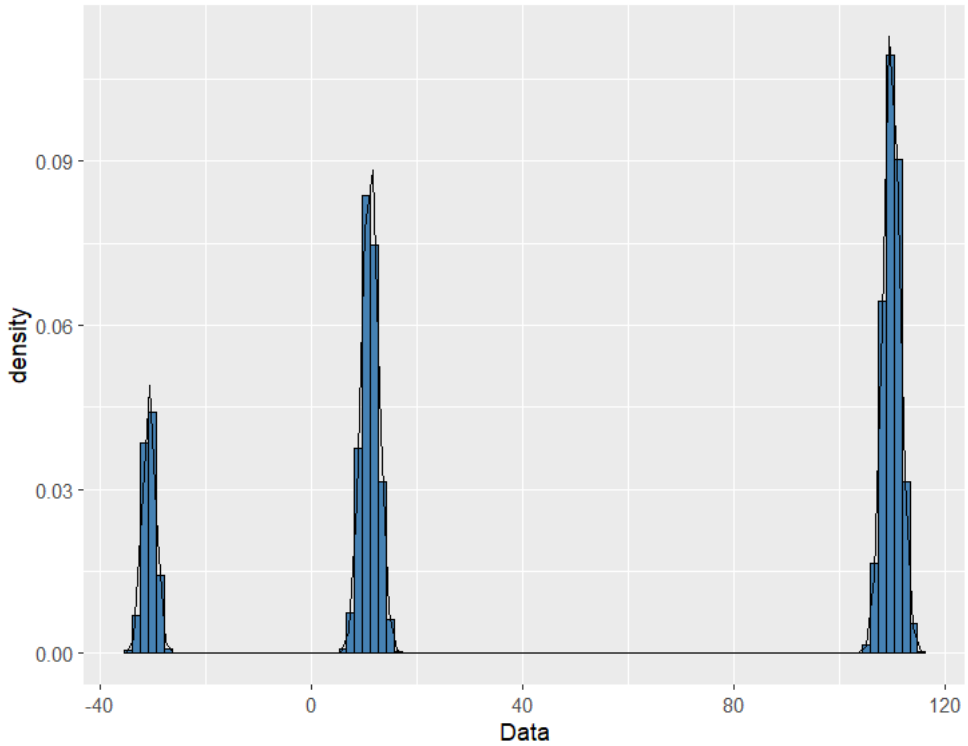

FIGURE 2.2 : Répartition des galaxies

C'est donc avec des raisonnements similaires que sommes arrivés par choisir ces a priori :

$$\pi = (\pi_1, ..., \pi_K) \sim Dir(3, \ldots, 3)$$
$$\sigma_k^2 \sim \mathcal{IG}(100, 100)$$
$$\mu_k | \sigma_k^2 \sim \mathcal{N}(20.8\ , 10 \times \sigma_k^2)$$

Voici donc, la distribution des galaxies représentée sur un histogramme 2.3 puis, la répartition de 10000 données simulées suivants nos a priori 2.4. Nous pouvons donc remarquer en comparant les répartitions que nos choix d'a priori ont bien un sens. Effectivement sur les 3 tirages montrés nous observons bien 3 clusters situés sur une plage cohérente avec les données et, avec des tailles non aberrantes.

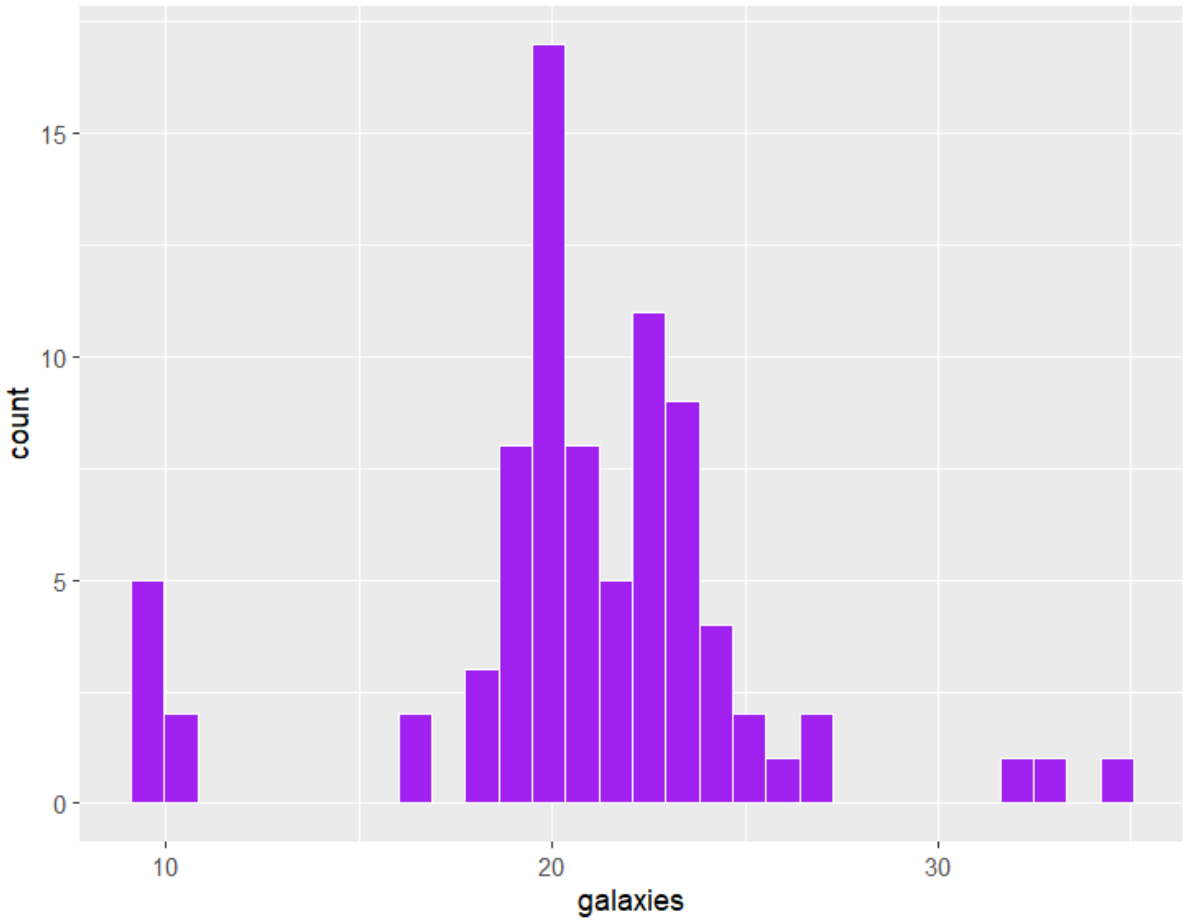

FIGURE 2.3 : Répartition des galaxies

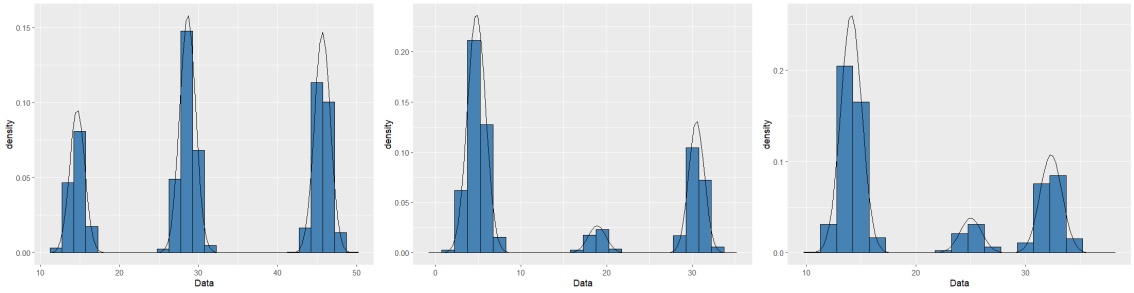

FIGURE 2.4 : Répartition des données générées suivant nos a priori

2.3 Gibbs sampling

Comme les distributions des θ sont complexes, il est difficile de déterminer explicitement la distribution a posteriori. Il est donc nécessaire de recourir à des algorithmes plus avancés pour échantillonner dans des distributions complexes ou inconnues. Ceux que nous avons étudiés pour ce problème proviennent de la méthode Monte Carlo Markov Chain (MCMC). Ces algorithmes sont itératifs et sans mémoire, ce qui signifie que seule la valeur précédente est nécessaire pour déterminer une nouvelle valeur. Ils permettent de générer $N \in \mathbb{N}$ vecteurs $y = (y_1, \ldots, y_n)$ (avec $n \in \mathbb{N}$) suivant approximativement une distribution de probabilité donnée π. Le Gibbs sampling (ou Échantillonneur de Gibbs en français) fait partie de ces algorithmes issus de la méthode MCMC. Il est particulièrement utile pour déterminer les paramètres d'une distribution multivariée. Examinons d'abord la théorie sous-jacente, puis les résultats une fois l'algorithme appliqué à notre problème.

2.3.1 Théorie

L'approche de l'échantillonneur de Gibbs consiste à découper le problème de manière à calculer $\theta^{(t+1)}$ à partir de $\theta^{(t)}$ en utilisant les probabilités conditionnelles. Selon la définition donnée par Robert et al. (2007), pour une distribution conjointe $p(\theta_1, \ldots, \theta_n)$ avec leurs lois conditionnelles respectives $p_1, \ldots, p_n$, l'algorithme général du Gibbs Sampling s'écrit, pour $(\theta_1^{(t)}, \ldots, \theta_n^{(t)})$ fixé à l'itération t, comme suit :

- $\theta_1^{(t+1)} \sim p_1(\theta_1 \mid \theta_2^{(t)}, \ldots, \theta_n^{(t)})$;
- $\theta_2^{(t+1)} \sim p_2(\theta_2 \mid \theta_1^{(t+1)}, \theta_3^{(t)}, \ldots, \theta_n^{(t)})$;

  $\ldots$
- $\theta_n^{(t+1)} \sim p_n(\theta_n \mid \theta_1^{(t+1)}, \ldots, \theta_{n-1}^{(t+1)})$

On constate alors que chaque θ_i est mis à jour conditionnellement aux autres paramètres, qui le sont à leur tour ensuite.

Dans notre cas, celui de la mixture gaussienne, nous avons pour K le nombre de groupes $\theta = (\pi, \mu, \sigma^2)$ avec $\pi = (\pi_1, \ldots, \pi_K)$, $\mu = (\mu_1, \ldots, \mu_K)$, $\sigma^2 = (\sigma_1^2, \ldots, \sigma_K^2)$ et enfin notre Z. On rappelle que $y = (y_1, \ldots, y_n)$ est le vecteur contenant nos observations.

Nous commençons par tirer $Z \mid \theta$: (On note $\theta_k = (\pi_k, \mu_k, \sigma_k^2)$)

$$1 \leq i \leq n,\ Z_i \mid y_i, \theta \sim \mathcal{M}_K(1;\ p_1(y_1,\ \theta_1),\ \ldots\ p_k(y_K,\ \theta_K))$$

Avec pour $1 \leq k \leq K$:

$$z_{ik} = \frac{\pi_k \times \mathcal{N}(y_i;\ \mu_k,\ \sigma_k^2)}{\sum_{p=1}^{K}(\pi_p \times \mathcal{N}(y_i;\ \mu_p,\ \sigma_p^2))}$$

Démonstration de l'équation :

$$\begin{aligned}
Z \mid \theta, y \sim p(Z \mid \theta, y) &= \frac{p(Z, \theta, y)}{\sum_Z p(\theta, y \mid Z) \times p(Z)} \\
&\propto p(\theta) \times p(Z \mid \theta) \times p(y \mid Z, \theta) \\
&= p(\theta) \times p(Z \mid \pi) \times p(y \mid Z, \mu, \sigma^2) \\
&\propto p(Z \mid \pi) \times p(y \mid Z, \mu, \sigma^2) \\
&\propto \prod_{i=1}^{N} \prod_{k=1}^{K} \left(\pi_k \times \mathcal{N}(y_i; \mu_k, \sigma_k^2)\right)^{Z_{ik}} \\
&\propto \prod_{i=1}^{N} p(Z_i \mid \theta, y)
\end{aligned}$$

$$\text{Donc, } \exists C_i \in \mathbb{R}_+^* \text{une constante, } p(Z_i \mid \theta, y) = \prod_{k=1}^{K} \left(\frac{\pi_k \times \mathcal{N}(y_i; \mu_k, \sigma_k^2)}{C_i}\right)^{Z_{ik}}$$

$$\text{Alors, } \sum_{k=1}^{K} P(Z_{ik} = 1 \mid \theta, y) = \sum_{k=1}^{K} \left(\frac{\pi_k \times \mathcal{N}(y_i; \mu_k, \sigma_k^2)}{C_i}\right) = 1$$

$$\text{Ainsi, } C_i = \sum_{k=1}^{K} (\pi_k \times \mathcal{N}(y_i; \mu_k,\ \sigma_k^2))$$

$$\text{Donc, } p(Z_i \mid \theta, y) = \prod_{k=1}^{K} \left(\frac{\pi_k \times \mathcal{N}(y_i; \mu_k, \sigma_k^2)}{\sum_{p=1}^{K}(\pi_p \times \mathcal{N}(y_i; \mu_p,\ \sigma_p^2))}\right)^{Z_{ik}}$$

$$\text{Ainsi, } z_{ik} = \frac{\pi_k \times \mathcal{N}(y_i;\ \mu_k,\ \sigma_k^2)}{\sum_{p=1}^{K}(\pi_p \times \mathcal{N}(y_i;\ \mu_p,\ \sigma_p^2))} \quad \square$$

Ensuite nous tirons successivement :

$$\pi \mid y, \mathbf{z} \sim Dir\left(\alpha_1 + m_1(\mathbf{z}),\ \ldots,\ \alpha_K + m_K(\mathbf{z})\right) \tag{2.7}$$

$$\sigma_k^2 \mid y, \mathbf{z} \sim \mathcal{IG}\left(\frac{\nu_0 + m_k(\mathbf{z})}{2},\ \frac{1}{2}\left[\Lambda_0 + \hat{s}_k^2(y, \mathbf{z}) + \frac{\kappa_0\ m_k(\mathbf{z})}{\kappa_0 + m_k(\mathbf{z})}\right]\right) \tag{2.8}$$

$$\mu_k \mid y, \mathbf{z}, \sigma_k^2 \sim \mathcal{N}\left(\xi_k(y, \mathbf{z}),\ \frac{\sigma_k^2}{\kappa_0 + m_k(\mathbf{z})}\right) \tag{2.9}$$

Avec $\alpha = (\alpha_1,\ \ldots,\ \alpha_k), \nu_0, \kappa_0, \Lambda_0$ et μ_0 les a priori vu en 2.2, et les fonctions, en s'inspirant de Robert et al. (2007) et de Gelman et al. (1995), comme suivent :

- $m_k(\mathbf{z})$: le nombre d'observations dans le cluster k :

$$m_k(\mathbf{z}) = \sum_{i=1}^{n} z_{ik}$$

- $\bar{y}_k(\mathbf{z})$: la valeur moyenne des observations du cluster k ;

$$\bar{y}_k(\mathbf{z}) = \frac{1}{m_k(\mathbf{z})} \sum_{i=1}^{n} z_{ik}\ y_i$$

- $\hat{s}_k^2(y, \mathbf{z})$: l'estimation de la variance dans le cluster k ;

$$\hat{s}_k^2(y, \mathbf{z}) = \sum_{i=1}^{n} z_{ik}\ (y_i - \bar{y}_k(\mathbf{z}))^2$$

- $\xi_k(X, \mathbf{z})$;

$$\xi_k(y, \mathbf{z}) = \frac{\kappa_0\ \mu_0 + m_k(\mathbf{z})\ \bar{y}_k(\mathbf{z})}{\kappa_0 + m_k(\mathbf{z})}$$

Voilà la majorité de la théorie requise pour pouvoir entâmer la programmation et l'application de notre premier Gibbs sampler sur le jeu de donnée des galaxies.

2.3.2 Application

Notre premier objectif dans ce projet était de réaliser l'entiéreté des algorithmes à la main afin de comprendre au maximum la théorie soujascente, nous avons donc essayé de programmer le Gibbs sampler par nous même en R. Nous avons créé toutes les fonctions données ci-dessus avec quelques modifications pratiques.

Effectivement dans la formule ci-dessus on voit que $m_k(\mathbf{z})$ peut valoir 0 si le cluster est vide, ce qui arrive parfois et d'autant plus avec un nombre de clusters élevé. Or cela pose un problème dans le calcul de $\bar{y}_k(\mathbf{z})$ qui fait une division par 0 dans le cas où le cluster k est vide, alors l'approche que nous avons choisi pour ce problème est de renvoyer $\bar{y}_k(\mathbf{z}) = 0$ si $m_k(\mathbf{z}) = 0$. De même dans le tirage des μ, avec la formule en (2.9), on retrouve un problème similaire pour $\frac{\sigma_k^2}{\kappa_0 + m_k(\mathbf{z})}$. En effet dans de rare cas, pour un Λ_0 très petit, un σ_k^2 aussi très petit et $m_k(\mathbf{z}) = 0$, la valeur dépasse $1e307$ qui est à peu près la limite mathématique de R. Ainsi la valeur était considérée comme un infini et faisait planter notre programme, pour résoudre ce problème, nous demandons si la valeur calculée est infinie et si c'est le cas, on la remplace par $1e300$ par sécurité, la valeur reste très haute mais elle ne pose plus de problème à l'exécution.

Nous avons parlé dans la partie précédente du fait qu'il faille une initialisation pour notre échantillonneur de Gibbs, ce dernier peut soit être $\mathbf{z}$, soit des valeurs π_0, μ_k, et σ_0^2. Dans notre cas, nous avons choisi de commencer par initialiser $\mathbf{z}$, pour ce faire nous suivons l'algorithme suivant qui correspond à l'utilisation d'une CAH (classification ascendante hiérarchique) :

Algorithm 2 Initalisation de $\mathbf{z}^{(0)}$

Entrées : y, K, N
Calculer d la distance euclidienne entre les galaxies (y)
Calculer $clust$ le résultat du clustering hiérarchique sur d selon la méthode de Ward
Calculer $group$ l'assignation de chaque observation à un des K clusters, à partir du découpage de $clust$
Initialiser $\mathbf{z}^{(0)} \in \mathcal{M}_{N\times K}(\mathbb{R})$ avec des 0
Pour i allant de 1 à N :
    Mettre $\mathbf{z}_{ik}^{(0)}$ à 1 pour $k = group[i]$ (la valeur de $group$ pour l'observation i)
Fin Pour
Sorties : $\mathbf{z}^{(0)} \in \mathcal{M}_{N\times K}(\mathbb{R})$.

Le Gibbs sampling est d'autant plus efficace lorsqu'on a déjà une idée de la structure des données, d'où l'importance des a priori. Ainsi, nous avons estimé que réaliser un clustering ascendant hiérarchique pour initialiser $\mathbf{z}^{(0)}$ offrait un bon point de départ, cette première partition étant généralement assez fidèle à la structure réelle des données et au résultat final.

Le Gibbs sampling repose sur un principe de convergence : on s'attend à ce qu'avec un nombre suffisant d'itérations, le modèle converge vers la distribution des données observée. Cependant, étant donné que les valeurs d'initialisation sont choisies manuellement en fonction de notre compréhension des données, elles introduisent un biais. Par conséquent, les premières itérations de l'échantillonneur sont biaisées et ne doivent pas être conservées. Ce phénomène est appelé *burn-in*, et dans notre cas, nous supprimons les 3000 premières itérations.

Le Gibbs sampling que nous avons mis en place suit l'algorithme suivant :

Algorithm 3 Gibbs sampling

Entrées : $X, \mathbf{z}^{(0)}, K, N, NbIterations, ValeurBurn$
Initialisation $M_{\sigma^2} \in \mathcal{M}_{NbIterations\times K}(\mathbb{R})$ la matrice contenant les σ^2 calculés
Initialisation $M_{\mu} \in \mathcal{M}_{NbIterations\times K}(\mathbb{R})$ la matrice contenant les μ calculés
Initialisation $M_{\pi} \in \mathcal{M}_{NbIterations\times K}(\mathbb{R})$ la matrice contenant les π calculés
Initialisation $M_{\mathbf{z}} \in Array_{N\times K\times NbIterations}(\mathbb{R})$ la matrice contenant tous les $\mathbf{z}_j$ avec $M_{\mathbf{z}}[,,0] = \mathbf{z}^{(0)}$

Pour t allant de 1 à T **:** (T le nombre d'itérations)
    Tirer $(\sigma^2)^{(t)}$ selon (2.8) avec $\mathbf{z}^{(t-1)} = M_{\mathbf{z}}[,,t-1]$ et le placer dans M_{σ^2} dans la rangée t
    Tirer $\mu^{(t)}$ selon (2.9) avec $\mathbf{z}^{(t-1)}, (\sigma^2)^{(t)}$ et le placer dans M_{μ} dans la rangée t
    Tirer $\pi^{(t)}$ selon (2.7) avec $\mathbf{z}^{(t-1)}$ et le placer dans M_{π} dans la rangée t
    Tirer $\mathbf{z}^{(t)}$ et le placer dans $M_{\mathbf{z}}$ à l'indice $[,,t]$
Fin Pour
Sorties : $M_{\sigma^2}, M_{\mu}, M_{\pi}, M_{\mathbf{z}}$ dont on ne garde que les $NbIterations - ValeurBurn$ dernières valeurs.

Ainsi en plottant les résultats avec la librairie R GGplot2 (Wickham (2016)), nous trouvons les résultats suivants pour un K fixé à 3 sur la figure 2.5.

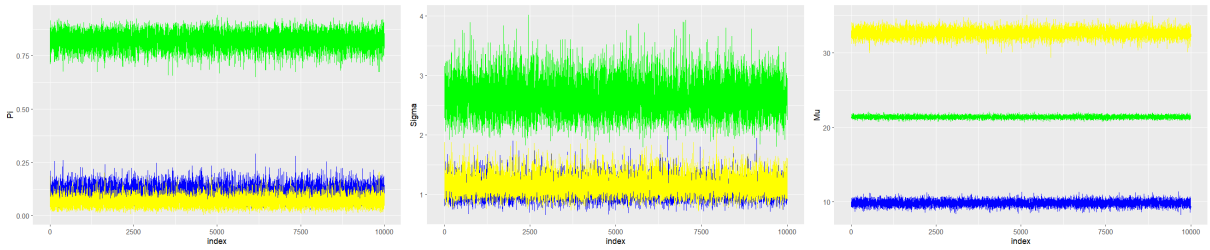

FIGURE 2.5 : Graphe des valeurs des paramètres pour K=3 (Dans l'ordre π, σ^2, μ)

2.4 Label Switching

Il arrive que, lorsque nous examinons les résultats obtenus à la sortie de l'échantillonneur Gibbs, nous obtenions des résultats tels que ceux présentés dans la figure 2.6. Ces résultats sont problématiques. Ce phénomène est connu sous le nom de label switching. Dans cette section, nous expliquerons la cause de ce problème et une méthodes pour y remédier.

FIGURE 2.6 : Résultat pour μ à la sortie de l'échantillonneur de Gibbs avec 5 groupes

2.4.1 Théorie

Le phénomène de label switching provient du fait que la fonction de densité des mélanges gaussiens est invariante par permutation des indices des composantes. En d'autres termes, la permutation des paramètres associés à ces composantes ne modifie pas la fonction de vraisemblance. Par exemple, on a :

$$p(y \mid \mu_1, \mu_2, \mu_3, \sigma_1^2, \sigma_2^2, \sigma_3^2, \pi_1, \pi_2, \pi_3) = p(y \mid \mu_2, \mu_1, \mu_3, \sigma_2^2, \sigma_1^2, \sigma_3^2, \pi_2, \pi_1, \pi_3)$$

Ainsi, la loi a posteriori est également invariante sous les permutations des indices des composantes. Cela pose un problème pour l'échantillonneur de Gibbs, qui, en échantillonnant la loi a posteriori, peut rencontrer différentes permutations des labels des composantes sans que cela n'affecte la vraisemblance des résultats. Par conséquent, l'échantillonneur peut assigner des étiquettes différentes aux mêmes composantes au cours des itérations, ce qui peut compliquer l'interprétation des résultats.

Ce problème a donné lieu à diverses solutions, et demeure un sujet de discussion actif. Dans le cadre de notre projet, nous n'avons pas cherché à analyser en détails le fonctionnement de ces algorithmes ni à déterminer lequel était le plus performant. Nous avons simplement choisi d'appliquer la méthode proposée par Stephens dans Stephens (2000). Cette approche repose sur la minimisation d'une fonction spécifique

2.4.2 Pratique

Pour implémenter la méthode de Stephens sur R, nous avons utilisé le package *label.switching* (Papastamoulis, 2016). L'utilisation de ce package nécessite la construction d'un tableau de dimensions $Nbrep \times N \times K$, où $Nbrep$ correspond au nombre d'itérations de l'échantillonneur de Gibbs. Le tableau H est défini comme suit :

$$H_{t,ik} = \frac{\pi_k \times \mathcal{N}(y_i; \mu_k^{(t)}, \sigma_k^{2^{(t)}})}{\sum_{l=1}^{K} \pi_l \times \mathcal{N}(y_i; \mu_l^{(t)}, \sigma_l^{2^{(t)}})}$$

La fonction retourne alors une liste des permutations nécessaires ainsi que les clusters associés à chaque individu. Nous avons utilisé cette liste de permutations avec une autre fonction du package (*permute.mcmc*) pour obtenir de nouvelles matrices représentant les valeurs de la distribution a posteriori de nos paramètres. Le résultat obtenu est illustré en 2.7. Nous avons constaté qu'il n'y avait alors plus de problème de label switching.

Nous avons effectué le label switching pour des fins de visualisation. Il est cependant important de souligner que, dans notre projet, le résultat final n'est pas affecté par ces permutations de groupes. En effet, le calcul du MAP et, plus généralement, de la vraisemblance sont indépendants de l'ordre des groupes.

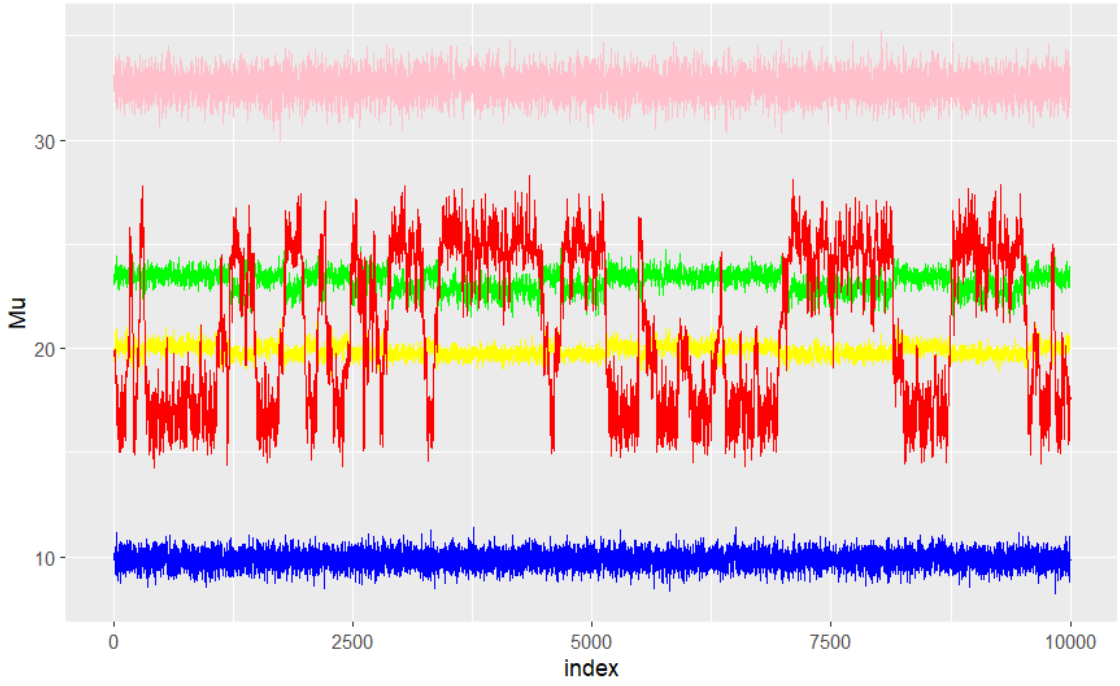

FIGURE 2.7 : Résultat pour μ à la sortie du label switching avec 5 groupes

2.5 Calcul du MAP et résultats finaux

Les parties précédentes nous ont permis d'extraire des valeurs de nos paramètres suivant la loi a posteriori de notre modèle. Il nous reste désormais à choisir les valeurs qui maximisent cette loi a posteriori. Pour ce faire, nous allons revenir sur la notion de MAP (Maximum A Posteriori), que nous réexpliquerons, puis calculerons afin de présenter nos résultats.

2.5.1 MAP

Ainsi pour calculer les valeurs de la distribution a posteriori nous faisons :

$$\begin{aligned} p(\theta|y) &\propto p(y|\theta) \times p(\theta) \\ &\propto p(y|\theta) \times p(\pi|\sigma^2, \mu) \times p(\sigma^2, \mu) \\ &\propto p(y|\theta) \times p(\pi) \times p(\mu|\sigma^2) \times p(\sigma^2) \end{aligned}$$

En passant au logarithme, on obtient,

$$\hat{\theta}_{MAP} = \arg\max_{\theta \in \Theta} \left\{ \log(p(y|\theta)) + \log(p(\pi)) + \log(p(\mu|\sigma^2)) + \log(p(\sigma^2)) \right\}$$

Nous calculons alors cette valeur a posteriori puis, nous gardons les paramètres et le z associé. Le passage au logarithme permet aussi d'éviter des problèmes numériques.

Pour obtenir à quel groupe appartient chaque individu, on récupère le $\hat{z}_{MAP}$ associé au $\hat{\theta}_{MAP}$, puis pour chaque ligne de z (donc pour chaque individu), le groupe dans lequel on met l'individu est l'argmax de z_i.

$$\forall\, i \in [\![1, N]\!], \quad \hat{G}_{iMAP} = \underset{k\in[\![1;K]\!]}{\arg\max}\, \{\hat{z}_{iMAP}\}$$

Ainsi, $\hat{G}_{MAP}$ est un vecteur de dimension N où chaque élément i représente le groupe d'appartenance de y_i.

2.5.2 Résultats

Lors de nos différents essais nous avons travaillé avec $K = 3$. Ce choix vient du fait que visuellement, les galaxies semblent facilement se diviser en 3 groupes. Il est donc plus aisé de vérifier si le résultat a bien un sens ou non. Nous obtenons alors en sortie ces valeurs (2.10), (2.11) et, (2.12) :

$$\hat{\pi} = (0.10432278, 0.82793671, 0.06774051) \tag{2.10}$$
$$\hat{\mu} = (9.979501, 21.455857, 33.048448) \tag{2.11}$$
$$\hat{\sigma}^2 = (1.041580, 1.964031, 1.098063) \tag{2.12}$$

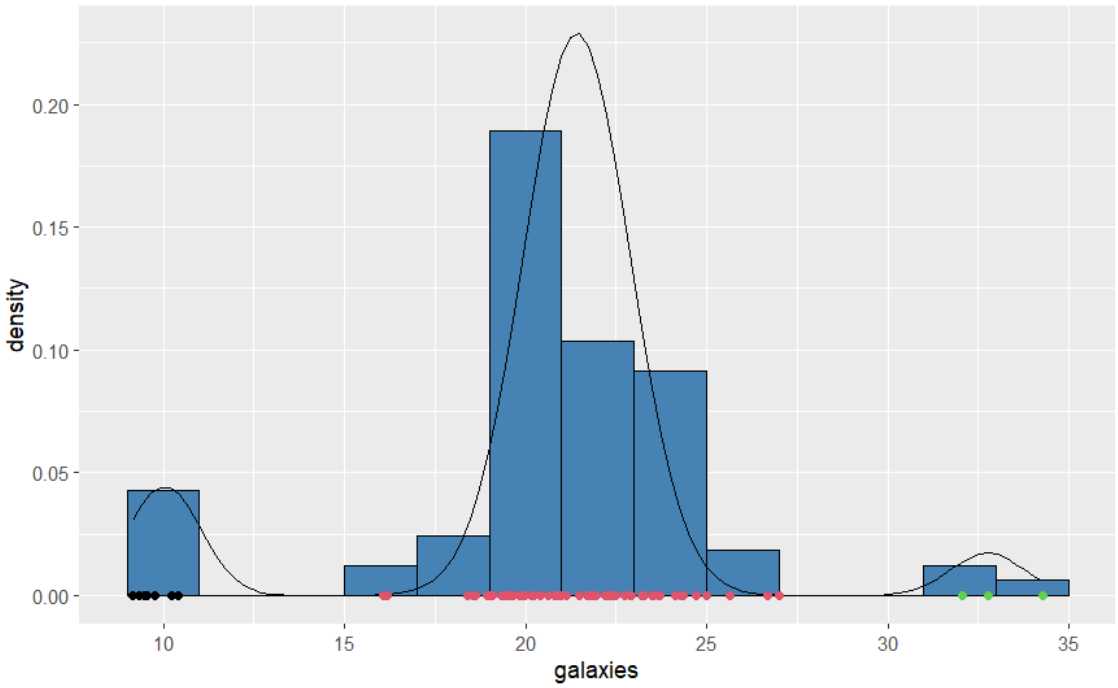

FIGURE 2.8 : Visualisation du clustering : répartition des galaxies, densités et groupes d'appartenance

La figure 2.5 montre clairement qu'il n'y a plus de phénomène de label switching sur nos paramètres. Concentrons-nous maintenant sur le clustering obtenu en sortie. La figure 2.8 présente la répartition des galaxies, représentée par des points sur l'axe des abscisses. Ces points sont colorés selon les groupes auxquels chaque galaxie appartient. On peut observer que le clustering obtenu, ainsi que la densité de la mixture gaussienne associée, sont cohérents avec ce que l'on pourrait attendre.

A noter aussi qu'il existe un package R qui permet de faire du Gibbs sampling pour les mixtures gaussiennes, c'est *bayesmix* (Gruen (2023)). Nous avons donc aussi utilisé ce package, puis appliqué la partie, label switc hing et MAP à la sortie. Nous obtenions alors des résultats assez différents, et qui ne nous satisfaisaient pas (voir (2.13), (2.14), (2.15) et 2.9). Ces résultats ne sont pas convenables car la moyenne du 3ème cluster est négative avec une variance très grande. Cependant l'assignation des groupes reste semblable à celle que l'on a trouvé à la main. Ces différences s'expliquent par le fait que nous avons laissé les fonctions du package décider des paramètres des a priori à notre place.

$$\hat{\pi} = (0.1179984, 0.7658334, 0.1161683) \tag{2.13}$$
$$\hat{\mu} = (9.914966, 21.401461, -1.901212) \tag{2.14}$$
$$\hat{\sigma^2} = (0.2540806, 5.490097, 1.303867e + 04) \tag{2.15}$$

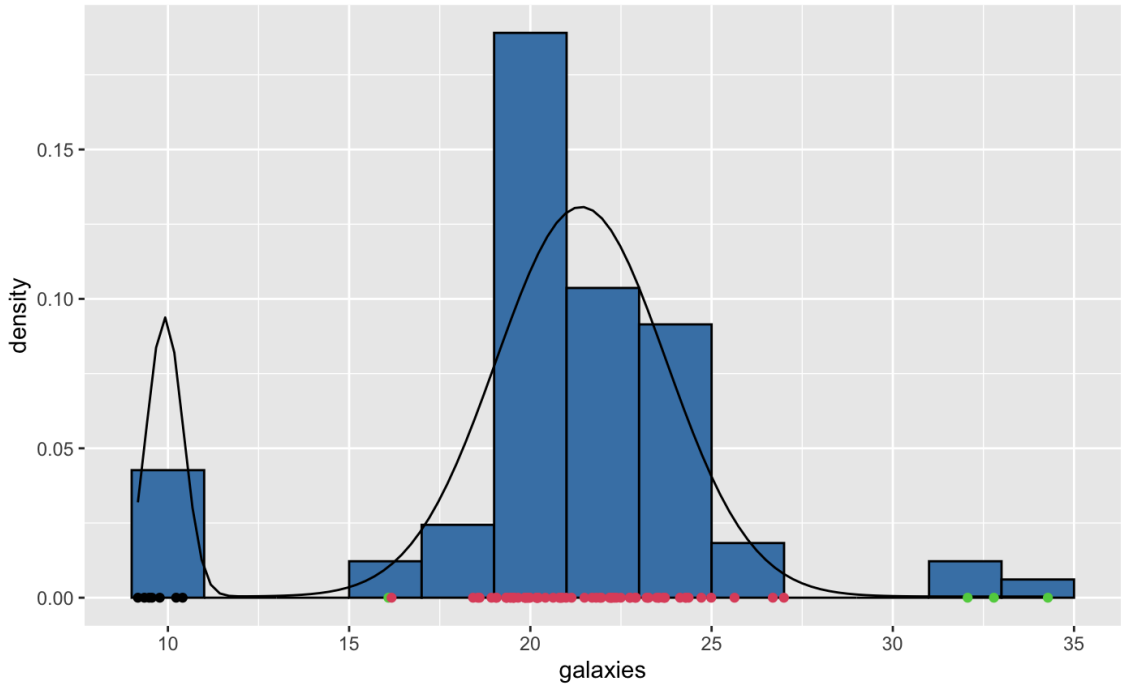

FIGURE 2.9 : Visualisation du clustering : répartition des galaxies, densités et groupes d'appartenance avec le package bayesmix

Cependant, bien que le clustering obtenu semble cohérent avec les attentes, cela ne garantit pas qu'il n'existe pas un clustering plus performant avec un nombre de groupes différent.

Chapitre 3

Détermination du nombre K de clusters et comparaison des résultats

Dans le chapitre précédent, nous avons détaillé les algorithmes que nous avons utilisés pour déterminer la distribution des paramètres θ en fonction des observations y, en fixant au préalable le nombre de clusters K. Nous avons choisi de travailler avec $K = 3$, car cette valeur semblait la plus plausible au regard de la visualisation des données, notamment à travers un histogramme, comme illustré à la figure 2.1, où l'on observe clairement trois groupes distincts. Toutefois, il est important de noter que cette estimation est biaisée et ne peut être justifiée d'un point de vue rigoureux. Bien que ce jeu de données soit relativement simple, cette approche ne serait pas viable pour un jeu de données à plus haute dimension, où une telle hypothèse sur le nombre de clusters serait bien moins défendable.

Ainsi, l'objectif principal de ce chapitre est de déterminer de manière plus objective le nombre de clusters "idéal" à partir des observations, en tenant compte de la variabilité et de la complexité du jeu de données. Pour se faire, nous allons explorer des méthodes plus robustes, en mettant l'accent sur l'estimation de K via des critères adaptés, avant de comparer les résultats obtenus avec ceux d'un autre algorithme de clustering répandu.

3.1 Estimation de K avec les critères BIC et ICL

Pour déterminer le nombre de groupes dans notre jeu de données, nous allons continuer à utiliser l'approche bayésienne. Ainsi, K devient une variable aléatoire dont la densité est donnée par $p(K)$. Nous pouvons alors écrire :

$$p(K|y) \propto p(y|K)\, p(K)$$

Nous ne possèdons aucune information sur le nombre de clusters potentiel. Ainsi pour notre a priori sur K nous avons choisi une loi uniforme. La densité de cette loi étant constante quelque soit les bornes choisies, on obtient :

$$p(K|y) \propto p(y|K)$$

Ainsi une estimation du nombre de groupes à choisir serait :

$$\begin{aligned}\hat{K}_{MAP} &= \arg\max_{K\in\mathbb{N}} \{p(y|K)\} \\ &= \arg\max_{K\in\mathbb{N}} \{2\log(p(y|K))\}\end{aligned}$$

Cependant ce dernier terme peut être complexe à calculer. C'est pourquoi nous avons utilisé une approximation de celui-ci, le BIC (Bayesian information criterion) proposé dans Bouveyron et al. (2019) défini comme suit :

$$BIC_K \;=\; 2\log(\; p(y|\hat{\theta}_{MAP}, K)\;) - \omega_K \log(N) \approx 2\log(p(y|K)) \tag{3.1}$$

Avec ω_K le nombre de paramètres libres et $p(y|\hat{\theta}_{MAP}, K)$ la vraisemblance de nos données sachant $\hat{\theta}_{MAP}$, et K

Ainsi on obtient finalement :

$$\hat{K}_{MAP} = \arg\max_{K\in\mathbb{N}} \{BIC_K\}$$

Nous avons utilisé un autre critère, expliqué dans Bouveyron et al. (2019) pour choisir le nombre de clusters, l'ICL (Integrated Complteted Likelihood) qui prend en compte l'estimation de **z** :

$$ICL_K = BIC_K \;-\; E(K) \tag{3.2}$$

Avec

$$E(K) = -\sum_{i=1}^{N}\sum_{k=1}^{K} \hat{z}_{ik}\log(\hat{z}_{ik}) \tag{3.3}$$

$\hat{z}_{ik}$ étant l'estimation de z_{ik} obtenue grâce au MAP.

Avec cette expression pour l'ICL, on remarque qu'il introduit une pénalisation supplémentaire. Cette pénalisation correspond à l'entropie moyenne des probabilités d'appartenance des individus aux clusters. Ainsi, l'ICL pénalise davantage les modèles où l'affectation des individus aux clusters est incertaine.

Pour sélectionner le meilleur modèle, nous prendrons celui qui maximise le BIC et l'ICL. Si ce ne sont pas les mêmes alors nous discuterons des résultats.

Dans la pratique, nous avons choisi de calculer ces critères pour un nombre de groupes K allant de 2 à 10. Nous avons fixé une limite à 10 groupes pour deux raisons. Tout d'abord, d'un point de vue visuel, il nous semblait peu probable que plus de 10 groupes émergent de ces données. Ensuite, après avoir analysé les résultats pour 10 clusters, nous avons constaté que certains groupes étaient vides. Dans 3.1 vous pouvez constater que les groupes 4 et 9 ne sont représentés. Cela indique que ces solutions ne sont pas optimales, et il est probable que ce problème persiste avec un nombre de clusters encore plus élevé.

Lors du calcul de $E(K)$ (3.3) nous avons été confronté au fait que $\hat{z}_{ik}$ soit nul et donc son logarithme infini. Ainsi pour éviter ce problème numérique nous avons rajouté l'epsilon machine qui est un nombre extrêmement petit ($\approx 2 \times 10^{-16}$).

N° groupe	1	2	3	4	5	6	7	8	9	10
Nombre d'individus	7	3	23	0	33	5	1	2	0	8

FIGURE 3.1 : Tableau récapitulatif de la répartition des individus avec K = 10

On peut observer les résultats des valeurs du BIC et de l'ICL dans 3.2. En rappelant que nous examinons les valeurs du BIC et de l'ICL pour un nombre de clusters allant de 2 à 10, nous pouvons conclure, d'après nos résultats, que le nombre de clusters à choisir serait 3. La visualisation de ces différents groupes est également disponible dans 3.3.

Nombre de groupe	2	3	4	5	6	7	8	9	10
BIC	-507.9	-468.5	-494.7	-483.9	-508.4	-500.1	-521.4	-534.3	-536.0
ICL	-507.9	-469.7	-506.3	-492.6	-523.5	-534.7	-576.3	-560.9	-590.1

FIGURE 3.2 : Valeurs du BIC et de l'ICL pour les différents nombre de groupes

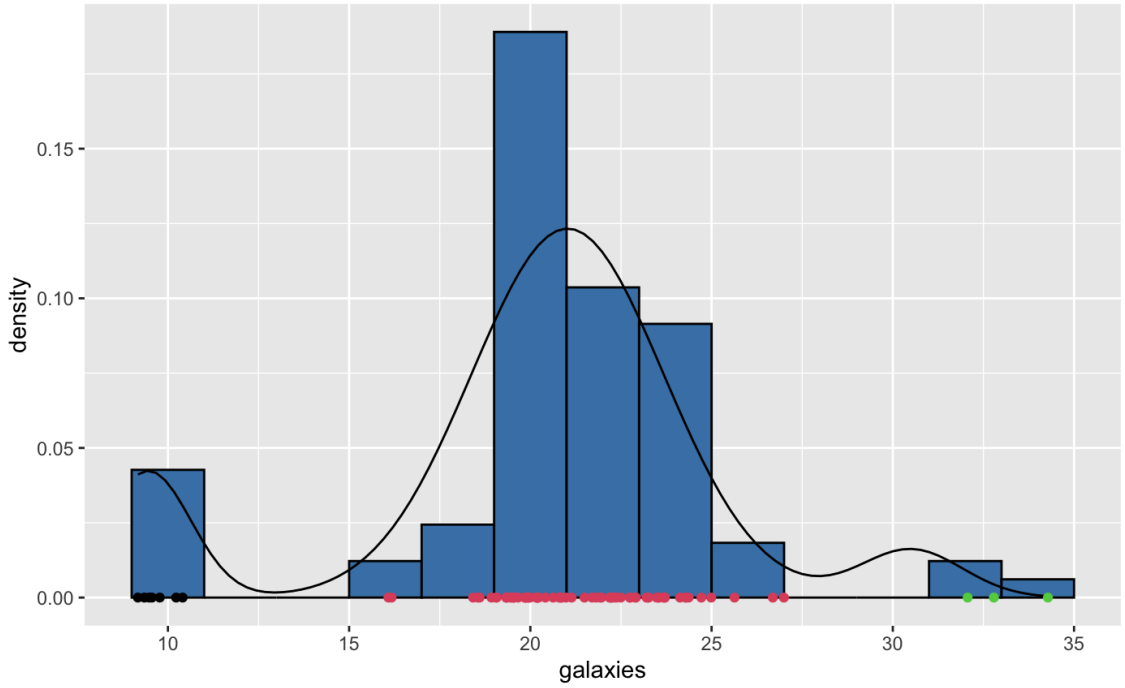

FIGURE 3.3 : Visualisation finale avec les groupes d'appartenance et la densité de la mixture gaussienne pour K = 3

Voici les valeurs de nos paramètres :

$$\hat{\pi} = (0.12513031, 0.81787585, 0.05699384)$$
$$\hat{\mu} = (9.45721, 21.02561, 30.50539)$$
$$\hat{\sigma^2} = (1.178187, 2.645985, 1.415630)$$

Nous pouvons constater que le nombre final de clusters choisi correspond au nombre choisi dans le chapitre précédent. Les résultats sont tout de même légèrement différents mais cela est

normal puisque l'échantillonneur de Gibbs est un processus aléatoire. Nous pouvons noter que le meilleur modèle est pour $K = 3$ mais que $K = 4$ ou $K = 5$ sont des modèles se démarquant aussi des autres avec des résultats visuellement cohérents (voir en 3.4 et 3.5).

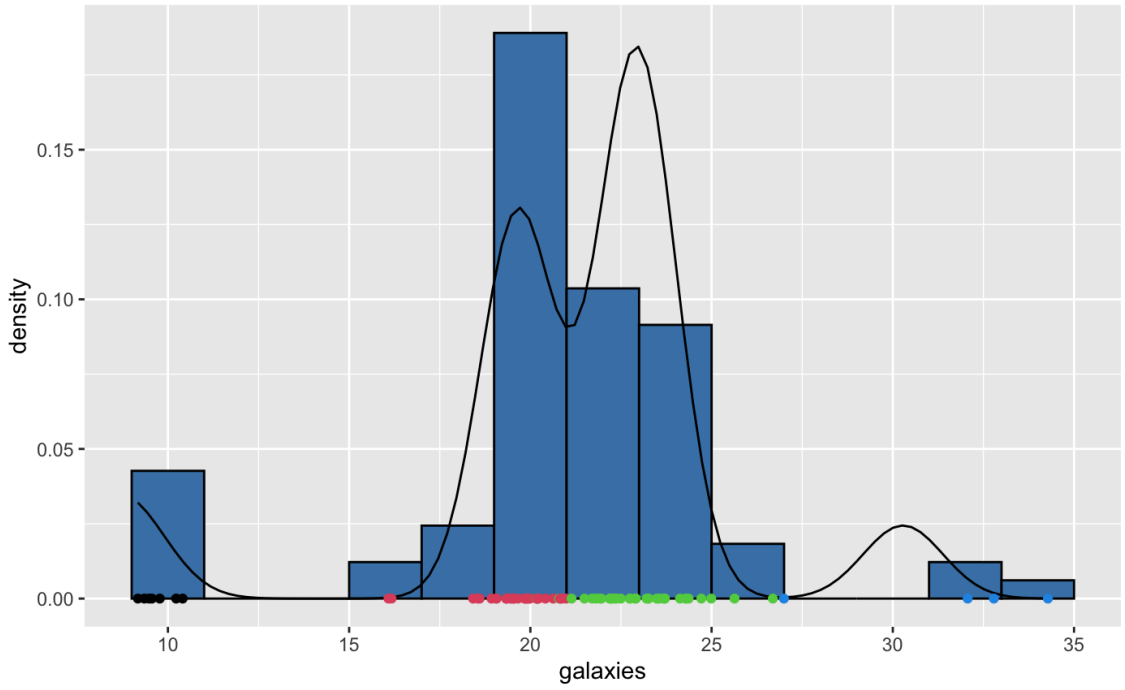

FIGURE 3.4 : Visualisation finale avec les groupes d'appartenance et la densité de la mixture gaussienne pour K = 4

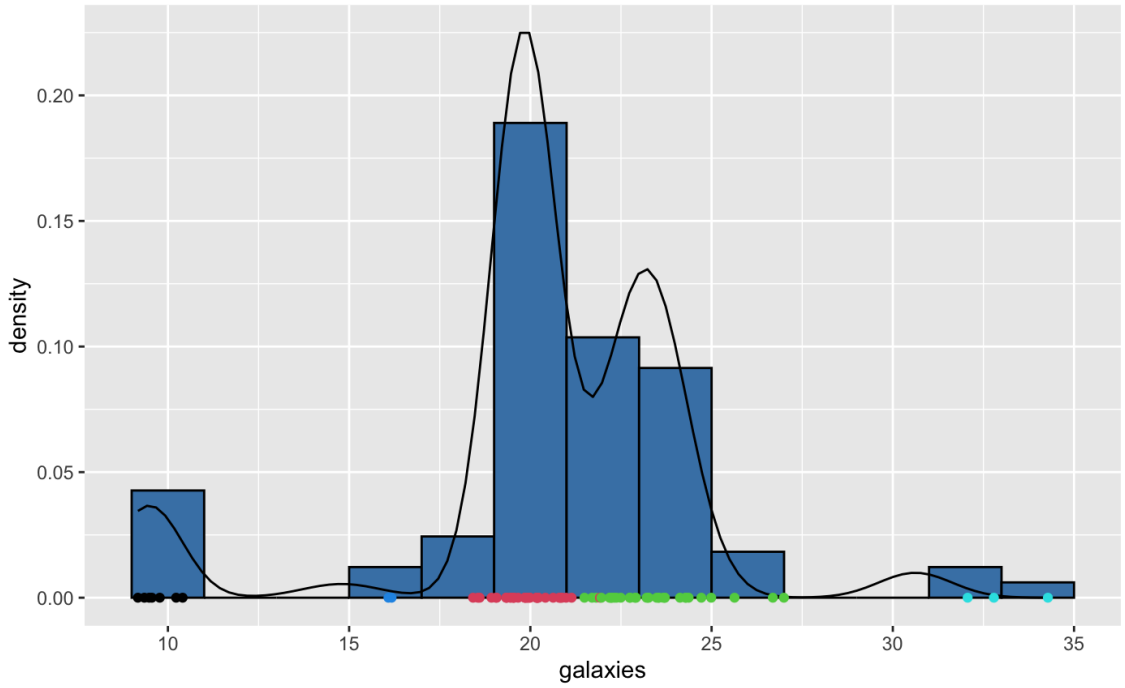

FIGURE 3.5 : Visualisation finale avec les groupes d'appartenance et la densité de la mixture gaussienne pour K = 5

3.2 Comparaisons

Dans cette section, nous allons comparer différentes méthodes de clustering appliquées au jeu de données des galaxies, en nous concentrant sur l'algorithme d'espérance-maximisation (EM) et en explorant les résultats obtenus dans plusieurs travaux scientifiques. L'algorithme EM, très utilisé pour l'estimation des paramètres dans les modèles de mélange gaussien, sera d'abord présenté de manière détaillée, en exposant son fonctionnement, ses étapes et ses résultats sur notre dataset. Ensuite, nous discuterons de deux articles scientifiques qui ont également abordé le problème du clustering sur ce même jeu de données, en utilisant des approches bayésiennes variées et en expérimentant avec différents a priori. Ces comparaisons permettront de mettre en lumière les points forts et les limites des approches que nous avons testées, tout en mettant en perspective la sensibilité des résultats selon les choix méthodologiques.

3.2.1 Algoritme EM

L'algorithme d'espérance-maximisation (EM) est une méthode itérative largement utilisée pour estimer les paramètres de modèles statistiques impliquant des variables latentes, c'est-à-dire des variables non observées mais essentielles à la structure du modèle.

Dans le cadre d'un modèle de mélange gaussien, chaque observation provient d'un des clusters modélisés par une distribution gaussienne. Cependant, l'appartenance à un cluster est inconnue, ce qui rend l'algorithme EM adapté pour estimer à la fois les paramètres du modèle (moyennes, variances et poids des composantes gaussiennes) et la probabilité d'appartenance de chaque observation à chaque cluster. On pourra noter que cette approche est très semblable à la notre avec Z et θ.

L'algorithme EM consiste en deux étapes principales, d'abord l'étape E (pour **Espérance**), soit l'estimation des probabilités d'appartenance des observations à chaque composante du modèle en fonction des paramètres actuels. Puis l'étape M (pour **Maximisation**), soit la mise à jour des paramètres du modèle (moyennes, variances et poids des clusters) pour maximiser la vraisemblance des données observées en utilisant les probabilités d'appartenance obtenues à l'étape E.

Ces étapes sont répétées de manière itérative jusqu'à convergence, permettant ainsi d'affiner les estimations des paramètres du modèle.

On reprend la définition du modèle de mélange Gaussien abordée en 2.1. Soit Z une matrice qui associe un cluster à chaque observation y_i :

$$\forall\, i \in [\![1, N]\!], \quad Z_i|\theta \sim \mathcal{M}(1, (\pi_1, ..., \pi_K))$$

Nous construisons ainsi la matrice $Z \in \mathcal{M}_{N\times K}(\mathbb{R})$ et notons **z** la matrice des observations des Z_i. Dans ce cas,

$$\forall\, i \in [\![1, N]\!], \quad y_i|Z_{ik} = 1 \sim \mathcal{N}(\mu_k, \sigma_k^2)$$

Notre θ est composé de 3 paramètres, μ, π et σ^2 estimé par Z selon : (avec n le nombre d'observations et n_k le nombre d'observations dans le cluster k)

$$\hat{\pi}_k = \frac{1}{n} \times \sum_{i=1}^{n} \hat{z}_{ik} \tag{3.4}$$

$$\hat{\mu}_k = \frac{1}{n_k} \times \sum_{i=1}^{n} \hat{z}_{ik} \times y_i \tag{3.5}$$

$$\hat{\sigma}_k^2 = \frac{1}{n_k} \times \sum_{i=1}^{n} \hat{z}_{ik} \times (y_i - \hat{\mu}_k)^2 \tag{3.6}$$

Avec n_k le nombre d'individus dans le cluster k.
A l'inverse, on met à jour les z_{ik} pour $1 \leq k \leq K$ à partir de θ grâce à la formule :

$$z_{ik} = \frac{\pi_k \times \mathcal{N}(y_i;\ \mu_k,\ \sigma_k^2)}{\sum_{p=1}^{K}(\pi_p \times \mathcal{N}(y_i;\ \mu_p,\ \sigma_p^2))} \tag{3.7}$$

Maintenant que l'on connait ces formules on peut schématiser l'algorithme EM comme suit :

Algorithm 4 Algorithme EM pour les modèles de mélange Gaussien

Initialisation des clusters (peut se faire avec un k-means)
Tant que la log-vraisemblance n'a pas convergée **faire :**
    Etape E : calculer $\hat{\theta}$ avec Z fixé avec (3.4), (3.5) et (3.6)
    Etape M : calculer Z avec θ fixé avec (3.7)
Fin Tant que;

Cependant nous n'allons pas le programmer nous même à la main car l'algorithme a déjà été implémenté sous R dans le package mclust (Scrucca et al. (2023)). L'application de cet algorithme sur nos observations révèle que le nombre optimal de clusters est de 4, en utilisant le critère BIC, comme indiqué dans la figure 3.6 ci-dessous.

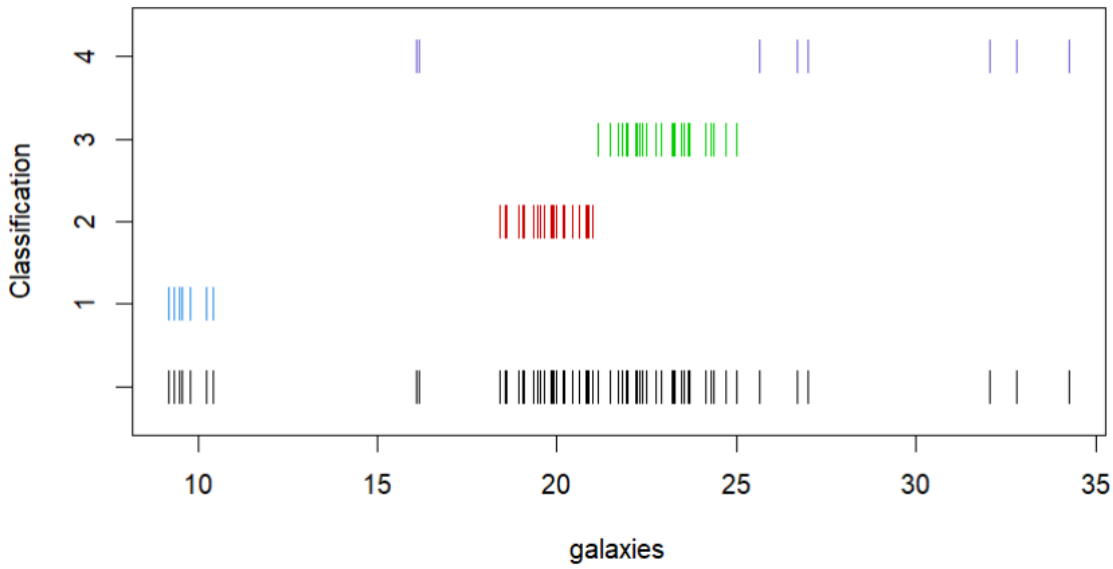

FIGURE 3.6 : Clusters assignés aux observations par la méthode mclust sur R

Le quatrième cluster semble particulièrement vaste, avec des données largement dispersées. Pour mieux comprendre la source de ce phénomène, examinons les paramètres associés aux clusters :

$$\hat{\pi} = (0.0844, 0.386, 0.370, 0.16)$$
$$\hat{\mu} = (9.71, 19.804, 22.878, 24.435)$$
$$\hat{\sigma^2} = (0.177, 0.435, 1.253, 34.122)$$

Il est clair que la variance du quatrième cluster est très élevée, avec $\sigma_4^2 = 34.122$. Cette grande variance entraîne une imprécision dans le résultat final du clustering, car les observations affectées à ce cluster le sont non pas en raison d'une forte probabilité d'appartenance, mais plutôt par un effet d'éparpillement. Cette situation est particulièrement manifeste sur la figure 3.7, où les observations du cluster 4 (représentées en bleu foncé) se trouvent sous une densité presque nulle, suggérant que ce cluster n'est pas bien défini.

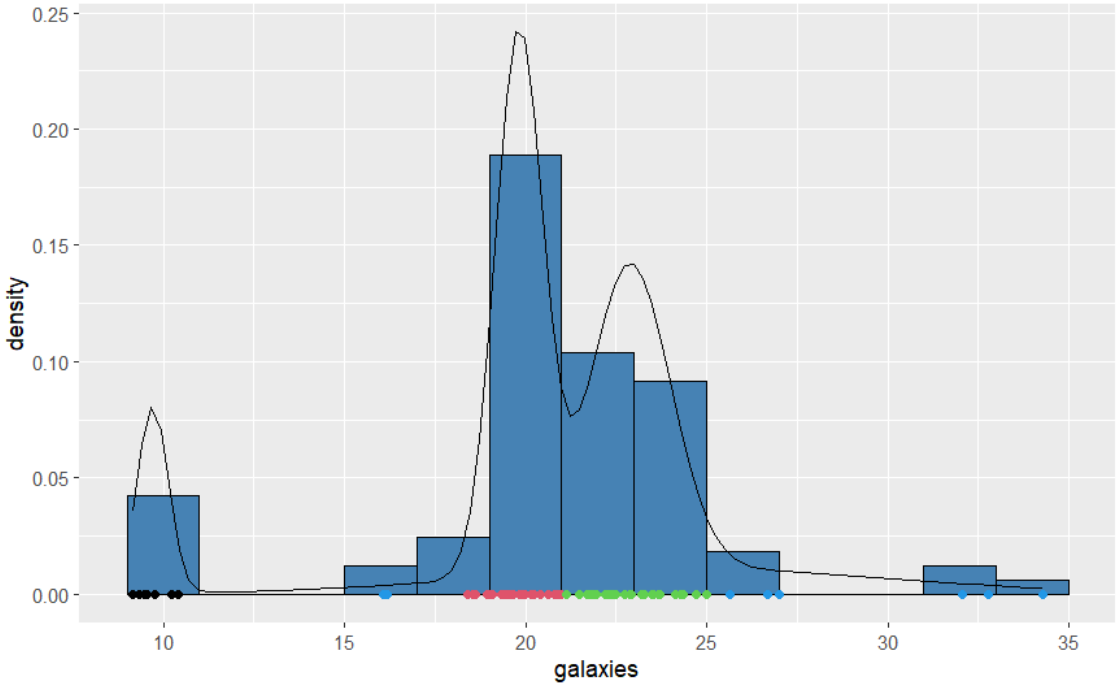

FIGURE 3.7 : Visualisation des clusters donnés par l'algorithme EM avec leurs densités

Il est important de souligner que l'algorithme d'espérance-maximisation (EM) ne repose sur aucun a priori concernant les données, ce qui signifie qu'il n'y a pas de paramètres explicites sur lesquels nous pouvons intervenir directement pour ajuster ou améliorer les résultats. En tenant compte de ces éléments, nous estimons que la méthode gaussienne, avec une approche plus adaptée et vraisemblable, constitue une meilleure solution. Bien qu'il n'existe pas de solution exacte pour ce type de problème, la solution obtenue par l'algorithme EM n'est pas pleinement satisfaisante. En effet, l'information contenue dans le quatrième cluster est minime, ce qui remet en question la pertinence de cette solution dans le cadre de notre analyse.

3.2.2 Papiers scientifiques

Nous allons discuter de deux papiers qui ont aussi fait du clustering sur le dataset des galaxies avec différentes méthodes, Grün et al. (2022) et Aitkin (2001).

Dans le premier article, les auteurs expérimentent divers a priori pour la distribution de K, incluant une loi uniforme, une loi de Poisson tronquée, une loi géométrique et une loi bêta-binomiale négative. Ils testent également différentes distributions pour la loi normale, la loi inverse-gamma et la loi de Dirichlet. Il est important de noter que les auteurs estiment K_+, qui représente le nombre de clusters avec au moins un individu, contrairement à K, qui inclut tous les clusters, même vides.

Les résultats montrent que le choix des a priori influence fortement le nombre de clusters, qui varie souvent entre 3 et 8. Toutefois, malgré ces variations, les résultats les plus fréquents suggèrent un nombre de clusters compris entre 3 et 5, ce qui est en accord avec nos propres observations, bien que les a priori diffèrent.

Le second article utilise également l'analyse bayésienne, mais combinée à une analyse de la vraisemblance plus classique. L'auteur en tire les conclusions suivantes : le nombre de clusters, déterminé par l'analyse de la vraisemblance, est de 3 ou 4, avec une grande certitude qu'il ne sera pas supérieur à 4. Concernant l'analyse bayésienne, il s'appuie sur les résultats d'autres chercheurs, qui, selon les a priori utilisés, trouvent des résultats très différents, variant entre 3 et 9 clusters. Les résultats de l'analyse bayésienne sont relativement semblables, mais l'auteur souligne que ces résultats peuvent varier considérablement en fonction des a priori choisis. Il est très critique vis-à-vis de la méthode bayésienne, la considérant comme sensible. Il estime que pour améliorer les méthodes bayésiennes, il est nécessaire de trouver un moyen d'en accroître la stabilité et d'améliorer l'interprétation des résultats.

Ainsi, les résultats présentés dans ce second article correspondent aux nôtres, mais il nous met également en garde, car les méthodes bayésiennes restent sensibles aux choix des a priori.

Conclusion

Dans ce projet, nous avons étudié une approche bayésienne pour effectuer du clustering sur le jeu de données des galaxies. Après avoir défini les notions fondamentales liées à la méthode bayésienne et au clustering, nous avons formalisé notre problématique en nous appuyant sur un modèle de mélange gaussien.

Nous avons d'abord exploré le cas où le nombre de clusters K est fixé, en définissant des lois a priori adaptées et en utilisant un échantillonneur de Gibbs pour estimer les paramètres du modèle. Nous avons ensuite traité le problème du label switching, qui peut fausser l'interprétation des résultats, en appliquant une méthode de correction basée sur l'algorithme de Stephens. Enfin, nous avons déterminé les valeurs optimales des paramètres grâce au calcul du MAP, obtenant ainsi les résultats pour K = 3. Ces valeurs ont permis d'obtenir une répartition des galaxies en trois groupes bien distincts, confirmant l'efficacité du modèle bayésien.

Dans un second temps, nous avons étudié la question de la sélection du nombre optimal de clusters. À l'aide des critères BIC et ICL, nous avons pu estimer K de manière plus rigoureuse. Nos résultats indiquent que le choix optimal du nombre de clusters est K = 3, en accord avec notre analyse initiale et la structure visuelle des données. Nous avons ensuite comparé notre approche bayésienne avec l'algorithme EM. Bien que ce dernier ait identifié quatre clusters, l'un d'eux présentait une variance très élevée, traduisant une classification moins précise et confirmant l'avantage d'une approche bayésienne intégrant des a priori.

Enfin, nous avons confronté nos résultats à ceux de plusieurs travaux scientifiques traitant du même jeu de données. Ces études ont montré que le choix des a priori et des méthodes de calcul pouvait avoir un impact significatif sur le nombre de clusters estimé. Toutefois, nos résultats, avec K = 3, sont cohérents avec les conclusions majoritaires de la littérature.

Ce projet nous a permis de mieux comprendre les atouts et les défis du clustering bayésien. Nos résultats confirment que cette approche est pertinente pour analyser des données complexes avec incertitude. Dans une perspective d'amélioration, une piste intéressante serait d'explorer des méthodes alternatives pour stabiliser davantage l'estimation de K et de tester notre approche sur des données plus complexes ou multidimensionnelles.

Bibliographie

Aitkin, M. (2001). Likelihood and bayesian analysis of mixtures. *Statistical Modelling 1*(4), 287–304.

Bouveyron, C., G. Celeux, T. B. Murphy, and A. E. Raftery (2019). *Model-based clustering and classification for data science : with applications in R*, Volume 50. Cambridge University Press.

Conn, P. B., D. S. Johnson, P. J. Williams, S. R. Melin, and M. B. Hooten (2018). A guide to bayesian model checking for ecologists. *Ecological Monographs 88*(4), 526–542.

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin (1995). *Bayesian data analysis*. Chapman and Hall/CRC.

Gruen, B. (2023). *bayesmix : Bayesian Mixture Models with JAGS*. R package version 0.7-6.

Grün, B., G. Malsiner-Walli, and S. Frühwirth-Schnatter (2022). How many data clusters are in the galaxy data set ? bayesian cluster analysis in action. *Advances in data analysis and classification 16*(2), 325–349.

Papastamoulis, P. (2016). label.switching : An R package for dealing with the label switching problem in mcmc outputs. *Journal of Statistical Software, Code Snippets 69*(1), 1–24.

Robert, C. P. et al. (2007). *The Bayesian choice : from decision-theoretic foundations to computational implementation*, Volume 2. Springer.

Scrucca, L., C. Fraley, T. B. Murphy, and A. E. Raftery (2023). *Model-Based Clustering, Classification, and Density Estimation Using mclust in R*. Chapman and Hall/CRC.

Stephens, M. (2000). Dealing with label switching in mixture models. *Journal of the Royal Statistical Society : Series B (Statistical Methodology) 62*(4), 795–809.

Venables, W. N. and B. D. Ripley (2002). *Modern Applied Statistics with S* (Fourth ed.). New York : Springer. ISBN 0-387-95457-0.

Wickham, H. (2016). *ggplot2 : Elegant Graphics for Data Analysis*. Springer-Verlag New York.