**Université de Montréal**

# Efficacité de l'algorithme EM en ligne pour des modèles statistiques complexes dans le contexte des données massives

par

**Yannick Martel**

Département de mathématiques et de statistique
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en statistique

18 novembre 2020

**Université de Montréal**
Faculté des arts et des sciences

Ce mémoire intitulé

# Efficacité de l'algorithme EM en ligne pour des modèles statistiques complexes dans le contexte des données massives

présenté par

**Yannick Martel**

a été évalué par un jury composé des personnes suivantes :

*Alejandro Murua*
(président-rapporteur)

*Florian Maire*
(directeur de recherche)

*Philippe Gagnon*
(membre du jury)

# Résumé

L'algorithme EM (Dempster *et al.*, 1977) permet de construire une séquence d'estimateurs qui converge vers l'estimateur de vraisemblance maximale pour des modèles à données manquantes pour lesquels l'estimateur du maximum de vraisemblance n'est pas calculable. Cet algorithme est remarquable compte tenu de ses nombreuses applications en apprentissage statistique. Toutefois, il peut avoir un lourd coût computationnel. Les auteurs Cappé et Moulines (2009) ont proposé une version en ligne de cet algorithme pour les modèles appartenant à la famille exponentielle qui permet de faire des gains d'efficacité computationnelle importants en présence de grands jeux de données. Cependant, le calcul de l'espérance a posteriori de la statistique exhaustive, qui est nécessaire dans la version de Cappé et Moulines (2009), est rarement possible pour des modèles complexes et/ou lorsque la dimension des données manquantes est grande. On doit alors la remplacer par un estimateur. Plusieurs questions se présentent naturellement : les résultats de convergence de l'algorithme initial restent-ils valides lorsqu'on remplace l'espérance par un estimateur ? En particulier, que dire de la normalité asymptotique de la séquence des estimateurs ainsi créés, de la variance asymptotique et de la vitesse de convergence ? Comment la variance de l'estimateur de l'espérance se reflète-t-elle sur la variance asymptotique de l'estimateur EM ? Peut-on travailler avec des estimateurs de type Monte-Carlo ou MCMC ? Peut-on emprunter des outils populaires de réduction de variance comme les variables de contrôle ? Ces questions seront étudiées à l'aide d'exemples de modèles à variables latentes. Les contributions principales de ce mémoire sont une présentation unifiée des algorithmes EM d'approximation stochastique, une illustration de l'impact au niveau de la variance lorsque l'espérance a posteriori est estimée dans les algorithmes EM en ligne et l'introduction d'algorithmes EM en ligne permettant de réduire la variance supplémentaire occasionnée par l'estimation de l'espérance a posteriori.

**Mots-clés** : Algorithme EM, approximation stochastique, algorithme en ligne, réduction de variance, statistique computationnelle

# Abstract

The EM algorithm Dempster *et al.* (1977) yields a sequence of estimators that converges to the maximum likelihood estimator for missing data models whose maximum likelihood estimator is not directly tractable. The EM algorithm is remarkable given its numerous applications in statistical learning. However, it may suffer from its computational cost. Cappé et Moulines (2009) proposed an online version of the algorithm in models whose likelihood belongs to the exponential family that provides an upgrade in computational efficiency in large data sets. However, the conditional expected value of the sufficient statistic is often intractable for complex models and/or when the missing data is of a high dimension. In those cases, it is replaced by an estimator. Many questions then arise naturally : do the convergence results pertaining to the initial estimator hold when the expected value is substituted by an estimator ? In particular, does the asymptotic normality property remain in this case ? How does the variance of the estimator of the expected value affect the asymptotic variance of the EM estimator ? Are Monte-Carlo and MCMC estimators suitable in this situation ? Could variance reduction tools such as control variates provide variance relief ? These questions will be tackled by the means of examples containing latent data models. This master's thesis' main contributions are the presentation of a unified framework for stochastic approximation EM algorithms, an illustration of the impact that the estimation of the conditional expected value has on the variance and the introduction of online EM algorithms which reduce the additional variance stemming from the estimation of the conditional expected value.

**Keywords** : EM algorithm, stochastic approximation, online algorithm, variance reduction, computational statistics

# Table des matières

# Liste des tableaux

# Table des figures

# Liste des sigles et des abréviations

| | |
|---|---|
| EM | Espérance et maximisation, de l'anglais *Expectation Maximisation* |
| iid | Indépendantes identiquement distribuées |
| MAD | Écarts absolus médians, de l'anglais *Median Absolute Deviations* |
| MCMC | Monte-Carlo par chaînes de Markov, de l'anglais *Markov Chain Monte Carlo* |
| SAEM | Algorithme EM en ligne par approximation stochastique, de l'anglais *Online Stochastic Approximation EM* |
| Var | Variance |
| ZVCV | Variable de contrôle à variance zéro, de l'anglais *Zero variance control variates* |

# Remerciements

La rédaction de ce mémoire de maîtrise n'aurait pas été possible sans l'appui de plusieurs personnes que je souhaite remercier. Tout d'abord, mon directeur de recherche, Florian Maire, m'a offert un soutien inconditionnel pendant la dernière année et demie. Sa disponibilité, sa générosité, sa curiosité et sa perspicacité ont grandement facilité ma tâche. Je lui suis également reconnaissant pour le soutien financier. Je voudrais aussi remercier les professeur·e·s de statistique du département, qui m'ont aidé à cultiver une affection pour la statistique grâce à leur enseignement et à leur passion. Merci à Tony et Sébastien qui m'ont beaucoup aidé dans la réalisation de travaux au baccalauréat, une époque où la maîtrise me semblait parfois une fantaisie inatteignable. Ce fut également un plaisir de travailler avec plusieurs statisticiens et statisticiennes au cours de la maîtrise. Je salue mes ami·e·s qui ont été d'une précieuse aide du fait de leur écoute, de leur humour et de leur soutien. Merci aussi à ma copine, Justine, qui m'a accompagné et encouragé tout au long de la recherche et la rédaction, crise sanitaire en toile de fond. Finalement, la plus grande contribution revient naturellement à mes parents. Il est difficile de synthétiser tout ce qu'ils ont fait au cours de mon existence pour me permettre d'aboutir au dépôt de ce mémoire, leurs contributions étant trop étendues et diverses. En bref, ils m'ont donné suffisamment d'impulsions au cours du dernier quart de siècle afin que je puisse voler de mes propres ailes et planer sur ma soif d'apprendre.

# Introduction

L'optimisation est l'un des champs les plus importants en mathématiques. Que ce soit en mathématiques appliquées, en statistique ou en apprentissage automatique, l'optimisation occupe un rôle central. Dans les circonstances les plus simples, c'est-à-dire, en présence d'un optimum global pour une fonction lisse $\ell : \Theta \to \mathbb{R}$, la méthode des points critiques permet de trouver les optima en résolvant l'équation $\nabla_\theta \ell(\theta) = 0_p$. Quand le système se résout analytiquement, on trouve aisément l'optimum. Quand le système n'est pas résoluble, des méthodes numériques permettent de trouver les points critiques. Un choix populaire parmi ces méthodes est l'algorithme du gradient qui permet de minimiser une fonction lisse en suivant un chemin qui va dans le sens inverse du gradient

$$\theta_t = \theta_{t-1} - \gamma \nabla_\theta \ell(\theta_{t-1}),$$

où le pas d'apprentissage $\gamma \in \mathbb{R}$. Cette méthode récursive est en fait une discrétisation de l'équation différentielle

$$\frac{\mathrm{d}\theta(t)}{\mathrm{d}t} = -\nabla_\theta \ell(\theta(t)). \tag{1}$$

Dans un contexte de données massives où la fonction objective est de la forme $\sum_{i=1}^n \ell_{i,n}(\theta)$, une approximation stochastique de l'algorithme du gradient qui consiste à estimer le gradient par $\nabla \ell_{l,n}(\theta)$, où $l$ suit une uniforme sur $(1, \ldots, n)$, permet de trouver le minimum. La méthode d'approximation stochastique dont il est question est celle de Robbins-Monro. L'algorithme du gradient stochastique, qui repose sur cette procédure, a été une avancée très importante en apprentissage automatique (Bottou, 2010). Sa complexité à chaque itération est d'ordre 1, ce qui est avantageux dans le contexte des données massives. Par contraste, l'algorithme du gradient a une complexité d'ordre $n$ à chaque itération, ce qui est prohibitif dans un tel contexte. Le problème d'optimisation auquel on s'intéresse dans ce mémoire est la maximisation de la fonction de vraisemblance qui peut être écrite sous la forme $\sum_{i=1}^n \ell_{i,n}(\theta)$. Dans le contexte des données massives, l'algorithme du gradient stochastique s'avère utile pour des problèmes de maximisation de vraisemblance.

Le cadre d'intérêt dans ce mémoire est celui des modèles à variables latentes. Ainsi, la fonction de vraisemblance à maximiser est basée sur des données incomplètes. Souvent, cette dernière n'est pas connue explicitement et ne peut donc pas être maximisée à l'aide de

méthodes du gradient telles que celle de l'algorithme du gradient. L'algorithme EM, dont « EM » signifie espérance et maximisation, est un algorithme conçu expressément pour l'optimisation d'une telle fonction de vraisemblance (Dempster *et al.*, 1977). De manière analogue à l'algorithme du gradient, l'algorithme EM n'est pas bien adapté au contexte des données massives, car étant donné que chacune des itérations traite toutes les données, elles sont accompagnées d'un lourd coût computationnel. Afin de contourner cette difficulté, l'algorithme EM en ligne repose sur la méthode d'approximation stochastique de Robbins-Monro. On peut voir l'algorithme EM en ligne comme l'analogue de l'algorithme du gradient stochastique dans le cas où la fonction de vraisemblance observée n'est pas connue. Dans ce mémoire, « en ligne » signifie que chaque donnée est disponible une seule fois et disparaît après avoir été traitée. En pratique, la faisabilité de ces algorithmes en ligne dépendent de la possibilité de calculer l'espérance a posteriori de la statistique exhaustive, ce qui est rarement possible pour des modèles complexes.

Une façon de schématiser ces algorithmes est de les voir comme des discrétisations bruitées d'une équation différentielle ordinaire. Dans ce qui précède, on a décrit différents niveaux de discrétisation et d'estimation qui donnent différents degrés d'implémentabilité, et aussi, différents niveaux de précision dans l'inférence. L'algorithme du gradient discrétise l'équation différentielle, l'équation 1. L'algorithme du gradient stochastique utilise un estimateur du gradient en ne tenant compte d'une seule observation par itération. On peut donc voir l'algorithme du gradient stochastique comme une version bruitée de l'algorithme du gradient, ce qui ajoute donc un bruit par rapport à la discrétisation faite par l'algorithme du gradient à l'origine. Pour des raisons qui seront étayées plus loin, l'algorithme EM en ligne converge sous des conditions de régularité vers un ensemble de points qui n'est pas exactement le maximum de vraisemblance, mais vers un ensemble de points néanmoins utiles pour l'inférence. Lorsque le calcul de l'espérance est impossible, l'espérance a posteriori peut être estimée par un estimateur MC (Monte-Carlo) ou MCMC (Monte-Carlo par chaînes de Markov), ce qui ajoute un niveau de bruit supplémentaire par rapport à l'algorithme EM en ligne.

Chaque niveau d'approximation affecte la trajectoire de la discrétisation initiale, et donc potentiellement, sa convergence. L'estimation supplémentaire, quant à elle, se répercute en une augmentation de variance des estimateurs. Cette augmentation de variance motive l'introduction de techniques de réduction de variance qui consiste en l'emploi judicieux d'informations auxiliaires afin de construire un estimateur moins variable. Une manière d'atteindre cet objectif est de créer, à l'aide d'une variable de contrôle, un autre estimateur dont l'espérance est la même que la variable d'intérêt, mais dont la variance est réduite (possiblement jusqu'à 0). Lorsque la variance est réduite à zéro, l'estimateur donne la vraie valeur du paramètre, sans incertitude. La variance est réduite grâce à la covariance entre la variable à réduire et une variable corrélée dont on aurait connaissance grâce à l'information auxiliaire. Par exemple, l'algorithme SVRG (gradient stochastique à variance réduite) est un algorithme

qui réduit la variance de l'algorithme du gradient stochastique. Cette méthode passe à travers des données de manière aléatoire et met à jour la variable de contrôle après chaque « époque » (Johnson et Zhang, 2013), c'est-à-dire après avoir traité toutes les données. L'algorithme EM à variance réduite de Chen *et al.* (2018) s'inspire de l'algorithme SVRG. Ces algorithmes ne sont cependant pas applicables au cadre purement en ligne qui nous intéresse dans ce mémoire. On va plutôt introduire des techniques de réduction de variance pour des méthodes Monte-Carlo ou MCMC puisque les algorithmes EM en ligne approximatifs qu'on va présenter contiennent des espérances Monte-Carlo ou MCMC. L'une des visées principales de ce mémoire est de réduire la variance de l'estimateur du paramètre d'intérêt lorsqu'on estime l'espérance a posteriori des statistiques exhaustives dans les algorithmes EM en ligne.

Les sections de ce mémoire seront divisées de la manière suivante. Le chapitre 1 se veut une revue succincte et non-exhaustive des thèmes et concepts abordés dans le reste du mémoire. Le chapitre 2 présente certaines variations de l'algorithme EM afin d'illustrer comment on construit un algorithme compatible avec le contexte des données massives à partir de l'algorithme EM de Dempster *et al.* (1977). Le chapitre 3 montre les impacts de la discrétisation et de l'estimation des espérances a posteriori sur la variance des estimateurs des paramètres du modèle d'intérêt. En particulier, on travaille sur des modèles de mélange à variable latente discrète et on présente des exemples de modèles à variable latente continue. Finalement, au chapitre 4, l'effet de trois approches de réduction de variance dans les algorithmes EM en ligne est étudié pour des modèles à variable latente de complexités variables.

Ce mémoire contient trois contributions principales : une présentation de la famille des algorithmes EM en ligne dans un contexte unifié, l'illustration de l'impact de l'estimation de l'espérance a posteriori sur l'efficacité d'algorithmes EM en ligne Monte-Carlo et MCMC, et finalement, l'impact de l'utilisation d'estimateurs variance zéro sur l'efficacité dans les algorithmes EM en ligne Monte-Carlo et MCMC. On montre, au travers d'exemples, qu'il existe des situations où il y a un intérêt computationnel à favoriser des algorithmes EM en ligne Monte-Carlo ou MCMC à variance réduite lorsque les algorithmes EM en ligne Monte-Carlo ou MCMC sans variance réduite exigent un grand nombre de réalisations afin d'avoir une performance comparable aux algorithmes EM en ligne Monte-Carlo ou MCMC de réduction de variance.

# Chapitre 1

# Notions préliminaires

Les sections suivantes regroupent un éventail de concepts abordés dans ce mémoire sans pour autant constituer une liste exhaustive des thèmes abordés dans ce mémoire. Les sujets sont présentés sommairement. Des références sont proposées pour une couverture plus exhaustive des détails qui sont omis.

## 1.1. Algorithme d'approximation stochastique de Robbins-Monro

Soit $\Theta \subseteq \mathbb{R}^p$, introduisons la fonction $M : \Theta \to \mathcal{M} \subseteq \mathbb{R}^d$, dont l'image pour chaque $\theta$ n'est pas connue explicitement. L'algorithme de Robbins-Monro (Robbins et Monro, 1951) permet de résoudre des équations du type

$$M(\theta) = 0_d, \tag{2}$$

lorsqu'un estimateur sans biais de $M(\theta)$ existe pour chaque $\theta$ et est calculable , où $\theta \in \Theta$ et $\theta^\star \in \Theta$ est la solution unique de l'équation 2. C'est une approche récursive qui construit une séquence d'approximations des racines en exploitant les estimateurs. Plus précisément, soit $\{\xi_t\}_{t\in T}$, un processus aléatoire sur l'espace mesurable $(\Xi,\mathcal{B}(\Xi))$ où $T = \mathbb{N}^+$. On adoptera désormais la notation $\{\xi_t\}$ qui omet l'indice pour les suites. On désigne par $H(\theta,\xi)$ l'estimateur de $M(\theta)$ avec $H : \Theta \times \Xi \to \mathbb{R}^d$. La récursion de l'algorithme de Robbins-Monro est

$$\theta_t = \theta_{t-1} - \gamma_{n-1} H\left(\theta_{t-1},\xi_{t-1}\right). \tag{3}$$

Par exemple, lorsque $M$ est le gradient d'une fonction, cette récursion coïncide avec l'algorithme du gradient stochastique. Lorsque certaines conditions sont vérifiées, la séquence $\{\theta_t\}$ converge la racine de $M$. Il existe plusieurs types de résultats garantissant la convergence. Ceux-ci diffèrent suivant les hypothèses ou le mode de convergence. Les conditions

pour obtenir la convergence au sens du théorème 2 de Robbins et Monro (1951) sont les suivantes :

— La fonction $\sup_{\xi\in\Xi} H(\theta,\xi)$ est bornée uniformément,

— La fonction $M(\theta)$ est non décroissante,

— $M'(\theta^\star)$ existe et elle est positive,

— La séquence $\{\gamma_t\}$ est décroissante, positive et telle que $\sum_{n=0}^{\infty}\gamma_n = \infty$ et $\sum_{n=0}^{\infty}\gamma_n^2 < \infty$.

Sous ces conditions, la suite $\{\theta_t\}$ converge en probabilité vers $\theta^*$ (Robbins et Monro, 1951). D'autres résultats dans Kushner et Yin (2003) établissent la convergence presque sûre.

**Exemple 1.1.1.** *Soit* $M(\theta) = \log(\theta) + 5\theta - \log e^4 - 5e^4$ *où* $\Theta = (0{,}100]$ *et* $H(\theta,\xi) = \log(\theta) + 5\theta - \log e^4 - 5e^4 + \xi$. *On voudrait trouver* $\theta = \theta^*$ *tel que*

$$M(\theta) = \log(\theta) + 5\theta - \log(e^4) - 5e^4 = 0.$$

*On pose* $\gamma_n = 1/n^{0{,}8}$ *dans la récursion 3. On génère une suite de bruits iid, indépendants et identiquement distribués,* $\{\xi_n\}$ *suivant une* Uniforme$(-3{,}3)$ *de sorte qu'on dispose d'une suite* $\{H(\theta_t,\xi_t)\}$ *de taille* $10^5$. *Il est facile de vérifier que* $\mathbb{E}_\xi[H(\theta,\xi)] = M(\theta)$. *La solution étant* $\theta^* = e^4 = 54{,}59815$, *l'algorithme oscille de plus en plus près de* $e^4$ *au fil des itérations. La figure 1.1 illustre la convergence de l'algorithme quand on initialise la séquence avec* $\theta_0 = 50$.

## 1.2. Vraisemblance maximale

### 1.2.1. Principe

En statistique paramétrique, on attribue des lois de probabilité aux phénomènes aléatoires. Ces lois comportent un paramètre $\theta \in \Theta \subseteq \mathbb{R}^p$, où $p$ est la dimension du paramètre. Ces paramètres sont fixes et inconnus. Habituellement, ils concernent la position, la forme ou l'échelle de ces distributions. Par exemple, la densité de la loi normale univariée contient un paramètre $\theta = (\mu,\sigma^2) \in \mathbb{R}\times\mathbb{R}^+$ où $\mu$ est le paramètre de position et $\sigma^2$ est le paramètre d'échelle. Ces phénomènes aléatoires sont caractérisés par une variable aléatoire $Y$ associée à une loi de probabilité $\mathbb{P}_\theta$. Lorsqu'on recueille des observations liées à un phénomène, on voudrait ajuster un modèle statistique afin de faire de l'inférence sur le phénomène sous-jacent. Il existe quelques manières d'atteindre cet objectif. La plus populaire, et la manière sur laquelle on se penche dans ce mémoire, est l'estimation par maximum de vraisemblance.

On suppose que les données qu'on veut analyser sont $y_1,\ldots,y_n \in \mathcal{Y}$. On suppose que ces données sont des réalisations d'une collection de variables aléatoires $Y_1,\ldots,Y_n$ iid (indépendantes identiquement distribuées) sur un espace mesurable $(\mathcal{Y},\mathcal{B}(\mathcal{Y}))$ de loi $\pi(y)$ par

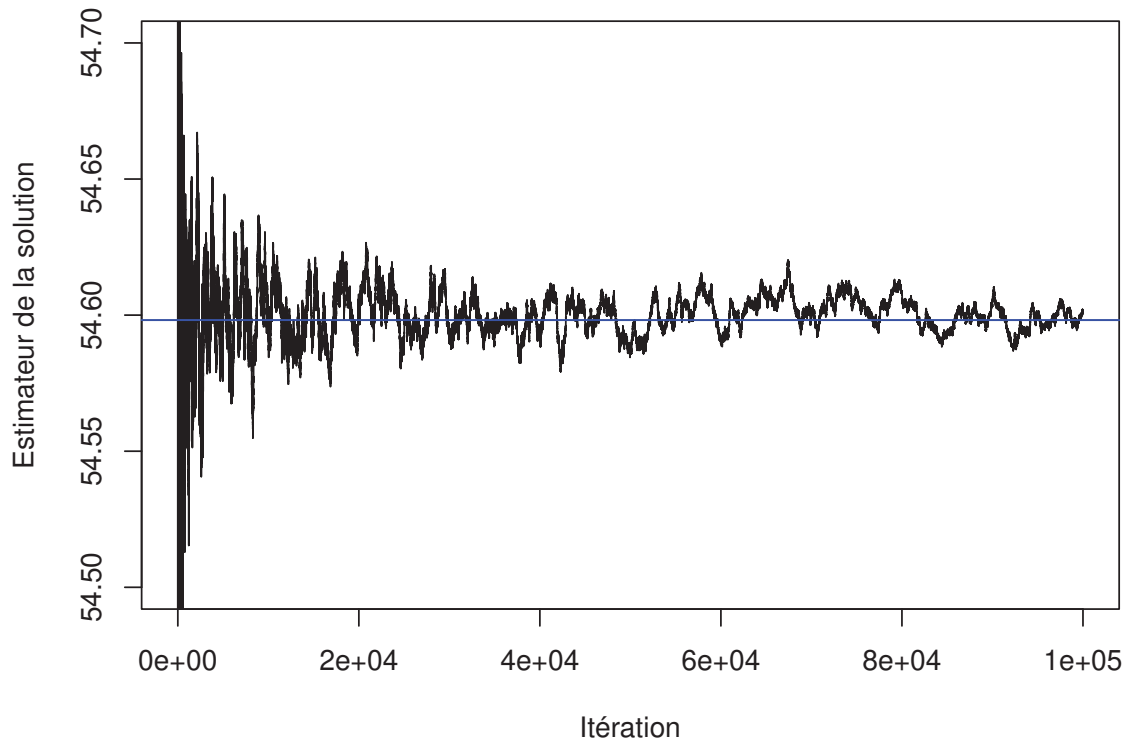


**Figure 1.1.** Convergence de l'algorithme Robbins-Monro pour l'exemple 1.1 où la ligne noire représente l'évolution de $\theta_n$ et la ligne bleue représente la solution de l'équation 2. La trajectoire est conditionnelle au point de départ.

rapport à la mesure sigma finie de référence $\nu$ sur $(\mathcal{Y}, \mathcal{B}(\mathcal{Y}))$. On travaille dans un cadre paramétrique et on spécule que la loi peut être approchée par $g(y;\theta)$, que l'on appelle aussi $g_\theta$, par rapport à la mesure sigma finie de référence notée $\nu(\mathrm{d}y)$ où $\theta = (\theta_1, \ldots, \theta_p) \in \Theta$. L'ensemble $\Theta \in \mathbb{R}^p$ est le support des paramètres. L'entier $p$ représente sa dimension, soit le nombre de paramètres. Dans la suite, si $Y$ est discrète, $\nu$ sera la mesure de comptage. Si $Y$ est continue, $\nu$ sera la mesure de Lebesgue.

La fonction de vraisemblance pour des variables aléatoires iid est définie par :

$$\mathcal{L}_n(\theta; y_1, \ldots, y_n) = \prod_{i=1}^{n} g(y_i;\theta).$$

L'estimateur de vraisemblance maximale consiste en la valeur maximisant $\mathcal{L}_n(\theta; y_1, \ldots, y_n)$,

$$\hat{\theta}_n = \underset{u\in\Theta}{\operatorname{argmax}}\ \mathcal{L}_n(u; y_1, ..., y_n).$$

La fonction logarithmique est une fonction monotone. Par conséquent, l'appliquer à la fonction de vraisemblance ne change pas la valeur du maximum. On définit la log-vraisemblance comme

$$\ell_n(\theta) \overset{\text{déf}}{=} \log \mathcal{L}_n(\theta; y_1, ..., y_n).$$

On peut écrire la log-vraisemblance comme

$$\ell_n(\theta) = \sum_{i=1}^{n} \log g\left(y_i; \theta\right).$$

L'estimateur du maximum de vraisemblance de $\theta$ est la solution de

$$\nabla_\theta \ \mathcal{L}\left(\theta; y_1, ..., y_n\right) = 0_p. \tag{4}$$

Lorsque $\Theta$ est un ensemble compact et que la fonction $\ell_n$ est continue en tout point de $\Theta$, il existe un ou plusieurs estimateurs du maximum de vraisemblance (Ferguson, 1996). Lorsque $\Theta$ est un ensemble convexe et que la fonction de vraisemblance est concave, l'estimateur de vraisemblance maximale est un maximum. Dans la famille exponentielle canonique, les fonctions de log-vraisemblance sont concaves et possèdent un unique maximum, car la matrice $\nabla^2_\theta \ell_n$ est définie négative (Lehmann et Casella, 2006). Comme la matrice de covariance est définie positive, le négatif de la matrice de covariance est définie négative. C'est cet estimateur défini à l'équation 4 qu'on utilise dans le présent mémoire, car dans la plupart des situations abordées, les vraisemblances appartiennent à la famille exponentielle, et l'équation 4 a une racine unique. Le maximum pourrait aussi être situé sur une borne du support de $\Theta$ dans certaines situations quand le support de $\Theta$ est ouvert mais borné (Casella et Berger, 2002).

Le théorème 17 de Ferguson (1996) montre que lorsque $\Theta$ est compact, sous certaines conditions de régularité

$$\hat{\theta}_n \overset{p.s.}{\to} \theta.$$

Lorsque le support est un ensemble ouvert, ce qui est souvent le cas pour les paramètres d'échelle, certaines conditions évoquées précédemment comme la compacité ne sont plus forcément remplies. Dans Ferguson (1996), on montre que sous certaines conditions, l'une des racines de 4 converge vers l'estimateur de vraisemblance maximale. Soit l'information de Fisher

$$I\left(\theta\right) \overset{\text{déf}}{=} \mathbb{E}\left[\left(\nabla_\theta \log g\left(Y; \theta\right)\right)\left(\nabla_\theta \log g\left(Y; \theta\right)\right)^T\right]. \tag{5}$$

Le théorème 18 de Ferguson (1996) montre qu'il est possible, sous des conditions générales, d'obtenir la normalité asymptotique même lorsque $\Theta$ est ouvert :

$$\sqrt{n}\left(\hat{\theta}_n - \theta\right) \overset{D}{\to} \mathcal{N}\left(0, I(\theta)^{-1}\right).$$

### 1.2.2. Algorithmes du gradient

Lorsque le système $\nabla_\theta \ell_n(\theta) = \mathbf{0}_p$ n'a pas de solution explicite, on peut utiliser des méthodes numériques pour trouver la solution. Une méthode utile pour identifier des estimateurs de vraisemblance maximale est l'algorithme du gradient qui permet d'obtenir une convergence d'ordre linéaire vers le minimum, ce qui est justifié dans la section 9.3.1 de Boyd *et al.* (2004). En pratique, l'algorithme détermine le minimum du négatif de la fonction de vraisemblance, soit son maximum. Cette méthode est utile lorsqu'on dispose de toutes les données et qu'on peut évaluer les gradients point à point. Soit $\theta_0 \in \Theta$, la récursion $t$ de l'algorithme du gradient est la suivante :

$$\theta_t = \theta_{t-1} - \gamma_t \nabla \ell_n(\theta_{t-1}), \tag{6}$$

où $\{\gamma_t\}$ est une suite de nombres positifs décroissants qui peut être vue comme le pas d'apprentissage.

**Remarque 1.2.1.** *Conditionnellement aux données et au point initial $\theta_0$, la suite $\{\theta_t\}$ est déterministe.*

**Remarque 1.2.2.** *Quand $\ell_n(\theta)$ est convexe sur $\Theta$ et que sa dérivée d'ordre deux existe, la suite $\{\theta_t\}$ converge vers un minimum (Boyd* et al.*, 2004).*

Dans le cas où les données deviennent disponibles au fur et à mesure, il existe une version en ligne de l'algorithme du gradient, l'algorithme du gradient stochastique. Cette procédure est efficace lorsqu'il y a un très grand nombre de données limitant l'efficacité computationnelle de l'algorithme du gradient. Tout comme l'algorithme du gradient, l'algorithme du gradient stochastique converge également vers un minimum, mais en traitant une réalisation à la fois. La forme de la log-vraisemblance se prête particulièrement bien à cet algorithme qui exige des fonctions de la forme

$$\ell_n(\theta) = \sum_{i=1}^{n} \ell^{(i)}(\theta) = \sum_{i=1}^{n} \log g(y_i; \theta).$$

où $\ell^{(i)}(\theta) = g(y_i; \theta)$. Soit la suite décroissante de nombre positifs $\{\gamma_t\}$ qui peut être vue comme un pas d'apprentissage, et $\theta_0$, un point initial et l'indice $i_t$ tiré aléatoirement entre 1 et $n$, la récurrence de l'algorithme du gradient stochastique est

$$\theta_t = \theta_{t-1} - \gamma_t \nabla_\theta \ell^{(i_t)}(\theta_{t-1}). \tag{7}$$

La suite $\{\theta_t\}$ générée par 7 converge presque sûrement vers un minimum sous certaines conditions (Bottou, 2010; Leluc et Portier, 2020). L'algorithme du gradient stochastique est computationnellement efficace, mais converge généralement lentement (Leluc et Portier, 2020). L'équation 7 ressemble beaucoup à l'équation 3. La récursion de Robbins-Monro, l'équation 3, met à jour itérativement la racine d'une fonction pour laquelle on possède

seulement une version bruitée. Dans le cas de l'algorithme du gradient stochastique, on trouve la racine à l'aide d'une séquence de versions bruitées du gradient.

**Remarque 1.2.3.** *L'algorithme du gradient stochastique est un cas particulier de l'algorithme de Robbins-Monro. Soit $\delta_{y_i}$, la masse de Dirac en $y_i$. Soit $M(\theta) = \mathbb{E}[\nabla_\theta \log(g(\xi;\theta))]$ où $\mathbb{E}$ est l'espérance sous la loi $\xi \sim \sum_{i=1}^n \delta_{y_i}$. On trouve l'estimateur de vraisemblance maximale parmi les racines de*

$$M(\theta) = 0_d.$$

*En fait, un estimateur sans biais de $M(\theta)$ est $H(\theta,\xi) := \nabla_\theta log(g(\xi;\theta))$. Ainsi, on peut voir l'algorithme du gradient stochastique comme une procédure d'approximation stochastique avec ce choix $M$ et $H$. Cet exemple est particulier, car dans ce cas, on connaît la fonction $M(\theta)$ qui est $\sum_{i=1}^n \nabla_\theta \ell^{(i)}(\theta)$, alors qu'en général, la fonction $M$ ne peut pas être évaluée point à point dans le contexte de Robbins-Monro.*

La récursion de l'algorithme de scoring de Fisher est la suivante :

$$\theta_t = \theta_{t-1} + I(\theta_{t-1})^{-1} \nabla_\theta \ell_n(\theta_{t-1}),$$

où le pas $I(\theta)$ est définie en 5. Il est possible de considérer la formulation stochastique de l'algorithme de scoring de Fisher :

$$\theta_t = \theta_{t-1} + I(\theta_{t-1})^{-1} \nabla_\theta \ell^{(i_t)}(\theta_{t-1}).$$

## 1.3. Maximisation de la vraisemblance dans les modèles à variable latente et l'algorithme EM

En présence d'un modèle contenant des variables $X$ et $Y$, où $Y$ est observée, mais $X$ est manquante, une fonction de vraisemblance ne peut pas toujours être évaluée ou maximisée. Dans le contexte des modèles à variables manquantes, on ne spécifie plus une loi paramétrée pour la variable aléatoire $Y$, mais une loi paramétrée pour le couple $(X,Y)$. La vraisemblance complète introduite en 9 est explicite, mais l'évaluer point à point requiert la connaissance de $X$ qui est inconnue dans ce modèle. La vraisemblance incomplète $g_\theta$ introduite en 10 nécessite l'intégration par rapport à $X$ de la vraisemblance complète, ce qui n'est pas forcément faisable. Le cadre ci-dessus correspond, par exemple, à celui des modèles de mélange gaussien, des modèles linéaires à effets mixtes (Lindstrom et Bates, 1988) ou des modèles à données tronquées ou censurées (Wolynetz, 1979). Dans le modèle de mélange qui revient souvent dans ce mémoire, la variable latente est la classe d'appartenance. C'est-à-dire qu'une donnée est issue de l'une des composantes du mélange, mais on ne sait pas laquelle. Le problème de maximisation de vraisemblance dans les modèles à variables latentes survient aussi dans les modèles de type «change points» (Yildirim *et al.*, 2013), les modèles de Markov cachés et les modèles de traitement de signal (Cappé, 2011). Différentes approches existent pour

obtenir des estimateurs de vraisemblance maximale dans de tels modèles. L'algorithme EM, l'approche la plus notoire, permet d'obtenir des estimateurs de vraisemblance maximale dans ces circonstances. On verra plus loin que d'autres approches comme celles basées sur l'algorithme du gradient peuvent accomplir la même tâche que l'EM dans certaines situations.

### 1.3.1. Notations

On appelle

$$\pi(y), \tag{8}$$

la vraie loi de la variable $Y$ par rapport à la mesure sigma finie $\nu(\mathrm{d}y)$. On appelle

$$f(x,y;\theta), \tag{9}$$

la densité associée au modèle du couple $(X,Y)$ par rapport à la mesure sigma finie $\mu(\mathrm{d}x)\nu(\mathrm{d}y)$. De même que pour $\nu$, si $X$ est discrète, $\mu$ sera la mesure de comptage et si $X$ est continue, $\mu$ sera une mesure de Lebesgue. On appelle

$$g(y;\theta), \tag{10}$$

la densité associée au modèle de la variable $Y$ par rapport à la mesure sigma finie $\nu(\mathrm{d}y)$. On suppose que la loi conditionnelle de $X$ sachant $Y = y$ est bien définie et est donnée pour tout $y \in \mathcal{Y}$ et $A \in \mathcal{B}(\mathcal{X})$ par

$$P_\theta(X \in A \mid Y = y) = \int_A p(x;y,\theta)\mu(\mathrm{d}x),$$

où

$$p(x;y,\theta) = \frac{f(x,y;\theta)}{g(y;\theta)} \tag{11}$$

est la densité par rapport à $\mu$ de $P_\theta(\cdot; Y = y)$. Soit $X \in \mathcal{X}$, $Y \in \mathcal{Y}$ et $S : (\mathcal{X},\mathcal{Y}) \to \mathbb{R}$, une fonction $f(\cdot;\theta)$-intégrable, l'espérance par rapport à la jointe (caractérisée par la fonction de densité $f$) est

$$\mathbb{E}_\theta\left[S(X,Y)\right] = \int S(x,y)f(x,y;\theta)\nu(\mathrm{d}y)\mu(\mathrm{d}x).$$

Soit $S$ une fonction $p(\cdot;\theta)$-intégrable, l'espérance a posteriori par rapport à la variable latente est

$$\mathbb{E}_\theta\left[S(X,Y) \mid Y = y\right] = \int S(x,y)p(x;y,\theta)\mu(\mathrm{d}x).$$

### 1.3.2. L'algorithme du gradient dans les modèles à variables latentes

En principe, la connaissance de la vraisemblance $g(y;\theta)$ n'est pas strictement requise. C'est-à-dire qu'on pourrait trouver l'estimateur de vraisemblance maximale dans les modèles à variables latentes à l'aide de l'algorithme du gradient si on avait connaissance de

$\nabla_\theta \log g(y;\theta)$, seulement. L'identité de Fisher donne une voie pour l'obtenir :

$$\nabla_\theta \log g(y;\theta) = \mathbb{E}_\theta \left[\nabla_\theta \log f(X,Y;\theta) \mid Y = y\right]. \tag{12}$$

Lorsqu'on est en mesure de calculer cette espérance, on peut faire appel à l'algorithme du gradient :

$$\hat{\theta}_t = \hat{\theta}_{t-1} + \gamma \left(\mathbb{E}_{\hat{\theta}_{t-1}} \left[\nabla_\theta \log f(X,Y;\hat{\theta}_{t-1}) \mid Y = y\right]\right), \tag{13}$$

où $\gamma \in \mathbb{R}$. Dans l'esprit des méthodes de Fisher ou de Newton abordées à l'équation Eq. 1.2.2, le pas $\gamma$ pourrait être substitué par une matrice $A$ de taille $p \times p$. À la section 2.1, quelques candidats pour la matrice $A$ sont présentés. Il est manifeste que le succès de cette méthode dépend de la possibilité de calculer l'espérance en 12, ce qui est difficile habituellement.

### 1.3.3. Motivation de l'algorithme EM

La fonction de vraisemblance observée $g(y;\theta)$ et son gradient ne sont pas forcément disponibles. Le modèle $g_\theta$ du cas usuel où il n'y a pas de variable cachée est la densité marginale de $Y$ où on a intégré $X$. Dans certaines situations, il n'est pas possible de résoudre l'intégrale. Par conséquent, maximiser la fonction de vraisemblance observée ne peut être effectué directement. Cependant, il est possible d'obtenir itérativement une quantité intermédiaire, un substitut, qui permettra de trouver les paramètres du maximum de vraisemblance sans avoir à maximiser directement la fonction de vraisemblance observée $g_\theta$.

Soit $(X_i,Y_i)$ iid pour $i = 1, \dots n$, on définit

$$Q(\theta;\theta') \overset{\text{déf}}{=} \frac{1}{n}\,\mathbb{E}_{\theta'} \left[\sum_{i=1}^{n} \log(f(X_i,Y_i;\theta)) \mid Y_1, \dots, Y_n\right] = \int \frac{1}{n} \sum_{i=1}^{n} \log(f(x_i,y_i;\theta)) p(x_i;y_i,\theta') \mu(\mathrm{d}x_i) \tag{14}$$

et

$$H(\theta;\theta') \overset{\text{déf}}{=} \frac{1}{n}\,\mathbb{E}_{\theta'} \left[\sum_{i=1}^{n} \log\left(p(X_i;Y_i,\theta)\right) \mid Y_1, \dots, Y_n\right] = -\int \frac{1}{n} \sum_{i=1}^{n} \log(f(x_i;y_i,\theta)) p(x_i;y_i,\theta') \mu(\mathrm{d}x_i). \tag{15}$$

**Lemme 1.3.1.** *La log-vraisemblance* $\log(g(y;\theta))$ *peut être écrite comme la somme de* $Q(\theta;\theta') + H(\theta;\theta')$.

Démonstration. Pour des raisons de présentation, on établit les résultats pour un jeu d'une donnée $n = 1$. Le raisonnement reste vrai en remplaçant chaque espérance par la somme de $n$ espérances, chacune correspondant à un jeu d'une donnée.

La densité jointe peut être exprimée comme le produit de la densité conditionnelle et la densité marginale :

$$f\left(x,y;\theta\right) = p\left(x;y,\theta\right) g\left(y;\theta\right).$$

On écrit la log-vraisemblance de la façon suivante :

$$\log\left(g\left(y;\theta\right)\right) = \log\left(f(x,y;\theta)\right) - \log\left(p(x;y,\theta)\right). \tag{16}$$

À partir de 16, on peut écrire :

$$\int \log(g(y;\theta))p(x;y,\theta')\mu(\mathrm{d}x) = \int \log(f(x,y;\theta))p(x;y,\theta')\mu(\mathrm{d}x) - \int \log(p(x;y,\theta))p(x;y,\theta')\mu(\mathrm{d}x). \tag{17}$$

Finalement, on insère 14 et 15 dans 17 de sorte qu'on obtienne

$$\log(g(y;\theta)) = Q(\theta;\theta') + H(\theta;\theta'). \tag{18}$$

□

La clé de l'algorithme EM réside dans le lemme suivant. Il garantit qu'une augmentation de la fonction $Q(\theta;\theta')$ entraîne une augmentation de la vraisemblance.

**Lemme 1.3.2.** *Il est vrai que* $\log(g(y;\theta)) - \log(g(y;\theta')) \geq Q(\theta;\theta') - Q(\theta';\theta')$.

Démonstration. L'équation 18 demeure vraie en posant $\theta = \theta'$. On soustrait cette quantité de l'équation 18 :

$$\log(g(y;\theta)) - \log(g(y;\theta')) = Q(\theta;\theta') - Q(\theta';\theta') + (H(\theta;\theta') - H(\theta';\theta')). \tag{19}$$

On réécrit de manière explicite

$$H(\theta;\theta') - H(\theta';\theta') = -\int \log(p(x;y,\theta))p(x;y,\theta')\mu(\mathrm{d}x) + \int \log(p(x;y,\theta'))p(x;y,\theta')\mu(\mathrm{d}x).$$

La convexité de -log permet d'utiliser l'inégalité de Jensen :

$$H(\theta;\theta') - H(\theta';\theta') = -\int \log\left(\frac{p(x;y,\theta)}{p(x;y,\theta')}\right) p(x;y,\theta')\mu(\mathrm{d}x) \tag{20}$$

$$\geq -\log \int \frac{p(x;y,\theta)}{p(x;y,\theta')} p(x;y,\theta')\mu(\mathrm{d}x) = 0. \tag{21}$$

L'inégalité 20 permet de déduire l'inégalité suivante à partir de l'équation 19 :

$$\log(g(y;\theta)) - \log(g(y;\theta')) \geq Q(\theta;\theta') - Q(\theta';\theta').$$

□

**Remarque 1.3.3.** *La démonstration des lemmes 1.3.1 et 1.3.2 demeure valide pour des vraisemblances contenant n observations.*

Le lemme 1.3.2 implique qu'une augmentation de la quantité $Q(\theta;\theta')$ engendre une augmentation équivalente ou supérieure de la log-vraisemblance $\log g(y;\theta)$. De plus, une augmentation aura lieu à moins que la valeur maximisant $Q(\theta;\theta')$ demeure $\theta'$. Des résultats

concernant la convergence sont montrés par Wu (1983). De ceux-ci, notons que lorsque la vraisemblance complète appartient à une famille exponentielle avec un support $\Theta$ compact, la séquence $\{\theta_t\}$ converge vers un point où le gradient de la vraisemblance observée est 0. En présence d'une fonction de vraisemblance unimodale, la suite $\{\theta_t\}$ converge vers un estimateur de vraisemblance maximale unique.

En résumé, l'algorithme EM alterne entre une étape d'estimation et de maximisation. Dans un premier temps, on fournit une valeur initiale $\hat{\theta}_0$ quelconque du paramètre d'intérêt. Grâce à cette valeur, on est mesure de calculer l'espérance et ainsi générer la nouvelle fonction à maximiser. On obtient $\hat{\theta}_1$, la valeur qui maximisent la fonction obtenue à l'étape précédente. On peut ensuite répéter le processus. On s'arrête à l'itération $N$ lorsque toutes les données ont été vues.

**Algorithme 1.3.4.** L'algorithme EM (Dempster *et al.*, 1977).

---

**Entrées :** $N \in \mathbb{N}$, $\hat{\theta}_0 \in \Theta$, un jeu de données $y_1, \ldots, y_n$ ;
**Initialisation** ;
    $k := 1$ ;
**Tant que** $k < N + 1$ :
    **Étape E :** *Calculer la fonction* $Q(\theta; \hat{\theta}_{k-1}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\hat{\theta}_{k-1}}[\log f(X_i, Y_i; \theta) \mid Y_i = y_i]$ ;
    **Étape M :** $\hat{\theta}_k = \underset{\theta}{\operatorname{argmax}}\ Q(\theta; \hat{\theta}_{k-1})$ ;
    $k \leftarrow k + 1$ ;
**fin** ;
**retourner** $\hat{\theta}_N$ ;

---

Notons que le critère de fin pourrait reposer sur une tolérance concernant la différence $Q(\hat{\theta}_k; \hat{\theta}_{k-1}) - Q(\hat{\theta}_{k-1}; \hat{\theta}_{k-1})$ plutôt qu'un nombre fixe d'itérations. Dans ce mémoire, on emploie l'algorithme EM pour un nombre prédéfini d'itérations.

**Remarque 1.3.5.** *L'algorithme du gradient obtenu quand* $\nabla_\theta \log g(y; \theta)$ *est connu est équivalent à l'algorithme EM, en régime asymptotique (Cappé et Moulines, 2009).*

## 1.4. Méthode de Monte-Carlo

La méthode de Monte-Carlo permet d'approximer des intégrales. Dans notre contexte, l'intégrale d'intérêt est l'espérance. Soit une fonction $h : \mathcal{X} \to \mathbb{R}$ et $p$ la densité associée à la variable aléatoire $X$, définie comme en 11.

$$\mathbb{E}[h(X)] = \mu = \int h(x)p(x)\mu(\mathrm{d}x)$$

Lorsque $X_1, \ldots, X_m$ sont iid, un estimateur sans biais de $\mathbb{E}[h(X)]$ est la moyenne arithmétique des observations $h(X_1), h(X_2), \ldots, h(X_m)$,

$$\hat{\mu}_{MC}^{m} \overset{\text{déf}}{=} \frac{1}{m}\sum_{i=1}^{m} h(X_i).$$

De plus, il s'agit d'un estimateur convergent en probabilité (et même presque sûrement) ; une conséquence de la loi faible des grands nombres (et de la loi forte des grands nombres) (Robert et Casella, 2013).

Soit l'estimateur Monte-Carlo de $\text{Var}[h(X)]$,

$$\hat{\sigma}_{m,MC}^{2} \overset{\text{déf}}{=} \sum_{i=1}^{m} \frac{\left(h(X_i) - \hat{\mu}_{MC}^{m}\right)^2}{m}.$$

Il est possible de faire de l'inférence sur $\hat{\mu}_{MC}^{m}$ puisque

$$\frac{\sqrt{m}(\hat{\mu}_{MC}^{m} - \mu)}{\sqrt{\hat{\sigma}_{m,MC}^{2}}} \xrightarrow{D} \mathcal{N}(0,1).$$

# 1.5. Méthodes MCMC

Lorsqu'il n'est pas possible d'échantillonner directement une loi à l'aide de la méthode Monte-Carlo, les méthodes MCMC contournent le problème lorsqu'on a connaissance de la densité de la loi à une constante de normalisation près. De plus, les méthodes MCMC sont efficaces en grande dimension.

## 1.5.1. Principe

L'approche consiste en la simulation d'une chaîne de Markov dont la distribution stationnaire coïncide avec la loi d'intérêt $p$ de laquelle on souhaiterait tirer un échantillon afin d'approximer $\mathbb{E}[h(X)]$. Un algorithme très populaire qui accomplit cette tâche est l'algorithme de Metropolis-Hastings qui est décrit à la sous-section 1.5.2. Avant de passer à l'algorithme, présentons quelques faits généraux. Soit $\{X_n; n \in \mathbb{N}^+\}$, une chaîne de Markov de noyau $K$ avec distribution stationnaire $p$.

L'estimateur de $\mathbb{E}[h(X)]$ est

$$\hat{\mu}_{MCMC}^{m} \overset{\text{déf}}{=} \frac{1}{m}\sum_{i=1}^{m} h(X_i), \tag{22}$$

où $\{X_n\}$ est une chaîne de Markov de noyau $K$ et de loi initiale $p_1$.

Il est manifeste que l'estimateur comporte un biais à moins que $p_1$ coïncide avec $p$. En général, la chaîne de Markov $\{X_k\}$ atteint sa loi stationnaire asymptotiquement. La moyenne arithmétique des $h(X_k)$ avec $k = 1, \ldots, m$ convergera alors vers $\mu = \mathbb{E}[h(X)]$ (Roberts et Rosenthal, 2004). On recommande aussi d'éliminer les $B$ premiers membres de la chaîne pour améliorer l'estimateur, ce qu'on appelle la phase de « burn in » (Rosenthal, 2000). Il existe des résultats concernant les conditions qui garantissent l'existence d'un théorème central

limite dans la partie 5.2 de Roberts et Rosenthal (2004) et de la loi des grands nombres dans la partie 17.0.1 de Meyn et Tweedie (1993). La loi faible des grands nombres pour chaînes de Markov garantit la convergence en probabilité d'un estimateur $\hat{\mu}^m_{MCMC}$ de $\mu = \mathbb{E}[h(X_k)]$. L'ergodicité de la chaîne donne lieu au théorème central limite pour chaînes de Markov. Cela permet de déduire une approximation sur la distribution asymptotique de l'estimateur $\hat{\mu}^m_{MCMC}$ quand $m$ est grand,

$$\sqrt{m}\left(\hat{\mu}^m_{MCMC} - \mu\right) \xrightarrow{D} \mathcal{N}\left(0, \mathrm{Var}\left(h\left(X_0\right)\right) + 2\sum_{k=0}^{\infty} \mathrm{Cov}\left[h\left(X_0\right), h\left(X_{k+1}\right)\right]\right).$$

Lorsque le terme $2\sum_{k=0}^{\infty} \mathrm{Cov}[h(X_0), h(X_{k+1})]$ est positif, la variance asymptotique de l'estimateur $\hat{\mu}^m_{MCMC}$ est supérieure à la variance de l'estimateur $\hat{\mu}^m_{MC}$. Ceci se produit quand les éléments de la chaîne de Markov sont grandement corrélés (Rosenthal, 2000).

### 1.5.2. L'algorithme de Metropolis-Hastings

La méthode MCMC employée dans ce mémoire est l'algorithme de Metropolis-Hastings qui a été élaboré par Metropolis *et al.* (1953) et Hastings (1970). L'algorithme permet de générer une chaîne de Markov qui converge vers la loi de probabilité $p$ que l'on peut évaluer point à point à une constante de normalisation près. Il est possible grâce à l'algorithme de Metropolis-Hastings de construire une chaîne de Markov qui a pour loi invariante $p$. L'algorithme est conçu de sorte que sous des conditions générales, le théorème limite central évoqué en 5.2 de Roberts et Rosenthal (2004) et la loi des grands nombres en 17.0.1 de Meyn et Tweedie (1993) s'appliquent.

Soit $f$, une fonction de $x$ proportionnelle à $p$ (typiquement $f$ est la densité non normalisée), et $K$, le noyau de proposition dont la loi est $\mathcal{N}\left(x,\sigma^2\right)$, la loi normale de moyenne $x$ et de variance $\sigma^2$ caractérisée par la fonction de densité

$$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right\}.$$

Il suffit de mettre en oeuvre l'algorithme de Metropolis-Hastings avec marche aléatoire à incrément gaussien. C'est donc la forme standard, l'algorithme de Metropolis. À partir d'une valeur initiale $x_0$, on propose une nouvelle valeur $z_1$ générée à l'aide de la loi normale centrée en $x_0$. On évalue le rapport $\alpha_1$ des densités proportionnelles à une constante de normalisation près. Si le rapport excède 1, on accepte la nouvelle valeur, c'est-à-dire que $x_1 = z_1$. Sinon, on tire un nombre $U_1$ suivant une loi uniforme entre 0 et 1. Si la réalisation $u_1$ excède le rapport $\alpha_1$, $x_1$ prend la valeur de $x_0$. Le processus continue jusqu'à l'itération $m$.

Soit $x_0$. Pour $i = 1, 2, 3, \ldots$ :

1. Générer $Z_i \sim \mathcal{N}\left(x_{i-1},\sigma^2\right)$.

2. Générer $U_i \sim$ Uniforme(0,1) et calculer

$$\alpha_i \stackrel{\text{déf}}{=} \min\left\{\frac{f(z_i)}{f(x_{i-1})}, 1\right\}. \tag{23}$$

3. Poser

$$X_i = \begin{cases} Z_i & \text{si} \quad U_i \leq \alpha_i, \\ X_{i-1} & \text{si} \quad U_i > \alpha_i. \end{cases} \tag{24}$$

**Algorithme 1.5.1.** L'algorithme de Metropolis à incrément gaussien.

---

**Entrées :** $X_0 \in \mathcal{X}, m \in \mathbb{N}, \sigma^2 \in \mathbb{R}^+$ ;
**Initialisation ;**
    $i := 1$ ;
**Tant que** $i < m + 1$ **:**
    *Générer* $Z_i \sim \mathcal{N}(x_{i-1}, \sigma^2)$ ;
    *Générer* $U_i \sim$ Uniforme(0,1) *et calculer* $\alpha_i \stackrel{déf}{=} \min\left\{\frac{f(z_i)}{f(x_{i-1})}, 1\right\}$ ;
    **Si** $U_i \leq \alpha_i$ **;**
        $X_i = Z_i$ ;
    **Sinon ;**
        $X_i = X_{i-1}$ ;
    $i \leftarrow i + 1$ **;**
**fin ;**

---

La loi de la chaîne de Markov $\{X_i; i \in \mathbb{Z}_+\}$ converge en variation totale vers $p$ quand $i \to \infty$. L'algorithme de Metropolis-Hastings permet donc de générer une chaîne de Markov dont la loi asymptotique est $p$ (Roberts et Rosenthal, 2004).

# 1.6. Réduction de variance

## 1.6.1. Variable de contrôle

Parfois, la méthode de Monte-Carlo engendre des approximations ayant une forte variance, et ce, même dans le cas univarié. Cela se produit lorsque la fonction d'intérêt $h$ est complexe ou quand le nombre de réalisations Monte-Carlo est insuffisant, par exemple. Bien sûr, plus la taille de l'échantillon $m$ est grande, plus l'estimateur sans biais sera efficace puisqu'il convergera asymptotiquement (à un taux $\sqrt{m}$) vers la vraie valeur. Une stratégie générale pour réduire la variance consiste à remplacer $h$ par une fonction $\tilde{h}$ dont l'espérance par rapport à $f$ est égale à celle de la fonction $h$, où $f$ est la loi à une constante de normalisation près de $X$. La fonction $\tilde{h}$ est construite de sorte qu'il y ait réduction de variance par rapport à $f$. Soit $\mathcal{X} \subseteq \mathbb{R}^d$. Soit une fonction $\phi : \mathcal{X} \to \mathbb{R}$, on définit une autre fonction

$\tilde{h} : \mathcal{X} \to \mathbb{R}$ comme suit,

$$\tilde{h}(X) = h(X) + c(\phi(X) - \tau)$$

où $c$ est une constante, $\phi(X)$ est une variable de contrôle et $\mathbb{E}_f[\phi(X)] \stackrel{\text{déf}}{=} \tau$. La variance de $\tilde{h}(X)$ est

$$\text{Var}\left[\tilde{h}(X)\right] = \text{Var}\left[h(X) + c(\phi(X) - \tau)\right] = \text{Var}\left[h(X)\right] + c^2\text{Var}\left[\phi(X)\right] + 2c\text{Cov}[h(X), \phi(X)] \tag{25}$$

Afin de minimiser la variance, il faut choisir la valeur de $c$ qui minimise la variance de $\tilde{h}$. En dérivant l'équation par rapport à $c$, il suit que la valeur optimale de $c$ est

$$c^* = -\frac{\text{Cov}\left[h(X), \phi(X)\right]}{\text{Var}\left[\phi(X)\right]}. \tag{26}$$

En insérant l'équation 26 dans l'équation 25, on voit l'effet potentiel de la variable de contrôle fortement corrélée sur la variance.

$$\begin{aligned}
\text{Var}\left[\tilde{h}(X)\right] &= \text{Var}\left[h(X)\right] + \frac{\left(\text{Cov}\left[h(X), \phi(X)\right]\right)^2}{\text{Var}\left[\phi(X)\right]} - 2\frac{\left(\text{Cov}\left[h(X), \phi(X)\right]\right)^2}{\text{Var}\left[\phi(X)\right]} \\
&= \text{Var}\left[h(X)\right] - \frac{\left(\text{Cov}\left[h(X), \phi(X)\right]\right)^2}{\text{Var}\left[\phi(X)\right]} \\
&= \left(1 - \rho^2_{h,\phi}\right)\text{Var}\left[h(X)\right]
\end{aligned}$$

où $\rho^2_{h,\phi} = \frac{\left(\text{Cov}[h(X),\phi(X)]\right)^2}{\text{Var}[\phi(X)]\text{Var}[h(X)]}$ est le coefficient de corrélation au carré entre $h$ et $\phi$.

**Exemple 1.6.1.** *Prenons l'exemple de l'intégration* $\int_0^1 \frac{1}{1+u}\text{d}u$. *Une variable de contrôle pour* $h(u) = \frac{1}{1+u}$ *est* $1 + U$, *où* $U \sim$ Uniforme(0,1).

*Supposons qu'on ne sache pas trouver*

$$\text{Cov}\left[\frac{1}{1+U}, 1+U\right] = \mathbb{E}\left[\frac{1}{1+U}(1+U)\right] - \mathbb{E}\left[\frac{1}{1+U}\right]\mathbb{E}[1+U] = 1 - \frac{3}{2}\log(2)$$

*et*

$$Var\left[\frac{1}{1+U}\right] = 0{,}5 - \log(2)^2 = 0{,}01954699,$$

*il faudrait proposer une version approximative de c que l'on pourrait obtenir numériquement.*

*La valeur optimale de c dans cet exemple est* $c^* = 0{,}4766$. *Les estimateurs de* $\int_0^1 \frac{1}{1+u}\text{d}u$ *que l'on cherche à comparer sont*

$$\begin{aligned}
\hat{I}_{MC} &= \frac{1}{m}\sum_{i=1}^{m}\frac{1}{1+U_i}, \\
\hat{I}_{Contrôle} &= \frac{1}{m}\sum_{i=1}^{m}\left\{\frac{1}{1+U_i} + c^*(1+U_i)\right\} - 3/2c^*.
\end{aligned}$$

*Les variances estimées sont respectivement* $\widehat{Var}(I_{MC}) = 2{,}029341 \times 10^{-07}$ *et* $\widehat{Var}(I_{Contrôle}) = 6{,}082569 \times 10^{-09}$.

*Le facteur* $1 - \rho_{h,\phi} = 0{,}0314$ *est du même ordre de grandeur que le rapport des variances obtenu :*

$$\frac{\widehat{Var}[I_{Contrôle}]}{\widehat{Var}[I_{MC}]} = 0{,}02997.$$

### 1.6.2. Réduction de variance dans les méthodes MCMC

Il n'est pas toujours évident d'identifier une variable de contrôle. Si une variable de contrôle existe, la variance du nouvel estimateur peut grandement diminuer dans le cas d'une intégrale ayant une forme assez simple, comme dans l'exemple 1.6.1. Une covariable fortement corrélée avec la variable aléatoire permet de réduire la variance d'un estimateur Monte-Carlo à condition d'avoir une très bonne idée de son espérance, de sa variance et de sa covariance avec la variable dont on cherche à réduire la variance, ce qui est rarement le cas en pratique. Lorsqu'on dispose d'un échantillon MCMC, il n'est pas aisé de trouver une variable de contrôle. Il existe une technique pour créer une telle variable après avoir obtenu une chaîne de Markov d'un algorithme MCMC. Pour ce faire, on fait un usage judicieux de notre connaissance de $f$, la loi à une constante de normalisation près de la distribution cible $p$.

Soit $H$, un opérateur quelconque auquel on reviendra plus tard. L'estimateur de variance réduite proposé dans Mira *et al.* (2003) est défini comme suit :

$$\tilde{h}(x) = h(x) + \frac{H\phi(x)}{\sqrt{f(x)}}.$$

On peut montrer la relation suivante :

$$\begin{aligned} f(x)\tilde{h}(x) &= f(x)h(x) + \sqrt{f(x)}H\phi(x), \\ \mathbb{E}\left[\tilde{h}(X)\right] &= \mathbb{E}\left[h(X)\right] + \int \sqrt{f(x)}H\phi(x)\mu(\mathrm{d}x). \end{aligned}$$

Il faut choisir $H$ et $\phi$ tel que $\int \sqrt{f(x)}H\phi(x)\mu(\mathrm{d}x)=0$ afin d'obtenir un estimateur sans biais. Soit $x \in \mathbb{R}^d$, considérons l'opérateur proposé par Mira *et al.* (2013),

$$H = -\frac{1}{2}\frac{\partial^2}{\partial x^2} + \frac{1}{2\sqrt{f(x)}}\frac{\partial^2\sqrt{f(x)}}{\partial x^2}.$$

Il existe plusieurs choix de fonction $\phi$. Un choix populaire consiste en la fonction :

$$\phi(x) = P(x)\sqrt{f(x)},$$

où $P(x) = \sum_{i=1}^{d} a_i x_i$, un polynôme de degré 1 ou $P(x) = \mathbf{a}^T x + \frac{1}{2} x^T B x$, un polynôme quadratique. C'est le polynôme quadratique qui sera employé dans ce mémoire. En pratique, il existe un package dans le progiciel R « ZVCV » permettant d'obtenir des estimateurs MCMC à variance réduite à la manière de Mira *et al.* (2013).

L'estimateur de variance zéro est la suivante :

$$\tilde{h}(x) = h(\mathbf{x}) - \frac{1}{2}\Delta P(x) + \nabla P(x) \cdot \mathbf{z},$$

où l'opérateur $\nabla \overset{\text{déf}}{=} (\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_n})$, l'opérateur $\Delta \overset{\text{déf}}{=} \sum_{i=1}^{n} \frac{\partial^2}{\partial x_i^2}$ et $\mathbf{z} \overset{\text{déf}}{=} -\frac{1}{2}\nabla \log f(x)$. Un théorème central limite s'applique à cet estimateur et est discuté à la section 5.1 de Mira *et al.* (2013).

**Proposition 1.6.2.** *(Mira* et al.*, 2013) Soit $f$ une densité de dimension $d$ dont le domaine est un ensemble ouvert borné $\Omega$ avec la frontière $\partial\Omega$, dont les dérivées d'ordre un et deux sont continues. Alors, si $\phi = P\sqrt{f}$, une condition suffisante pour que l'estimateur $\tilde{h}$ soit sans biais est $f(x)\frac{\partial P(x)}{\partial x_j} = 0$, pour tout $x \in \partial\Omega$, $j = 1, \ldots, d$.*

La démonstration se trouve à l'annexe A.2.

Le résultat suivant est montré à l'annexe 2 de Mira *et al.* (2013).

**Proposition 1.6.3.** *(Mira* et al.*, 2013) Soit $f$ la densité de dimension $d$ avec un support $\mathcal{X}$ non-borné, dont la première et la deuxième dérivée sont continues. Soit $\{B_r\}_r$ une suite de sous-ensembles bornés tels que $B_r \nearrow \mathcal{X}$. Une condition suffisante pour que l'estimateur variance zéro MCMC soit sans biais est* $\lim_{r\to\infty} \frac{1}{2} \int_{\partial B_r} f(x)P(x) = 0$.

### 1.6.3. Réduction de variance dans l'algorithme du gradient

Il existe des algorithmes du gradient à variance réduite de l'algorithme du gradient stochastique. L'un de ceux-ci, l'algorithme de SVRG (Johnson et Zhang, 2013) est le suivant. L'algorithme SVRG nécessite un $\theta_0$ initial. Dans la première boucle itérée par $s$, on pose $\tilde{\theta} = \theta_0$, car c'est la dernière valeur disponible, et on calcule le gradient en $\tilde{\theta}$. On pose $\theta_0 = \tilde{\theta}$, ce qui sera pertinent pour les itérations suivantes. Dans la deuxième boucle itérée par $j$, on choisit un $i_1$ aléatoirement entre 1 et $n$, ce qui permet de choisir un $y_{i_j}$. À l'étape d'approximation stochastique, on insère le gradient évaluée en $\theta_0$ car c'est la dernière valeur disponible, le gradient évaluée $\tilde{\theta}$ et $\tilde{\mu}$, qui permettent de constituer une variable de contrôle. On recmmence le processus jusqu'à $j = m$. On pose $\theta_1 = \theta_m$. On retourne dans la première

boucle, on pose $\tilde{\theta} = \theta_1$ et on répète les étapes décrites plus haut. On continue le processus jusqu'à $s = N$.

**Algorithme 1.6.4.** L'algorithme du gradient stochastique à variance réduite (SVRG) (Johnson et Zhang, 2013).

---

**Entrées** $\gamma \in \mathbb{R}$, $\theta_0 \in \Theta$ **et** $m, n, N \in \mathbb{N}$ ;
**Initialisation** ;
**Itération : Pour** $s = 1,2,\ldots,N$ **;**
  $\tilde{\theta} = \theta_{s-1}$ **;**
  $\tilde{\mu} = \frac{1}{n}\sum_{i=1}^{n} \nabla \ell_{i,n}(\tilde{\theta})$ **;**
  $\theta_0 = \tilde{\theta}$ ;
  **Itération : Pour** $t$ **:** $1, 2, \ldots, m$
    *Choisir aléatoirement* $i_t \in \{1, \ldots, n\}$ ;
    $\theta_t = \theta_{t-1} - \gamma \left( \nabla \ell_{i_t,n}(\theta_{t-1}) - \nabla \ell_{i_t,n}(\tilde{\theta}) + \tilde{\mu} \right)$ ;
  **fin ;**
  Poser $\theta_s = \theta_m$ ;
**fin ;**

---

L'algorithme 1.6.4 converge géométriquement en espérance sous certaines conditions, voir le théorème 1 de Johnson et Zhang (2013). Bien qu'il s'agisse d'un algorithme à variance réduite, l'algorithme SVRG nécessite beaucoup de calculs. À chaque itération, on doit calculer le gradient entier $\tilde{\mu}$, puis refaire toute une séquence d'algorithme du gradient stochastique. La quantité de calculs entraîne une perte d'efficacité computationnelle par rapport à l'algorithme du gradient stochastique.

# Chapitre 2

# Variantes stochastiques de l'algorithme EM

L'algorithme EM proposé par Dempster *et al.* (1977) permet d'obtenir itérativement une séquence de variables aléatoires qui converge vers l'ensemble des points stationnaires de la vraisemblance, ensemble qui inclut l'estimateur du maximum de vraisemblance. Cependant, cette approche comporte quelques difficultés potentielles. En premier lieu, l'étape E d'espérance pourrait être impossible à accomplir quand l'intégrale est difficile à résoudre. En second lieu, l'étape M de maximisation pourrait ne pas être faisable. Cette infaisabilité se produit quand le système d'équations découlant de l'équation du gradient est impossible à résoudre analytiquement. En troisième lieu, puisqu'il faut calculer $n$ espérances par itération, la procédure devient inefficace quand le volume de données est très grand, ce qui se produit couramment à l'ère des données massives.

L'algorithme EM par approximation stochastique de Delyon *et al.* (1999) est une approche permettant de résoudre le premier problème à l'aide d'une simulation de Monte-Carlo. Il s'agit d'échantillonner la loi a posteriori de la variable latente afin d'obtenir un estimateur de $Q(\cdot;\theta)$. L'algorithme de Lange (1995) trouve une solution quand la maximisation à l'étape M n'est pas possible. L'astuce de l'algorithme est de réécrire la procédure de maximisation comme une approximation d'une étape de Newton. Finalement, lorsque le volume des données est très grand, il est possible de spécifier une approche en ligne par approximation stochastique. L'algorithme EM en ligne de Cappé et Moulines (2009) adopte cette stratégie. Toutes les variantes de l'algorithme EM mentionnées dans ce mémoire sont des algorithmes incrémentaux. On dira d'un algorithme incrémental qu'il est en ligne lorsqu'il traite une seule donnée par itération et que chacune des données est traitée une fois au total et intervient qu'une seule fois au total dans l'inférence. Ceci peut être utile lorsque, pour des raisons de capacité de mémoire, les données ne peuvent être conservées et/ou lorsque celles-ci deviennent disponibles au compte-gouttes, de façon peu prévisible. À l'instar de l'algorithme de Cappé et Moulines (2009), l'algorithme de Titterington (1984) est aussi un algorithme en ligne. L'approximation stochastique est employée dans l'étape d'espérance de tous les algorithmes

sauf Lange (1995) et Neal et Hinton (1998). Les auteurs de ces différentes méthodes ont tous proposé des garanties théoriques pour ces variantes (Delyon *et al.*, 1999; Lange, 1995; Cappé et Moulines, 2009; Titterington, 1984).

Avant de présenter succinctement ces méthodes, rappelons que la fonction de vraisemblance des données complètes est $f(x,y;\theta)$ et la fonction de vraisemblance des données incomplètes est $g(y;\theta)$.

## 2.1. L'algorithme EM incrémental du gradient

Une variante incrémentale, mais pas en ligne, de l'algorithme EM est proposée par Lange (1995). C'est une forme de l'algorithme du gradient dont les solutions sont les estimateurs de vraisemblance maximale. Cette méthode est utile quand la maximisation explicite n'est pas possible.

La récurrence est

$$\hat{\theta}_t = \hat{\theta}_{t-1} + \gamma_t J\left(Y_{1:n};\hat{\theta}_{t-1}\right)^{-1} \sum_{i=1}^{n} \mathbb{E}_{\hat{\theta}_{t-1}}\left[\nabla_\theta \log f(X_i,Y_i;\hat{\theta}_{t-1}) \mid Y_i\right] \tag{27}$$

avec $J(Y_{1:n};\hat{\theta}_{t-1}) = -\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\hat{\theta}_{t-1}}\left[\nabla^2_\theta \log f(X_i,Y_i;\hat{\theta}_{t-1}) \mid Y_i\right]$ et $\{\gamma_t\}$ un pas d'apprentissage. Lorsque $\gamma_t = 1$ pour tout $t$, ceci correspond à une étape de la méthode de Newton pour maximiser $Q(\theta;\hat{\theta}_t) = \sum_{i=1}^{n} \frac{1}{n} \mathbb{E}_{\hat{\theta}_t}\left[\log f(X_i,Y_i;\theta) \mid Y_i\right]$, la fonction intermédiaire de l'EM. Sous certaines conditions, la récursion 27 possède la propriété de monotonie qui permet un accroissement de la vraisemblance à chaque itération quand on se rapproche de la convergence (Lange, 1995). L'inversion de la matrice $J$ à chaque itération est une difficulté, car celle-ci exige beaucoup de mémoire computationnelle. De plus, cette matrice pourrait être singulière.

Une modification possible consiste à remplacer $J(Y_{1:n};\theta)$ par $-\mathbb{E}_\theta[\nabla^2_\theta \log f(X,Y;\theta)]$, la matrice d'information de Fisher basée sur les données complètes. Rappelons que cette espérance est calculée par rapport à la densité de $f(x,y;\theta)$. Il s'agit d'une approximation de l'algorithme de scoring de Fisher de l'équation 1.2.2. Une autre modification serait la matrice $-\mathbb{E}_\pi[\mathbb{E}_\theta[\nabla^2_\theta \log f(X,Y;\theta) \mid Y]]$, l'espérance par rapport à loi dont la densité est $p(x;y,\theta)\pi(y)$. Ces dernières sont des choix de la matrice $A$ de l'équation 13. Lorsque $\pi = g_\theta$, les matrices sont équivalentes.

## 2.2. L'algorithme EM par approximation stochastique de Delyon

L'algorithme EM par approximation stochastique repose sur une procédure de Robbins-Monro et une simulation de la méthode de Monte-Carlo. L'algorithme contient deux boucle, la première concerne l'itération en $k$ au niveau de la mise-à-jour de $\hat{\theta}$. À l'intérieure de celle-ci, il u a une autre boucle itérée par $j$. À partir d'une valeur initiale quelconque $\hat{\theta}_0$, on génère

$m$ réalisations de $X_j$ suivant la loi conditionnelle a posteriori $P(x; y_j, \hat{\theta}_0)$. On fait ceci pour $j = 1, \ldots, n$. On a donc généré des réalisations $X_{j,l}$ pour $j = 1, \ldots, n$ et $l = 1, \ldots, m$. À la sortie de la boucle, l'étape d'approximation stochastique utilise un estimateur Monte-Carlo de la fonction $Q$, que l'on peut obtenir à l'aide des $X_{j,l}$. On maximise la fonction $\hat{Q}_1$ afin de mettre à jour l'estimateur de $\hat{\theta}_1$. C'est celui-ci qu'on utilise afin d'obtenir de nouvelles réalisations de $X_j$ à l'itération suivante avec $P(x; y_j, \hat{\theta}_1)$. On répète le processus jusqu'à l'itération $N$. Au fil des itérations, $\hat{Q}_k(\hat{\theta}_k)$ diffère de moins en moins de $\hat{Q}_k(\hat{\theta}_{k-1})$.

### 2.2.1. Algorithme

**Algorithme 2.2.1.** L'algorithme EM par approximation stochastique Monte-Carlo (Delyon *et al.*, 1999).

---

**Entrées** Une suite décroissante de nombres positifs $\{\gamma_t\}$, $\hat{\theta}_0 \in \Theta$ et $N, n, m \in \mathbb{N}$ ;
**Initialisation ;**
    $k := 1$ ;
**Tant que** $k < N + 1$ **:**
    $j := 1$ ;
    **Tant que** $j < n + 1$ **:**
        *Générer* $X_{j,1}, \ldots, X_{j,m}$ à *partir de la loi de* $P_{\hat{\theta}_{k-1}}(x_j; y_j, \hat{\theta}_{k-1})$ ;
        $j \leftarrow j + 1$ ;
    **fin ;**
    *Calculer une approximation de* $Q(\theta; \hat{\theta}_k)$,
    $\hat{Q}_k(\theta) = \hat{Q}_{k-1}(\theta) - \gamma_k \left( \hat{Q}_{k-1}(\theta) - \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{1}{nm} \log f(X_{i,j}, Y_i; \theta) \right)$ ;
    $\hat{\theta}_k = \underset{\theta \in \Theta}{\operatorname{argmax}}\ \hat{Q}_k(\theta)$ ;
    $k \leftarrow k + 1$ ;
**fin ;**
**retourner** $\hat{\theta}_N$ ;

---

Cet algorithme ne demande pas le calcul de l'espérance, remplaçant la fonction $Q(\cdot, \hat{\theta}_{k-1})$ par un estimateur $\hat{Q}_k(\cdot)$ intégré dans une étape d'approximation stochastique. L'algorithme peut paraître difficilement implémentable à cause de la dépendance en $\theta$ de la récursion. Mais cette difficulté n'est pas propre à l'algorithme 2.2.1. La remarque vaut plus généralement pour l'algorithme EM et ses variantes. Lorsqu'on est en présence d'un modèle précis, par exemple d'une vraisemblance complète qui est de la famille exponentielle, la dépendance en $\theta$ disparaît de cette récurrence et l'implémentation est simplifiée, ce qui est justifié à la sous-section 2.4.2. De plus, il existe dans ce cas des résultats de convergence qui se basent sur les résultats d'approximation stochastique (Delyon *et al.*, 1999).

### 2.2.2. Caractérisation de la convergence de l'algorithme EM

On peut caractériser la convergence de l'EM de deux manières équivalentes. La première repose sur la stationnarité de $\{\theta_t\}$. Soit,

$$M_{EM}(\theta) \stackrel{\text{déf}}{=} \underset{u\in\Theta}{\text{argmax}} \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\theta}\left[\log f(X_i,Y_i;u) \mid Y_i\right] - \theta.$$

En effet, l'EM cherche à résoudre l'équation suivante :

$$M_{EM}(\theta) = 0.$$

Cette formulation de l'algorithme EM ne se prête pas à la procédure Robbins-Monro, car ce n'est pas une espérance. La seconde manière est d'identifier les fonctions de l'ensemble suivant :

$$\left\{ Q^* : \Theta \to \mathbb{R} \mid Q(\theta) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\underset{u\in\Theta}{\text{argmax}} Q(u)}\left[\log f(X_i,Y_i;\theta) \mid Y_i\right] \right\},$$

ou encore

$$\left\{ Q^* : \Theta \to \mathbb{R} \mid Q(\theta) = \frac{1}{n}\sum_{i=1}^{n} \int p\left(x_i; y_i, \underset{u\in\Theta}{\text{argmax}}\ Q(u)\right) \log f(x_i,y_i;\theta)\mu(\mathrm{d}x_i) \right\}.$$

On pourrait alors parler d'algorithme ME (maximisation et espérance), car une itération de l'algorithme consisterait, en un permier temps, à maximiser la fonction $Q$ et, en un second temps, à calculer la nouvelle espérance basée sur le paramètre $\theta$ qui avait maximisé la fonction $Q$ précédente. En clair, on pourrait fournir une fonction $Q_0$ initiale à maximiser, ce qui permet d'obtenir $Q_1$ et ainsi de suite jusqu'à $Q^\star$. Soit,

$$M_{ME}(Q) \stackrel{\text{déf}}{=} \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\underset{u\in\Theta}{\text{argmax}} Q(u)}\left[\log f(X_i,Y_i;\theta) \mid Y_i\right] - Q.$$

Afin de retrouver une approximation stochastique, on cherche à trouver $Q$ tel que

$$M_{ME}(Q) = 0.$$

Notons que c'est une équation fonctionnelle. Bien que cette caractérisation semble moins naturelle et, en fait, plus difficile à résoudre en pratique, elle se prête bien à l'approximation stochastique de Robbins-Monro, car elle fait apparaître une espérance :

$$M_{ME}(Q) = \mathbb{E}_{\text{argmax}Q}\left[\frac{1}{n}\sum_{i=1}^{n} \log f(X_i,Y_i;\cdot) - Q \mid Y_1,\ldots,Y_n\right].$$

Comparativement à la formulation traditionnelle de l'EM, les étapes E et M ont été interverties. Ainsi, lorsqu'on dispose d'une fonction approximative telle que

$$H_{ME}(Q,\xi) = \alpha(\theta,Y_1,\ldots,Y_n;\xi) - Q$$

où $\mathbb{E}_{\xi}\left[\alpha(\theta,Y_1,\ldots,Y_n;\xi)\right] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{\underset{u\in\Theta}{\operatorname{argmax}Q(u)}}\left[\log f(X_i,Y_i;\theta)|Y_1,\ldots,Y_n\right]$, il est possible de faire de l'approximation stochastique sur l'expression $H_{ME}(Q,\xi)$. Pour l'algorithme 2.2.1, on prend $\alpha(\theta,Y_1,\ldots,Y_n;\xi) = \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{1}{nm}\log f(X_{i,j},Y_i;\theta)$ et $\xi$ qui est $n$ séquences iid de longueur $m$ (qui peut être pris égale à 1), chaque séquence ayant pour loi $p(\cdot;Y_i,\theta)$.

L'inconvénient de cette approche est que la convergence de l'approximation stochastique dans le cadre des fonctions est peu connue. La clé est de revenir dans le modèle des familles exponentielles pour lesquelles la fonction $Q$ est naturellement remplacée par le vecteur de statistiques exhaustives $s$ pour que la dépendance en $\theta$ disparaisse, ce qui sera justifié à la sous-section 2.4.2. En somme, avec l'approche ME et un modèle de la famille exponentielle, il est possible de mettre en œuvre une approximation stochastique de manière relativement immédiate. Finalement, notons qu'une version MCMC de l'algorithme de 2.2.1 est présentée dans Kuhn et Lavielle (2004) où l'on retrouve également des résultats sur la convergence.

## 2.3. L'algorithme EM en ligne de Titterington

Le premier algorithme en ligne permettant d'obtenir des estimateurs de vraisemblance maximale dans les modèles à variables latentes a été proposé par Titterington (1984). La récursion est la suivante :

$$\hat{\theta}_t = \hat{\theta}_{t-1} + \gamma_t I_c^{-1}\left(\hat{\theta}_{t-1}\right)\nabla_\theta \log g\left(Y_t;\hat{\theta}_{t-1}\right),$$

où

$$I_c\left(\theta\right) = -\,\mathbb{E}_\theta\left[\nabla_\theta^2 \log f\left(X_i,Y_i;\theta\right)\right]. \tag{28}$$

L'algorithme s'apparente à l'algorithme du gradient stochastique, mais le gradient est conditionné par la matrice d'information de Fisher des données complètes. C'est en quelque sorte un algorithme stochastique de scoring de Fisher. On voit bien qu'il suffit d'une et d'une seule donnée pour faire une nouvelle étape dans la séquence des estimateurs. Un inconvénient de cette procédure est l'inversion et l'évaluation de la matrice d'information de Fisher associée à la vraisemblance complète en $\hat{\theta}_{k-1}$ à l'itération $k$ qui sert de pondération, ce qui est exigeant en temps computationnel. L'algorithme est utile pour des problèmes dont le gradient de la log-vraisemblance incomplète est connu et dont la matrice d'information de Fisher de la vraisemblance complète est connue. Si le gradient de la log-vraisemblance de $g_\theta$ n'est pas connue, l'algorithme peut s'écrire comme

$$\hat{\theta}_t = \hat{\theta}_{t-1} + \gamma_t I_c^{-1}\left(\hat{\theta}_{t-1}\right)\mathbb{E}_{\hat{\theta}_{t-1}}\left[\nabla_\theta \log f\left(X_t,Y_t;\hat{\theta}_{t-1}\right) \mid Y_t\right]$$

que l'on obtient grâce à l'identité de Fisher évoquée en 12, où $I_c(\theta)$ est identique à l'équation 28. Le terme d'espérance n'est pas connu, habituellement. Toutefois, cela peut donner des idées pour construire d'autres approximations, comme l'ont fait les méthodes de Delyon *et al.* (1999) et Cappé et Moulines (2009).

## 2.4. L'algorithme EM en ligne par approximation stochastique

L'algorithme EM en ligne par approximation stochastique, tout comme l'EM, repose sur un processus itératif avec une étape E et une étape M. Toutefois, en tant qu'algorithme en ligne, il se distingue de l'algorithme standard en tenant compte d'une seule donnée par itération. Cela sous-entend qu'un nouvel estimateur du paramètre est disponible pour chaque observation traitée nouvellement disponible comme dans le cas de l'algorithme de Titterington. La récurrence à l'itération $t$ dans le cas général est :

$$\hat{Q}_t(\theta) = \hat{Q}_{t-1}(\theta) - \gamma_t \left( \mathbb{E}_{\hat{\theta}_{t-1}}[\log f(X_t,Y_t;\theta) \mid Y_t] - \hat{Q}_{t-1}(\theta) \right). \tag{29}$$

En écrivant $H(\hat{Q}) = \mathbb{E}_{\underset{u\in\Theta}{\operatorname{argmax}\hat{Q}(u)}}[\log f(X_t,Y_t;\theta) \mid Y_t] - \hat{Q}$, il devient clair que la récursion est de la forme ME et qu'il est possible de faire de l'approximation stochastique. Une telle approche s'avère avantageuse dans le contexte de données massives. Dans ce contexte, l'algorithme EM perd de l'efficacité computationnelle à mesure que la quantité de données traitées augmente. En effet, dans l'EM, l'espérance a posteriori de la vraisemblance complète de toutes les observations doit être mise à jour à chaque itération. Comme ce calcul tient compte de toutes les données observées, il est nécessaire de les conserver, ce qui consomme de la mémoire. De plus, l'EM n'est pas compatible dans les contextes où les données arrivent au fur et à mesure, puisqu'il fait l'inférence sur un jeu de données de $n$ données fixes. Un avantage en termes d'efficacité computationnelle de l'algorithme de Cappé et Moulines (2009) par rapport à la variante de Titterington (1984) est qu'il n'y a pas de matrice d'information de Fisher à estimer et à inverser. L'algorithme de Cappé et Moulines (2009) se distingue avantageusement de ceux vus jusqu'à présent puisqu'il est plus général. Des résultats de convergence sont établis sans même que le modèle soit bien spécifié, c'est-à-dire qu'il existe $\theta^* \in \Theta$ tel que $\pi = g_{\theta^*}$.

Pour la suite de ce mémoire, on s'intéresse plus particulièrement à l'algorithme EM en ligne par approximation stochastique pour des vraisemblances appartenant à la famille exponentielle. On va le présenter plus en détail dans ce cas-là.

### 2.4.1. Hypothèses du modèle

Les hypothèses suivantes sont nécessaires afin d'assurer la convergence de l'algorithme 2.4.1 (Cappé et Moulines, 2009).

(1) La vraisemblance des données complètes appartient à la famille exponentielle. La fonction de densité d'un modèle appartenant à la famille exponentielle peut s'écrire :

$$f(x,y;\theta) = h\,(x,y) \exp\left\{-\psi\,(\theta) + \langle S(x,y),\phi(\theta)\rangle\right\},$$

où $S(x,y)$ est un vecteur de statistiques exhaustives appartenant à $\mathbb{R}^m$ et l'opérateur $\langle\cdot,\cdot\rangle$ est le produit scalaire usuel entre deux vecteurs appartenant à $\mathbb{R}^m$. On définit les fonctions $\psi : \Theta \to \mathbb{R}^m$ et $\phi : \Theta \to \mathbb{R}^m$.

(2) La fonction $\bar{s}(y;\theta) \overset{\text{déf}}{=} \mathbb{E}_\theta[S(X,Y) \mid Y = y]$, l'espérance a posteriori est bien définie pour tout $(y,\theta) \in (\mathcal{Y},\Theta)$.

(3) Il existe un ensemble ouvert convexe $\mathcal{S} \subseteq \mathbb{R}^m$ tel que

i) $\forall s \in \mathcal{S}, (y,\theta) \in \mathcal{Y} \times \Theta$ et $\gamma \in [0,1), (1-\gamma)s + \gamma\bar{s}(y;\theta) \in \mathcal{S}$.

ii) $\forall s \in \mathcal{S}$, $\theta \mapsto \ell(s,\theta) \overset{\text{déf}}{=} \psi(\theta) + \langle s,\phi(\theta)\rangle$ possède un maximum global unique dans $\Theta$, défini comme suit :

$$\bar{\theta}(s) \overset{\text{déf}}{=} \underset{\theta\in\Theta}{\operatorname{argmax}}\ \ell(s,\theta).$$

L'algorithme EM en ligne par approximation stochastique est initialisé à l'aide d'une valeur quelconque de $s_0$. À l'aide de la relation entre $s_0$ et $\hat{\theta}_0$, on trouve $\hat{\theta}_0$. On dispose de la réalisation $y_1$, rendue disponible. À l'étape d'approximation stochastique, on évalue directement $\bar{s}\left(y_1;\hat{\theta}_0\right)$, ce qui permet de trouver $s_1$. À l'étape suivante, on déduit $\hat{\theta}_1$ qui maximise la fonction $\ell(s_1,\theta)$. À l'aide d'une nouvelle donnée $y_2$, on recommence le processus jusqu'à l'itération $N$.

**Algorithme 2.4.1.** L'algorithme EM en ligne par approximation stochastique (Cappé et Moulines, 2009).

---

**Entrée** Soit une suite positive décroissante $\{\gamma_t\}$, $s_0 \in \mathcal{S}$ et $N \in \mathbb{N}$ ;
**Initialisation** ;
$\hat{\theta}_0 = \bar{\theta}(s_0)$ ;
$k := 1$ ;
**Tant que** $k < N+1$ **:**
*Générer* $Y_k \sim \pi$ ;
*Calculer* $s_k = s_{k-1} - \gamma_k\left(s_{k-1} - \bar{s}\left(y_k;\hat{\theta}_{k-1}\right)\right)$ ;
$\hat{\theta}_k = \bar{\theta}(s_k)$ ;
$k \leftarrow k+1$ ;
**fin** ;
**retourner** $\hat{\theta}_N$ ;

---

L'algorithme 2.4.1 est une version bruitée de l'algorithme suivant. L'algorithme qui suit suppose la connaissance de $\pi$ tandis que le précédent suppose la connaissance de réalisations $Y_1,\ldots,Y_n$ suivant $\pi$. Le fait de connaître $\pi$ permet calculer exactement l'espérance dans l'étape d'approximation stochastique, alors que pour l'algorithme précédent, on avait un estimateur de cet espérance. Cet algorithme est déterministe conditionnellement aux valeurs

initiales et ne dépend pas des observations. On le nomme par ailleurs algorithme en ligne idéal.

**Algorithme 2.4.2.** L'algorithme EM en ligne idéal.

---

**Entrée** Soit une suite positive décroissante $\{\gamma_t\}$, $s_0 \in \mathcal{S}$ et $N \in \mathbb{N}$ ;
**Initialisation** ;
$\hat{\theta}_0 = \bar{\theta}(s_0)$ ;
$k := 1$ ;
**Tant que** $k < N + 1$ :
*Calculer* $s_k = s_{k-1} - \gamma_k \left( s_{k-1} - \mathbb{E}_\pi \left[ \bar{s} \left( y_k; \hat{\theta}_{k-1} \right) \right] \right)$ ;
$\hat{\theta}_k = \bar{\theta}(s_k)$ ;
$k \leftarrow k + 1$ ;
**fin** ;
**retourner** $\hat{\theta}_N$ ;

---

L'algorithme EM consiste en la séquence d'estimateurs $\{\hat{\theta}_t\}$ alors que l'algorithme ME équivalent est $\{\hat{s}_t\}$. L'approximation stochastique travaille sur les statistiques exhaustives et non sur les paramètres.

### 2.4.2. Problème vu sous forme d'approximation stochastique

Dans l'algorithme de Dempster *et al.* (1977), l'étape E de l'algorithme consiste à calculer $Q(\theta;\theta') = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\theta'}[\log(f(X_i,Y_i;\theta)) \mid Y_i = y_i]$. Lorsqu'on est en présence d'une vraisemblance appartenant à la famille exponentielle, la récursion 29 peut être simplifiée :

$$Q(\theta;\bar{\theta}(s)) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\bar{\theta}(s)} \left[ \ell \left( S\left(X_i,Y_i\right),\theta \right) \mid Y_i = y_i \right] \tag{30}$$

$$= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\bar{\theta}(s)} \left[ -\psi(\theta) + \langle S(X_i,Y_i), \phi\left(\theta\right) \rangle \right] \tag{31}$$

$$= \psi(\theta) + \left\langle \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\bar{\theta}(s)}[S(X_i,Y_i) \mid Y_i = y_i], \phi(\theta) \right\rangle . \tag{32}$$

On peut donc comparer l'algorithme EM avec l'algorithme 2.4.1. À la première ligne, on résume l'étape E et M de l'algorithme EM 1.3.4. À la deuxième ligne, on résume l'étape E et l'étape M de l'algorithme EM en ligne 2.4.1 dans les modèles appartenant à la famille exponentielle.

Étape M : $\hat{\theta}_{t-1} = \bar{\theta}(s_{t-1})$ Étape E : $s_t = \sum_{i=1}^{n} \frac{1}{n} \mathbb{E}_{\hat{\theta}_{t-1}}[S(X_i,Y_i) \mid Y_i = y_i]$

Étape M : $\hat{\theta}_{t-1} = \bar{\theta}(\hat{s}_{t-1})$ Étape E : $\hat{s}_t = (1-\gamma_t)\hat{s}_{k-1} + \gamma_t \left(\mathbb{E}_{\hat{\theta}_{t-1}}[S(X_t,Y_t) \mid Y = y_t]\right)$

Dans les deux cas, l'étape de maximisation est la même. La différence est située au niveau de l'étape du calcul de l'espérance. Dans la version en ligne, on obtient une espérance reposant sur une donnée plutôt que sur $n$ données. Une approximation de l'espérance de la statistique exhaustive est construite à partir d'une pondération entre l'espérance et l'approximation précédente ; c'est le principe d'approximation stochastique.

Par analogie à ce qui a été présenté à la section 2.2.2, on peut proposer une fonction $H(s,\xi) = \alpha(s,\xi) - s$. Il suffit de choisir $\alpha(s,\xi)$ tel que $\mathbb{E}_\xi[\alpha(s,\xi)] - s = \mathbb{E}_\pi[\mathbb{E}_{\bar{\theta}(s)}[S(X_i,Y_i) \mid Y_i]] - s = M(s)$. Dans ce cas-ci, le choix pour $t$ quelconque est $\alpha(s_{t-1},\xi) = \mathbb{E}_{\hat{\theta}_{t-1}}[S(X_t,Y_t) \mid Y_t = \xi_t]$ où $\xi$ est une collection iid de taille $n$, chaque élément de loi $\pi$.

### 2.4.3. Convergence

Définissons la divergence de Kullback-Leibler entre $\pi$ et $g_\theta$

$$K(\pi \| g_\theta) \stackrel{\text{déf}}{=} \mathbb{E}_\pi \left[\log\left(\frac{\pi(Y)}{g(Y;\theta)}\right)\right]. \tag{33}$$

La divergence de Kullback-Leibler permet de quantifier la dissimilarité entre deux distributions de probabilité. Une divergence nulle signifie que les lois sont identiques tandis qu'une grande divergence montre que les lois exhibent des différences importantes, au moins dans les zones de forte probabilité de $\pi$. Ce n'est pas une métrique, car la divergence ne respecte pas la propriété de symétrie ou l'inégalité du triangle, notamment.

L'algorithme EM en ligne 2.4.1 permet, pour les modèles appartenant à la famille exponentielle, de déterminer les paramètres qui minimisent la divergence de Kullback-Leibler $K(\pi \| g_\theta)$ entre la vraisemblance $\pi(y)$ et la vraisemblance sous le modèle $g(y;\theta)$ sans conserver chacune des données. Pour ce faire, il repose sur une approximation stochastique qui assure que la suite $\{\theta_t\}$ converge vers les points stationnaires de la divergence de Kullback-Leibler entre la vraisemblance incomplète (observée) et la vraisemblance complète (inobservée). À titre comparatif, l'EM traditionnel converge vers les points stationnaires de la vraisemblance des données observées, ce qui est une différence remarquable. L'estimateur de l'algorithme en ligne est en quelque sorte l'équivalent de l'estimateur de vraisemblance maximale lorsqu'on a un nombre infini d'observations iid qui viennent de $\pi$ et non de $g(y;\theta^*)$. La récursion de type Robbins-Monro de l'algorithme 2.4.1 est la suivante :

$$s_t = s_{t-1} + \gamma_t \left(\mathbb{E}_{\bar{\theta}(s_{t-1})}[S(X_t,Y_t) \mid Y_t = y_t] - s_{t-1}\right). \tag{34}$$

Intuitivement, après un nombre suffisant de récursions, la différence $\mathbb{E}_{\bar{\theta}(s_{t-1})}[s(X_t,Y_t) \mid Y_t = y_t] - s_{t-1}$ approche de plus en plus zéro. Par conséquent, $s_n$ approche de plus en plus de $s_{t-1}$ lorsque $t$ est grand. Clairement, la récursion 34 est une procédure d'approximation

stochastique qui vise à résoudre l'équation

$$M(s) \stackrel{\text{déf}}{=} \mathbb{E}_\pi \left[ \mathbb{E}_{\bar{\theta}(s)} \left[ S(X,Y) \mid Y \right] \right] - s = 0, \tag{35}$$

où $M : \mathcal{S} \to \mathbb{R}^d$. Puisque la loi $\pi$ n'est généralement pas connue, la solution est trouvée grâce à une approximation stochastique de Robbins-Monro.

**Proposition 2.4.3.** *Sous les hypothèses de la section 2.4.1, si $s^* \in \mathcal{S}$ est une racine de $M$, c'est-à-dire que $M(s^*) = 0$, alors $\theta^* = \bar{\theta}(s^*)$ est un point stationnaire de la fonction $K(\pi \| g_\theta)$, c'est-à-dire que $\nabla_\theta K(\pi \| g_\theta)_{|\theta=\theta^*} = 0$. De plus, si $\theta^*$ est un point stationnaire de $K(\pi \| g_\theta)$, alors $s^* = \mathbb{E}_\pi[\bar{s}(Y;\theta^*)]$ est une racine de $M$.*

La démonstration se trouve à l'annexe A.1.1.

La proposition montre que la solution de l'algorithme de Robbins-Monro est également la solution qui minimise la divergence de Kullback-Leibler et vice versa. L'estimateur proposé du paramètre $\theta$ est celui qui entraîne que $g_\theta$ est la plus similaire à la vraie loi de $Y$ au sens de la divergence de Kullback-Leibler. Soit $\Gamma \stackrel{\text{déf}}{=} \{s \in \mathcal{S} : M(s) = 0\}$ et $\Lambda \stackrel{\text{déf}}{=} \{\theta \in \Theta : \nabla_\theta K(\pi \| g_\theta) = 0\}$.

**Proposition 2.4.4.** *On peut montrer que la fonction $w : \mathcal{S} \to [0,\infty)$ qui est définie par*

$$w(s) \stackrel{déf}{=} K(\pi \| g_{\bar{\theta}(s)})$$

*est une fonction de Lyapounov pour la fonction $M$ et l'ensemble $\Gamma$, c'est-à-dire pour tout $s \in \mathcal{S}$, $\langle \nabla_s w(s), M(s) \rangle \leq 0$ et $\langle \nabla_s w(s), M(s) \rangle = 0$ quand $M(s) = 0$.*

La démonstration se trouve à l'annexe A.1.2.

Cette proposition permet de montrer la propriété de monotonie dans le cas de l'algorithme EM en ligne idéal qui est présenté en 2.4.2. La monotonie n'est vraie que si $\{\gamma_t\}$ est suffisamment petit.

**Remarque 2.4.5.** *La propriété 2.4.4 est analogue à la propriété de monotonie de l'algorithme EM, c'est-à-dire que $s_{k+1} = s_k + \gamma_{k+1}(M(s_k))$ fait décroître la divergence de Kullback-Leibler entre $\pi$ et $g(\cdot;\theta)$ quand $\{\gamma_t\}$ est suffisamment petit. À l'aide d'un développement de Taylor, on peut écrire*

$$\begin{aligned} w(s_n) - w(s_{n-1}) &\approx (s_n - s_{n-1})^T \nabla_s w(s_n) \\ &= (s_{n-1} - \gamma_n(\bar{s}(y_n, \bar{\theta}(s)) - s_{n-1}) - s_{n-1})^T \nabla_s w(s_{n-1}) \\ &= \gamma_n M(s_{n-1})^T \nabla_s w(s_{n-1}) \leq 0. \end{aligned}$$

La remarque 2.4.5 montre que dans le cas déterministe, c'est-à-dire, lorsque $\mathbb{E}_\pi \left[ \mathbb{E}_{\bar{\theta}(s)} \left[ S(X,Y) \mid Y \right] \right]$ est connue comme dans l'algorithme 2.4.2 et si le pas d'apprentissage $\gamma$ est suffisamment petit, alors la divergence de Kullback-Leibler décroît de

façon monotone pour atteindre son minimum. L'aspect technique de l'idée de la preuve de la convergence de l'algorithme 2.4.1 (EM en ligne) est de montrer qu'en présence d'un bruit, la monotonie est certes perdue, mais la convergence vers un point stationnaire est maintenue, pourvu que $\{\gamma_t\}$ soit suffisamment petit et que d'autres hypothèses de régularité soient remplies. On fera maintenant l'illustration de la monotonie associée à l'algorithme EM en ligne idéal à l'aide d'un exemple.

### 2.4.4. Exemple pour lequel l'algorithme idéal 2.4.2 est implémentable

Soit $K \in \mathbb{Q}$, une constante de troncature et $\delta \in \mathbb{Q}$, la fonction de masse de $Y$ est

$$\pi(y) = \omega_1^* \int_{y-\delta}^{y+\delta} \frac{1}{(2\pi\sigma_1^{2*})^{\frac{1}{2}}} \exp\left\{-\frac{1}{2\sigma_1^{2*}}(z-\mu_1^*)^2\right\} \mathrm{d}z + (1-\omega_1^*) \int_{y-\delta}^{y+\delta} \frac{1}{(2\pi\sigma_2^{2*})^{\frac{1}{2}}} \exp\left\{-\frac{1}{2\sigma_2^{2*}}(z-\mu_2^*)^2\right\} \mathrm{d}z \quad (36)$$

où $y \in \mathtt{Y} = \{\frac{1}{2\delta}y \in \mathbb{Z} \mid -K\frac{1}{2\delta} < \frac{1}{2\delta}y < K\frac{1}{2\delta}\}$ et $\theta^*$ est défini comme $(\omega_1^* = 0.55, \mu_1^* = 0, \mu_2^* = 5, \sigma_1^{2*} = 1, \sigma_2^{2*} = 4)$. Comme $\pi$ est connue, la récursion 34 est réécrite comme

$$s_n = s_{n-1} + \gamma_n \left(\mathbb{E}_\pi \left[\mathbb{E}_{\bar{\theta}(s_{n-1})}\left[S(X_n,Y_n) \mid Y_n\right]\right] - s_{n-1}\right), \quad (37)$$

qui est dorénavant implémentable puisqu'il est possible de calculer $\mathbb{E}_\pi[\mathbb{E}_{\bar{\theta}(s_{n-1})}[S(X_n,Y_n) \mid Y_n]]$ grâce à la discrétisation. La récursion 37 représente l'étape E du calcul d'espérance dans cet exemple. L'algorithme n'est plus stochastique, mais bien déterministe, conditionnellement au point initial, dans un tel cas. En effet, les données observées $y_1, \ldots, y_n$ n'interviennent pas dans la récursion. Un exemple de calcul d'espérance est le suivant :

$$\mathbb{E}_\pi \left[\mathbb{E}_{\bar{\theta}(s_{n-1})}\left[S_{2,i}(X_n,Y_n) \mid Y_n\right]\right] = \sum_{y_j \in \mathtt{Y}} y_j \frac{\Phi\left(\frac{y_j-\delta-\hat{\mu}_{i,n-1}}{\hat{\sigma}_{i,n-1}}\right) - \Phi\left(\frac{y_j+\delta-\hat{\mu}_{i,n-1}}{\hat{\sigma}_{i,n-1}}\right)}{\sum_{i=1}^2 \left(\Phi\left(\frac{y_j-\delta-\hat{\mu}_{i,n-1}}{\hat{\sigma}_{i,n-1}}\right) - \Phi\left(\frac{y_j+\delta-\hat{\mu}_{i,n-1}}{\hat{\sigma}_{i,n-1}}\right)\right)} \pi(y_j),$$

ce qui montre qu'il n'est plus nécessaire d'observer $Y$ pour calculer l'espérance. La remarque 2.4.5 montre que lorsque $\pi$ est connue et que la récursion 37 est calculable, la divergence de Kullback-Leibler décroît de manière monotone vers un minimum.

Dans cet exemple où on a posé $K = 0{,}012$, $\delta = 0{,}005$, $\hat{\theta}_0 = (0{,}633; 0{,}185; 5{,}66; 1{,}23; 2{,}63)$. La divergence de Kullback-Keibler tend vers le minimum au fil des itérations comme on voit à la figure 2.2. De surcroît, les paramètres ont une convergence lisse telle que le montre la figure 2.1, contrairement aux algorithmes en ligne.

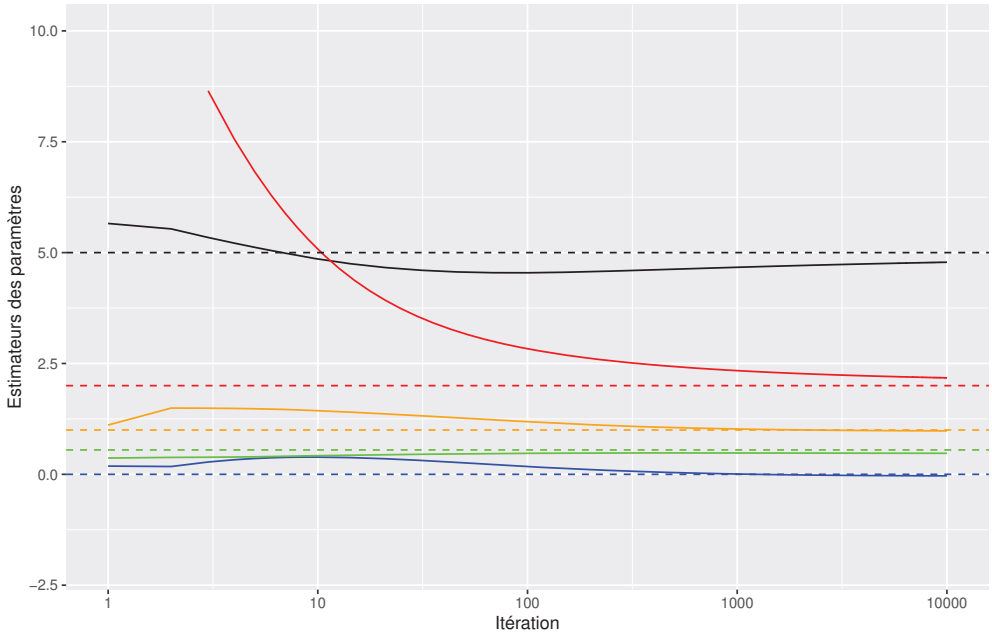


**Figure 2.1.** Convergence des paramètres pour l'algorithme EM en ligne idéal ($\omega_1$ en vert, $\mu_1$ en bleu, $\mu_2$ en noir, $\sigma_1$ en orange et $\sigma_2$ en rouge) dans le cadre l'exemple de la section 3.1. Les estimateurs des paramètres convergent vers la vraie valeur des paramètres.

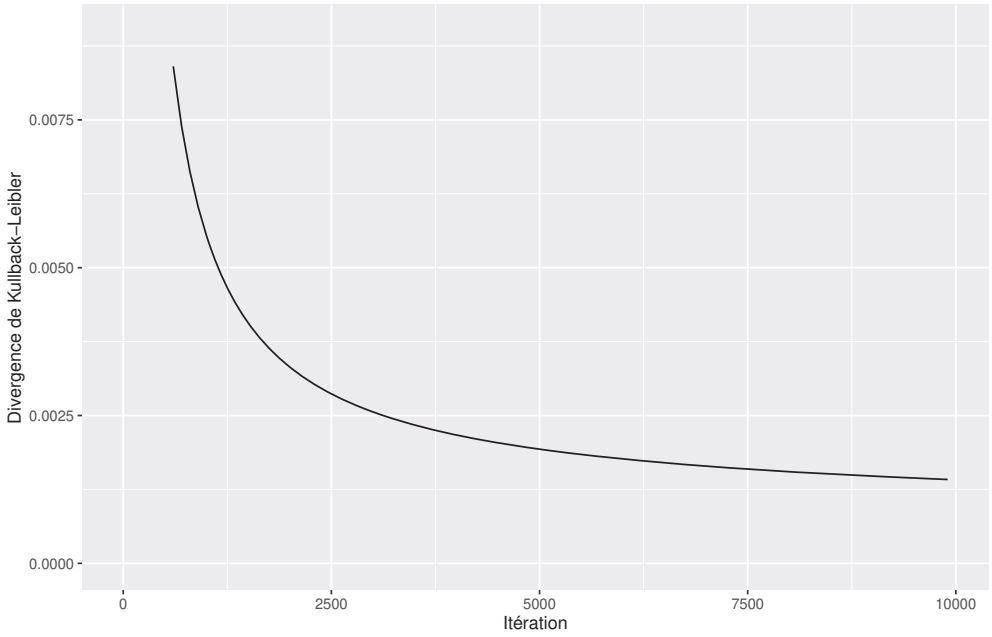


**Figure 2.2.** Divergence de Kullback-Leibler entre $\pi$ et $g_\theta$ pour l'algorithme EM en ligne idéal 2.4.2, dans le cadre du modèle de l'exemple d'un mélange de distributions normales. La divergence converge vers le minimum au fil des itérations.

**Théorème 2.4.6.** *Soit* $x \in \mathbb{R}^m$, $A \subset \mathbb{R}^m$, $d(x,A) = \inf\{y \in A, |x-y|\}$. *Sous les hypothèses,* $\lim_{n\to\infty} d(s_n,\Gamma) = 0$, *presque sûrement et* $\lim_{n\to\infty} d(\theta_n,\Lambda) = 0$, *presque sûrement.*

La démonstration se trouve à l'annexe A.1.3.

Le théorème ci-dessus montre que les séquences de $\{s_t\}$ et $\{\theta_t\}$ convergent respectivement presque sûrement vers l'ensemble solution de l'équation $M(s) = 0$ et vers l'ensemble des valeurs qui minimisent $K(\pi\|g_\theta)$. Ainsi, il n'est plus nécessaire que le modèle soit bien spécifié, c'est-à-dire qu'il existe $\theta^\star$ tel que $\pi = g_{\theta^\star}$ (Cappé et Moulines, 2009).

### 2.4.5. Moyenne de Polyak-Ruppert

À condition que $\gamma_n = \gamma_0 n^{-\alpha}$ avec $\alpha \in (\frac{1}{2},1)$ et $\gamma_0 = (0,1)$, il est possible d'obtenir un estimateur suivant asymptotiquement une loi normale. On prend la moyenne mobile

$$\tilde{\theta}_n = \frac{1}{n - n_0} \sum_{i=n_0+1}^{n} \hat{\theta}_i. \tag{38}$$

La séquence $\{\tilde{\theta}_t\}$ converge vers la loi normale à un taux de $1/n^{\frac{1}{2}}$, pour tout $\gamma_0$ (Cappé et Moulines, 2009). L'équation 38 est une moyenne de Polyak-Ruppert (Polyak, 1990; Ruppert, 1988). La matrice de covariance asymptotique est définie par

$$\left\{-\nabla_\theta^2 K\left(\pi\|g_\theta\right)\mid_{\theta=\theta^*}\right\}^{-1} \mathbb{E}_\pi\left[\nabla_\theta \log\left(g_{\theta^*}\right)\left(\nabla_\theta \log\left(g_{\theta^*}\right)\right)^T\right] \left\{-\nabla_\theta^2 K\left(\pi\|g_\theta\right)\mid_{\theta=\theta^*}\right\}^{-1}. \tag{39}$$

Quand il existe $\theta = \theta^\star$ tel que $\pi = g_\theta$, la matrice de covariance se réduit à l'inverse de la matrice $\mathbb{E}_\pi\left[\nabla_\theta \log(g_{\theta^*})(\nabla_\theta \log(g_{\theta^*}))^T\right]$ qui est en fait la matrice d'information de Fisher observée puisque

$$\begin{aligned}
I(\theta^\star) &= -\mathbb{E}_{\theta^\star}\left[\nabla_\theta^2 \log g(Y;\theta)\mid_{\theta=\theta^\star}\right] = -\nabla_\theta^2\left[\int g(y;\theta^\star)\log g(y;\theta)\nu(dy)\right]\mid_{\theta=\theta^\star} \\
&= \nabla_\theta^2 \int g(y;\theta^\star)\log g(y;\theta^\star)\nu(dy) - \nabla_\theta^2\left[\int g(y;\theta^\star)\log g(y;\theta)\nu(dy)\right]\mid_{\theta=\theta^\star} \\
&= \nabla_\theta^2\left[\int g(y;\theta^\star)\log\left(\frac{g(y;\theta^\star)}{g(y;\theta)}\right)\nu(dy)\right]\mid_{\theta=\theta^\star} \\
&= \nabla_\theta^2 K\left(g\left(y;\theta^\star\right)\|g\left(y;\theta\right)\right)\mid_{\theta=\theta^\star} \\
&= \nabla_\theta^2 K\left(g_{\theta^\star}\|g_\theta\right)\mid_{\theta=\theta^\star} \\
&= \nabla_\theta^2 K\left(\pi\|g_\theta\right)\mid_{\theta=\theta^\star},
\end{aligned}$$

et $I(\theta^\star) = -\mathbb{E}_{\theta^\star}[\nabla_\theta^2 \log g(Y;\theta^\star)] = -\mathbb{E}_\pi[\nabla_\theta^2 \log g(Y;\theta^\star)] = \mathbb{E}_\pi\left[\nabla_\theta \log\left(g_{\theta^*}\right)\left(\nabla_\theta \log\left(g_{\theta^*}\right)\right)^T\right]$. Ainsi, la séquence $n^{\frac{1}{2}}(\tilde{\theta}_n - \theta^*)$ possède une matrice de covariance équivalente asymptotiquement à l'inverse de la matrice d'information de Fisher observée (Cappé et Moulines, 2009).

## 2.5. L'algorithme EM en ligne par approximation stochastique Monte-Carlo

Lorsqu'il est impossible de calculer l'espérance $\mathbb{E}_{\bar{\theta}(s_k)}[S(X_k,Y_k) \mid Y_k = y_k]$, on peut l'estimer à l'aide d'une simulation de Monte-Carlo. La convergence d'algorithmes EM stochastique dont l'approximation stochastique est basée sur un estimateur Monte-Carlo a été abordée dans Delyon *et al.* (1999), entre autres. L'algorithme qui suit, contrairement à ce dernier, est en ligne. On peut voir l'approche suivante comme étant inspirée des idées de l'algorithme de Delyon *et al.* (1999) et Cappé et Moulines (2009). On génère $m$ réalisations à l'itération $k$ provenant des variables aléatoires $X_{k,1}, \ldots, X_{k,m}$ suivant la loi a posteriori de $X$. Pour obtenir l'estimateur, il suffit de prendre la moyenne arithmétique des réalisations générées,

$$\hat{\mu}_{m,k}^{MC} \overset{\text{déf}}{=} \frac{1}{m} \sum_{i=1}^{m} S(X_{k,i},Y_k). \tag{40}$$

L'algorithme EM en ligne par approximation stochastique Monte-Carlo est pratiquement le même que l'algorithme 2.4.1. Comme c'était le cas pour l'algorithme 2.4.1, on obtient $\hat{\theta}_0$ à l'aide $s_0$. On connaît $y_1$, rendue disponible. L'étape supplémentaire par rapport à l'algorithme 2.4.1 est la génération des réalisations $X_{1,1}, \ldots X_{1,m}$ de la loi $X_1$ que l'on échantillonne directement, sa loi étant connue. L'étape d'approximation stochastique, qui permet de calculer $s_1$, substitue $\bar{s}\left(y_1; \hat{\theta}_0\right)$ par $\hat{\mu}_{m,1}^{MC}$. La maximisation de $\ell(s_1,\theta)$ s'effectue de la même façon que l'algorithme 2.4.1. À l'aide du $\hat{\theta}_1$ obtenu, on peut répéter le processus jusqu'à l'itération $N$.

**Algorithme 2.5.1.** L'algorithme EM par approximation stochastique en ligne Monte-Carlo.

---

**Entrées** Soit une suite décroissante $\{\gamma_t\} \in \mathbb{R}^+$, $s_0 \in \mathcal{S}$, $m, N \in \mathbb{N}$ ;
**Initialisation** ;
  $\hat{\theta}_0 = \bar{\theta}(s_0)$ ;
  $k := 1$ ;
**Tant que** $k < N + 1$ **:**
  *Générer* $Y_k \sim \ \pi$ ;
  *Générer* $X_{k,1}, \ldots X_{k,m}$ *iid* à *partir de la loi de* $P_{\hat{\theta}_{k-1}}(x_k; y_k)$ ;
  *Calculer* $\hat{\mu}_{m,k}^{MC} = \frac{1}{m} \sum_{i=1}^{m} S(X_{k,i}, Y_k)$ ;
  *Calculer* $s_k = s_{k-1} - \gamma_k \left(s_{k-1} - \hat{\mu}_{m,k}^{MC}\right)$ ;
  $\hat{\theta}_k = \bar{\theta}(s_k)$ ;
  $k \leftarrow k + 1$ ;
**fin** ;
**retourner** $\hat{\theta}_N$ ;

---

On peut proposer une fonction $H(s,\xi) = \alpha(s,\xi) - s$. Il suffit de choisir $\alpha(s,\xi)$ tel que $\mathbb{E}_\xi[\alpha(s,\xi) - s] = \mathbb{E}_\pi[\mathbb{E}_{\bar{\theta}(s)}[S(X_i,Y_i) \mid Y_i] - s] = M(s)$. Dans ce cas-ci, le choix est $\alpha(s,\xi) = \hat{\mu}_{m,k}^{MC}$ et $\xi$ qui est $n$ séquences iid de longueur $m$ (qui peut être pris égale à 1), chaque séquence $k$ ayant pour loi $p(\cdot; y_k, \hat{\theta}_{k-1})\pi(y_k)$. On pourrait qualifier cet algorithme de « doublement stochastique » puisque l'espérance sous la jointe $\pi p$ est estimée en échantillonnant $\pi$ et la loi a posteriori $p$. Dans l'algorithme 2.2.1 de Delyon, on échantillonne juste la loi a posteriori $p$ et dans l'algorithme 2.4.1, on échantillonne juste $\pi$.

## 2.6. L'algorithme EM en ligne par approximation stochastique MCMC

Dans le cas où la distribution conditionnelle de la variable latente n'est pas disponible explicitement, on génère des réalisations de cette loi à l'aide d'un échantillonnage MCMC. Dans cet algorithme, l'espérance de la statistique exhaustive $S(X,Y)$ de type MCMC est un estimateur biaisé. Toutefois, il est asymptotiquement sans biais alors on utilise tout de même l'approximation stochastique. Après une période de « burn-in » de $B$ itérations, à l'itération $k$ de l'algorithme, on calcule $S(X_{k,B+1},Y_k)$, ...,$S(X_{k,m},Y_k)$. On retrouve l'espérance MCMC en prenant une moyenne arithmétique.

L'espérance MCMC est définie par

$$\hat{\mu}_k^{MCMC} \stackrel{\text{déf}}{=} \frac{1}{m-B} \sum_{i=B+1}^{m} S\left(X_{k,i},Y_k\right). \tag{41}$$

L'algorithme EM en ligne par approximation stochastique MCMC est semblable à l'algorithme 2.5.1. Comme c'était le cas pour les algorithmes 2.4.1 et 2.5.1, on obtient $\hat{\theta}_0$ à l'aide $s_0$. Cette fois-ci, on génère des réalisations $X_{1,1}, \ldots X_{1,m}$ de la loi $X_1$ à l'aide de l'algorithme de Metropolis, sa loi étant potentiellement impossible à échantillonner directement, mais connue à une constante de normalisation près. L'étape d'approximation stochastique, qui permet de calculer $s_1$, substitue $\bar{s}\left(y_1; \hat{\theta}_0\right)$ par $\hat{\mu}_1^{MCMC}$. La maximisation de $\ell(s_1,\theta)$ s'effectue de la même façon que pour les algorithmes 2.4.1 et 2.5.1. À l'aide du $\hat{\theta}_1$ obtenu, on peut répéter le processus jusqu'à l'itération $N$.

**Algorithme 2.6.1.** L'algorithme EM par approximation stochastique en ligne MCMC.

**Entrées** Une suite décroissante positive $\{\gamma_t\} \in \mathbb{R}$, $s_0 \in \mathcal{S}$, $\sigma^2 \in \mathbb{R}^+$, $N, m, B \in \mathbb{N}, B < m$ ;
**Initialisation** ;
$\hat{\theta}_0 = \bar{\theta}(s_0)$ ;
$k := 1$ ;
**Tant que** $k < N + 1$ **:**
*Générer* $Y_k \sim \pi$ ;
Générer $X_{k,1}, \ldots, X_{k,m}$ avec l'algorithme de Metropolis à incrément gaussien invariant pour $f(x,y;\theta)$ et le noyau de proposition $K \sim \mathcal{N}(x,\sigma^2)$ ;
*Calculer* $s_k = s_{k-1} - \gamma_k \left(s_{k-1} - \hat{\mu}_k^{MCMC}\right)$ ;
$\hat{\theta}_k = \bar{\theta}(s_k)$ ;
$k \leftarrow k + 1$ ;
**fin** ;
**retourner** $\hat{\theta}_N$ ;

On peut proposer une fonction $H(s,\xi) = \alpha(s,\xi) - s$. Il suffit de choisir $\alpha(s,\xi)$ tel que $\mathbb{E}_\xi[\alpha(\xi) - s] = \mathbb{E}_\pi[\mathbb{E}_{\theta(s)}[S(X_i,Y_i) \mid Y_i]] - s = M(s)$. Dans ce cas-ci, le choix est $\alpha(s,\xi) = \hat{\mu}_k^{MCMC}$. La variable aléatoire $\xi$ est $n$ séquences iid de longueur $m - B$ (qui peut être pris égale à 1), chaque séquence $k$ ayant pour loi approximative $p(\cdot; y_k,\hat{\theta}_{k-1})\pi(y_k)$, où $p$ est la loi stationnaire de la chaîne de Markov $\{X_i; i \in \mathbb{N}\}$ dont le noyau de proposition est $K$.

## 2.7. L'algorithme EM incrémental mini-lots

Les algorithmes EM incrémentaux de type « mini-lots », *mini-batch* en anglais, offre un compromis entre la vitesse de convergence de l'algorithme EM et l'efficacité computationnelle de l'algorithme en ligne. On souligne la pertinence des algorithmes du type mini-lots dont le calcul à chaque itération repose sur un plus petit nombre d'observations. La convergence est plus rapide que dans les versions où les données sont traitées en ligne (Neal et Hinton, 1998; Karimi *et al.*, 2019a). De plus, l'efficacité computationnelle est supérieure à celle dans la version standard, ce qui en fait un compromis intéressant. Les algorithmes reposant sur l'approche mini-lots exigent la connaissance des données à l'avance. Cette contrainte n'est pas compatible à la situation où les données deviennent disponibles au fur et à mesure. L'algorithme susmentionné de Chen *et al.* (2018) est un exemple d'algorithme mini-lots.

### 2.7.1. Principe

Le principe mini-lots consiste à utiliser un sous-ensemble des données à chaque itération. En particulier, l'algorithme de Neal et Hinton (1998) met à jour un sous-ensemble des

termes de l'espérance des statistiques exhaustives $\bar{s}(y;\theta)$ à chaque itération à partir des observations $y_1,\ldots,y_n$ disponibles. Lorsque ce dernier tient compte des $n$ observations, il est équivalent à la version standard de l'EM (Neal et Hinton, 1998). L'algorithme de Neal et Hinton (1998) a inspiré des versions Monte-Carlo et MCMC de l'algorithme EM mini-lots qui sont étudiées dans Karimi *et al.* (2019a). Il est montré que sous certaines hypothèses, ces algorithmes convergent presque sûrement (Karimi *et al.*, 2019a). Ce ne sont pas des procédures d'approximation stochastique contrairement à l'algorithme EM en ligne de Cappé et Moulines (2009). Un inconvénient des algorithmes mini-lots est qu'il n'y pas de certitude que la vraisemblance augmente à chaque itération (Karimi *et al.*, 2019a). Une approche de type mini-lots comprenant une approximation stochastique est celle qu'on retrouve dans Kuhn *et al.* (2019). Ces approximations stochastiques n'ont pas la même fonction $M$ que l'approche en ligne. En présence de méthodes MCMC dans l'étape d'approximation stochastique, une approche mini-lots consiste à mettre à jour un sous-ensemble des composantes de la variable latente par MCMC lorsque la variable latente est en grande dimension Kuhn *et al.* (2019). Dans Kuhn *et al.* (2019), il est montré que si l'algorithme EM incrémental MCMC de Kuhn et Lavielle (2004) converge, l'algorithme EM incrémental MCMC mini-lots converge aussi. Ce n'est pas l'objet de ce mémoire, car les exemples des chapitres 3 et 4 contiennent uniquement des exemples où la variable latente est univariée.

Un algorithme de type mini-lots général est présenté ci-dessous. C'est un algorithme mini-lots d'approximation stochastique, contrairement à celui de Neal et Hinton (1998). Les algorithme mini-lots étudiés dans Karimi *et al.* (2019a) ne sont pas des algorithmes d'approximation stochastique non plus, mais demeurent néanmoins stochastiques.

L'algorithme EM incrémental par approximation stochastique de type « mini-lots » est semblable à l'algorithme 2.4.1, mais n'est plus en ligne. Tout d'abord, on obtient $\hat{\theta}_0$ à partir de $s_0$. On dispose des réalisations $y_1,\ldots,y_n$. On tire $R$ nombres sans remise entre 1 et $n$. On retient l'ensemble $I_1$ de ces indices. Ensuite, à l'étape d'approximation stochastique, on calcule directement la quantité $R^{-1}\sum_{i\in I_1}\bar{s}\left(y_i;\hat{\theta}_0\right)$, ce qui permet d'obtenir $s_1$. La fonction $\ell(s_1,\theta)$ est maximisée de la même façon que dans les algorithmes 2.4.1, 2.5.1 et 2.6.1. À l'aide de $\hat{\theta}_1$, on peut répéter le processus jusqu'à l'itération $N$.

**Algorithme 2.7.1.** L'algorithme EM incrémental par approximation stochastique de type « mini-lots ».

---

**Entrées** Soit une suite décroissante positive $\{\gamma_t\} \in \mathbb{R}^+$, $s_0 \in \mathcal{S}$, $N, R, m \in \mathbb{N}$, $R < n$ ;
**Initialisation** ;
$\hat{\theta}_0 = \bar{\theta}(s_0)$ ;
$k := 1$ ;
*Générer* $Y_1, \ldots, Y_n \sim \pi$ ;
**Tant que** $k < N + 1$ :
Choisir $R$ entiers aléatoirement parmi $1, \ldots, n$ et définir $I_k$, l'ensemble de ces indices ;
*Calculer* $\hat{\mu}_k^{MB} = \frac{1}{|I_k|} \sum_{i \in I_k} \bar{s}(y_i; \hat{\theta}_{\text{k}-1})$ ;
*Calculer* $s_k = s_{k-1} - \gamma_k \left( s_{k-1} - \hat{\mu}_k^{MB} \right)$ ;
$\hat{\theta}_k = \underset{\theta}{\text{argmax}}\ \ell(s_k; \theta)$ ;
$k \leftarrow k + 1$ ;
**fin** ;
**retourner** $\hat{\theta}_N$ ;

---

Par analogie à ce qui a été présenté à la section 2.2.2, on peut proposer une fonction $H(s,\xi) = \alpha(s,\xi) - s$. Il suffit de choisir $\alpha(s,\xi)$ tel que $\mathbb{E}_\xi[\alpha(s,\xi) - s] = \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{\bar{\theta}(s)}[S(X_i,Y_i) \mid Y_i] - s = M(s)$. Dans ce cas-ci, le choix est $\alpha(s,\xi) = \hat{\mu}_k^{MB} = \frac{1}{|I_k|} \sum_{i \in I_k} \bar{s}(y_i; \hat{\theta}_{\text{k}-1}) = \frac{1}{|I_k|} \sum_{i=1}^n \mathbb{1}(i \in I_k) \bar{s}(y_i; \hat{\theta}_{\text{k}-1})$ et $\xi = I$, $I$ suivant un plan d'échantillonnage aléatoire simple sans remise de taille $R$.

### 2.7.2. L'algorithme EM par approximation stochastique à variance réduite

L'algorithme du gradient stochastique à variance réduite (1.6.4) de Johnson et Zhang (2013), a inspiré un algorithme à variance réduite pour l'algorithme EM en ligne (Chen *et al.*, 2018). Cet algorithme peut être mis à profit uniquement lorsqu'on dispose du jeu complet de données disponibles en tout temps. De la même façon que la variance des estimateurs issus de l'algorithme du gradient stochastique est typiquement plus grande que celle issue de l'algorithme du gradient, un inconvénient des approches en ligne (ou mini-lots) est que la variance des estimateurs est souvent observée comme étant plus grande que celle de l'algorithme EM équivalent. C'est un autre exemple d'algorithme mini-lots sauf qu'on fait plusieurs itérations avec un mini-lot complet à chaque itération. Ce mini-lot est utilisé afin de créer une variable qui stabilise l'étape d'approximation stochastique. La réduction

de variance a donc été combinée à un algorithme incrémental. Compte tenu de la discussion ci-dessus, l'algorithme n'est pas en ligne. Au chapitre 4, on proposera une approche de réduction de variance applicable pour un algorithme en ligne d'approximation stochastique.

L'algorithme EM incrémental de réduction de variance est une autre variante de l'algorithme EM en ligne. Il y incorpore une variable de contrôle dans l'étape d'approximation stochastique. Dans la première boucle itérée par $k$, à l'aide de $s_{1,0}$, on calcule directement, pour $k = 1$, $\tilde{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\bar{\theta}(s_{k,0})}\left[S(X_i,Y_i) \mid Y_i = y_i\right] = \frac{1}{n}\sum_{i=1}^{n} \bar{s}\left(y_i; \bar{\theta}\left(s_{k,0}\right)\right)$. Ensuite, dans la deuxième boucle itérée par $j$, on choisit un seul indice $i_1$, au hasard entre 0 et $n$ afin de prendre une réalisation parmi $y_1, \ldots, y_n$. On calcule ensuite directement l'approximation stochastique qui met en oeuvre, une variable de contrôle comprenant $\bar{s}\left(y_{i_j}; \bar{\theta}\left(s_{k,0}\right)\right)$ et $\tilde{\mu}$, obtenu dans l'autre boucle. À chaque itération, on obtient un nouveau $s_{k,j}$. On tire ensuite $i_2$, et on répète les étapes. Après avoir parcouru $j = 1, \ldots, m$, on pose $s_{2,0} = s_{1,m}$. On peut ainsi retourner dans la première boucle et recommencer le processus jusqu'à l'itération $k = N$.

**Algorithme 2.7.2.** L'algorithme EM incrémental de réduction de variance (Chen *et al.*, 2018).

---

**Entrées** Soit une suite décroissante positive $\{\gamma_t\} \in \mathbb{R}^+$, $s_{1,0} \in \mathcal{S}$, $m$, $n$, $N \in \mathbb{N}$ ;

**Itération : Pour** $k = 1,2, \ldots, N$ **;**

$\tilde{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\bar{\theta}(s_{k,0})}\left[S(X_i,Y_i) \mid Y_i = y_i\right] = \frac{1}{n}\sum_{i=1}^{n} \bar{s}\left(y_i; \bar{\theta}\left(s_{k,0}\right)\right)$ **;**

**Itération : Pour** $j$ **:** $1, 2, \ldots, m$ **;**

*Choisir aléatoirement* $i_j \in \{1, \ldots, n\}$ ;

$s_{k,j} = (1 - \gamma_j)\, s_{k,j-1} + \gamma_j \left(\bar{s}\left(y_{i_j}; \bar{\theta}\left(s_{k,j-1}\right)\right) - \bar{s}\left(y_{i_j}; \bar{\theta}\left(s_{k,0}\right)\right) + \tilde{\mu}\right)$ ;

**fin ;**

$s_{k+1,0} \leftarrow \bar{s}\left(y_m; \bar{\theta}\left(s_{k,m}\right)\right)$ **;**

**fin ;**

---

# Chapitre 3

# Comparaison empirique des algorithmes d'apprentissage en ligne Monte-Carlo et MCMC

Afin d'étudier le comportement des différents algorithmes, on procède à une série d'expériences aléatoires conduites sur un logiciel muni d'un générateur de nombres aléatoires. La simulation consiste à répéter 1000 fois une expérience dans laquelle on génère $n = 10^4$ réalisations d'une variable aléatoire $Y$, qui suit par exemple un mélange, ou une variable aléatoire $Y$ qui est une variable de réponse dans un modèle de régression. Une expérience consiste en l'application de l'un des algorithmes EM en ligne à une séquence de $10^4$ données afin de trouver un estimateur de $\theta$, le paramètre du modèle. Avec ces 1000 réplicats ou répétitions d'expérience, la distribution des estimateurs $\hat{\theta}$ pour les divers algorithmes peut être approchée. En clair, on répète 1000 fois une expérience comprenant $10^4$ données et les données sont différentes d'une expérience à l'autre. Les algorithmes en ligne qui nous intéressent ici utilisent des estimateurs de la quantité $\bar{s}(\theta,y) = \mathbb{E}_{\tilde{\theta}(s)}[S(X,Y) \mid Y = y]$, qui intervient systématiquement dans les algorithmes EM pour ce type de modèle. L'une des questions principales est de comprendre comment de tels estimateurs impactent la convergence des estimateurs de $\theta$. De plus, on s'intéresse aux possibles conséquences sur le biais et la variance des estimateurs de $\theta$. Approche-t-on l'estimateur qui minimise la valeur minimale de la divergence de Kullback-Leibler aussi rapidement que lorsque la quantité peut être calculée ? Conserve-t-on la normalité asymptotique ?

L'algorithme de référence est l'algorithme EM en ligne 2.4.1 dont les propriétés théoriques sont assez bien comprises dans le cadre des modèles exponentielles. Quand on remplace l'espérance a posteriori par un estimateur, les conséquences sont moins connues. Dans ce mémoire, on fait des expériences pour voir si les algorithmes EM en ligne Monte-Carlo 2.5.1 et EM en ligne MCMC 2.6.1 héritent des propriétés théoriques de Cappé et Moulines (2009). Si oui, on voudrait vérifier si l'aspect quantitatif, comme les propriétés de normalité

asymptotique et de variance asymptotique, est conservé. Un enjeu sous-jacent est celui du passage à la limite. Lorsqu'on dispose d'un estimateur sans biais et que la variance de celui-ci tend vers zéro, l'algorithme converge-t-il vers le même point que l'algorithme 2.4.1, et si oui, en quel sens ?

Ces questions sont analysées à l'aide d'un exemple de mélange de distributions normales (avec $\mu$ et $\sigma^2$ inconnus) et d'un exemple de mélange de régression linéaire multiple (avec les paramètres de régression $\beta_0$, $\beta_1$ et $\beta_2$ inconnus). Les mélanges mettent en œuvre deux poids $\omega \in (0,1)$ et $1-\omega \in (0,1)$ représentant chacun la probabilité d'appartenir à l'une ou l'autre des classes.

Pour chaque estimateur provenant des algorithmes en ligne, on présente des graphiques de convergence, les densités empiriques, les boîtes à moustaches et des tableaux sur l'efficacité relative de ceux-ci. Les graphiques de convergence pour les paramètres d'intérêt sont présentés pour seulement 5 jeux de données quelconques, chacun associé à une couleur sur le graphique ; l'algorithme dispose d'un plus grand nombres de réalisations de $Y$ que $10^4$ puisqu'on ne répète pas 1000 fois l'expérience. Les graphiques de densités empiriques, les boîtes à moustaches et l'efficacité relative, quant à eux, reposent sur 1000 expériences de $10^4$ données. Les exigences de mémoire computationnelle sont élevées lorsque l'algorithme en ligne traite 1000 séquences de taille $10^5$ ou $10^6$, on se limite donc à $10^4$ données par expérience dans ce-cas là. Notons qu'on affiche uniquement 1% des données pour les graphiques de convergence. La statistique d'efficacité relative cherche à quantifier la dégradation de l'algorithme EM en ligne approximatif comparativement au cas idéal où on saurait calculer l'espérance a priori. Pour obtenir une statistique de l'efficacité relative, le rapport des variances est effectué. Cependant, la variance n'est pas robuste aux valeurs aberrantes qui pourraient survenir quand l'algorithme n'a pas encore convergé. Afin d'avoir une mesure plus robuste, le rapport des écarts absolus médians au carré est présenté également. La mesure de variabilité de l'algorithme EM en ligne par approximation stochastique 2.4.1 est placée au numérateur tandis que la mesure de variabilité de l'algorithme concurrent, comme l'algorithme en ligne par approximation stochastique Monte-Carlo (2.5.1) ou l'algorithme en ligne par approximation stochastique MCMC (2.6.1), est placée au dénominateur. On ne considère pas le rapport de variance avec l'algorithme EM standard, puisqu'on s'intéresse à l'augmentation de variance causée par des méthodes Monte-Carlo et MCMC dans les algorithmes en ligne. Ces statistiques quantifient donc bien la dégradation de la variance. Des valeurs proches de 1 indiquent une faible dégradation tandis que des valeurs faibles indiquent une forte dégradation.

La section 3.1 contient un premier exemple simple où on ne s'attend pas à beaucoup de variabilité sur l'estimation de l'espérance a posteriori. La section 3.2 contient un second exemple pour lequel il y a un peu de plus de variabilité dans l'estimation de l'espérance a posteriori. À chaque fois, les algorithmes convergent vers le minimum de $K(\pi\|g_\theta)$ et la distribution des estimateurs est qualitativement normale après $10^4$ itérations. Cependant, la

variabilité des estimateurs augmente en utilisant les algorithmes 2.5.1 et 2.6.1. Des variantes de l'exemple de régression où la covariable $X$ est latente sont proposées en fin de chapitre. Une analyse plus approfondie de ces modèles est réservée au chapitre 4 où l'on proposera certaines pistes pour réduire la variance. Les sections 3.4.1 et 3.4.3 présentent deux exemples de modèles où la variable latente est continue.

## 3.1. Exemple d'un mélange de deux distributions normales

Soit $Y$ une variable aléatoire dont la loi est

$$\pi(y) = \omega_1^* \frac{1}{(2\pi\sigma_1^{2*})^{\frac{1}{2}}} \exp\left\{-\frac{1}{2\sigma_1^{2*}}(y-\mu_1^*)^2\right\} + (1-\omega_1^*)\frac{1}{(2\pi\sigma_2^{2*})^{\frac{1}{2}}} \exp\left\{-\frac{1}{2\sigma_2^{2*}}(y-\mu_2^*)^2\right\}, \tag{42}$$

où on a utilisé les valeurs

$$\theta^* = (\omega_1^* = 0.55, \mu_1^* = 0, \mu_2^* = 5, \sigma_1^{2*} = 1, \sigma_2^{2*} = 4). \tag{43}$$

Supposons que la loi $\pi$ ne soit pas connue, mais qu'on souhaiterait néanmoins modéliser $Y$ dont on observe des réalisations iid. On spécifie le modèle bivarié suivant, comme il est soupçonné que $Y$ dépende d'une variable aléatoire $X \in \{1,2\}$ indiquant la classe,

$$f\,(x,y;\theta) = \prod_{i=1}^{2} \omega_i^{\mathbb{1}(x=i)} \ \exp\left\{\mathbb{1}(x \ \ = i)\left[\frac{1}{2\sigma_i^2}(y-\mu_i)^2 - \frac{1}{2}\log\left(2\pi\sigma_i^2\right)\right]\right\}, \tag{44}$$

où $\theta = (\omega_1,\mu_1,\mu_2,\sigma_1^2,\sigma_2^2)$ et $\omega_2 = 1-\omega_1$. En intégrant la vraisemblance complète $f(x,y;\theta)$ par rapport à la variable cachée $X$, la vraisemblance observée sous le modèle de $Y$ est retrouvée,

$$g(y;\theta) = \sum_{i=1}^{2} \omega_i \frac{1}{(2\pi\sigma_i^2)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2\sigma_i^2}(y-\mu_i)^2\right\}. \tag{45}$$

La loi a posteriori de la variable latente $X$ conditionnellement à $Y$ et sous un paramètre $\theta$ est

$$p(x=i;y,\theta) = \frac{f(x=i,y;\theta)}{g(y;\theta)} = \frac{\omega_i \frac{1}{(2\pi\sigma_i^2)^{\frac{1}{2}}} \exp\{-\frac{1}{2\sigma_i^2}(y-\mu_i)^2\}}{\sum_{i=1}^{2} \omega_i \frac{1}{(2\pi\sigma_i^2)^{\frac{1}{2}}} \exp\{-\frac{1}{2\sigma_i^2}(y-\mu_i)^2\}}.$$

La vraisemblance complète est un modèle exponentiel. En particulier, les six statistiques exhaustives ($i = 1$ ou $i = 2$) sont

$$\left.\begin{aligned} S_{1,i}(X,Y) &= \mathbb{1}(X \ \ = i), \\ S_{2,i}(X,Y) &= \mathbb{1}(X \ \ = i)Y, \\ S_{3,i}(X,Y) &= \mathbb{1}(X = i)Y^2. \end{aligned}\right\}$$

Les estimateurs de la quantité $\mathbb{E}_\pi[\mathbb{E}_{\theta'}[S(X,Y) \mid Y]]$ sont les suivants :

$$\left.\begin{array}{rcl} \bar{s}_{1,i}(y;\theta') = \mathbb{E}_{\theta'}[s_{1,i}(X,Y) \mid Y = y] & = & p(x = i; y,\theta'), \\ \bar{s}_{2,i}(y;\theta') = \mathbb{E}_{\theta'}[s_{2,i}(X,Y) \mid Y = y] & = & p(x = i; y,\theta')y, \\ \bar{s}_{3,i}(y;\theta') = \mathbb{E}_{\theta'}[s_{3,i}(X,Y) \mid Y = y] & = & p(x = i; y,\theta')y^2. \end{array}\right\}$$

La mise-à-jour des paramètres à l'étape de maximisation pour $\omega_i$, $\mu_i$ et $\sigma_i$ dans l'algorithme EM en ligne 2.4.1 de Cappé et Moulines (2009) est

$$\left.\begin{array}{rcl} \hat{\omega}_i & = & s_{i,1}, \\ \hat{\mu}_i & = & s_{i,2}/s_{i,1}, \\ \hat{\sigma}_i^2 & = & \frac{s_{i,3}}{s_{i,1}} - \left(\frac{s_{i,2}}{s_{i,1}}\right)^2. \end{array}\right\}$$

En effet, les quantités $s_{i,1}$, $s_{i,2}$ et $s_{i,3}$, où $i = \{1,2\}$, définies à l'aide de la récursion 34, sont des approximations stochastiques de $\mathbb{E}_\pi[\bar{s}_{i,1}(x,y)]$, $\mathbb{E}_\pi[\bar{s}_{i,2}(x,y)]$ et $\mathbb{E}_\pi[\bar{s}_{i,3}(x,y)]$, les espérances de $S_{i,1}(X,Y)$, $S_{i,2}(X,Y)$ et $S_{i,3}(X,Y)$, respectivement. À l'annexe A.3, les estimateurs de vraisemblance maximale sont écrits en fonction des statistiques exhaustives. Par exemple, l'estimateur du maximum de vraisemblance $\hat{\mu}_i$ avec $n = 1$ correspond à $\mathbb{1}(X = i)Y\mathbb{1}(X = i)$, ce qui correspond à $S_{i,2}(X,Y)/S_{i,3}(X,Y)$. Cette quantité est estimée par $\bar{s}_{2,i}(x,y)/\bar{s}_{1,i}(x,y)$ lors des étapes M. Dans les algorithmes en ligne, la dernière étape consiste à effectuer une approximation stochastique, d'où $s_{i,2}/s_{i,1}$. Le même raisonnement mène à l'étape de maximisation de $\sigma_i^2$ et $\omega_i$.

### 3.1.1. Spécification des algorithmes

*L'algorithme 2.4.1 dans le mélange de normales.* L'algorithme en ligne 2.4.1 est implémenté avec $\{\gamma_t\} = \frac{0,99}{t^{0,51}}$ et $s_0$ obtenu en inversant $\theta_0$ obtenu de la manière suivante. Un algorithme k-moyennes est appliqué sur 100 réalisations disponibles a priori afin de discriminer les catégories. Ensuite, on construit des estimations de $\mu_i$ $\sigma_i$ et $\omega_i$ à l'aide des estimateurs de vraisemblance maximale usuels conditionnellement aux classes de l'algorithme k-moyennes. Finalement, comme $\theta_0$ est fonction de $s_0$, on inverse $\theta_0$ pour exprimer $s_0$ en fonction de $\theta_0$. Pour les 1000 répétitions de l'expérience, l'algorithme traite $10^4$ observations séquentiellement. Afin de mieux illustrer la convergence de l'algorithme, ce nombre passe à $10^6$ pour les 5 graphiques de convergence des paramètres, uniquement. La technique de lissage de Polyak-Ruppert est appliquée à partir de l'itération 5001 ou $5 \times 10^5 + 1$ pour l'illustration de la convergence.

*L'algorithme 2.5.1 dans le mélange de normales.* L'algorithme 2.5.1 est implémenté de la même manière que l'algorithme 2.4.1 en ce qui a trait à l'initialisation de $s_0$ et $\{\gamma_t\}$ et la technique de Polyak-Ruppert. L'espérance $\mathbb{E}_{\bar{\theta}(s_{n-1})}[S(X_n,Y_n) \mid Y_n]$ est estimée à l'aide d'une espérance Monte-Carlo reposant sur $m = 10$ réalisations. Ce choix du nombre de particules

Monte-Carlo mène à une situation ou l'algorithme risque d'être plus variable que l'algorithme en ligne même si cet exemple est relativement simple dans le sens où l'estimateur Monte-Carlo n'est pas très variable. La complexité computationnelle est supérieure à l'algorithme 2.4.1 étant donné qu'on génère 10 réalisations de la variable latente $X$ conditionnellement à $Y$ à chaque itération afin de construire un estimateur Monte-Carlo plutôt que de calculer directement l'espérance a posteriori.

*L'algorithme 2.6.1 dans le mélange de normales.* L'algorithme 2.6.1 est implémenté de la même manière que les autres concernant $s_0$, $\{\gamma_t\}$ et la technique de Polyak-Ruppert. De plus, on spécifie $m = 100$ et $B = 50$. L'espérance $\mathbb{E}_{\bar{\theta}(s_{n-1})}\left[S(X_n,Y_n) \mid Y_n\right]$ est estimée à l'aide d'une espérance MCMC reposant sur $m - B = 50$ éléments de la chaîne de Markov. La chaîne de Markov générée par l'algorithme de Metropolis-Hastings à l'itération $k$ est telle que son noyau est $K \sim \mathcal{N}(x_k; 9)$ et sa loi stationnaire est $p(x_k; y_k,\hat{\theta}_k)$. La raison pour laquelle on prend 50 réalisations plutôt que 10, comme on avait fait pour l'espérance Monte-Carlo, est de limiter le biais de l'estimateur MCMC. La différence du nombre de réalisations est sans conséquence, car on cherche pas à comparer les estimateurs des algorithmes en ligne Monte-Carlo et MCMC entre eux. On comparera plutôt ceux-ci à ceux de l'algorithme EM en ligne.

**Remarque 3.1.1.** *Dans l'algorithme 2.6.1, on combine effectivement deux algorithmes, l'algorithme de Robbins-Monro et l'algorithme MCMC, afin d'effectuer la tâche qu'accomplit l'algorithme EM lorsque toutes les données sont disponibles. Cela signifie que lorsqu'on dispose d'un jeu de données duquel on souhaite extraire des estimateurs de vraisemblance maximale, il faut utiliser l'algorithme de Metropolis-Hastings à chaque itération de l'algorithme EM en ligne plutôt que de calculer directement l'espérance a posteriori de la variable latente. Par conséquent, même si l'algorithme est compétitif au niveau de l'efficacité relative, ce qui est le cas quand m est assez grand, sa complexité computationnelle est supérieure. Cependant, le recours au MCMC est parfois nécessaire lorsque l'espérance a posteriori ne peut être ni calculée explicitement, ni estimée par Monte-Carlo, ce qui est une situation très fréquente en pratique.*

### 3.1.2. Convergence des estimateurs

Tel que mentionné en début de chapitre, on prend 5 répétitions de l'expérience (qui, on le rappelle, disposent exceptionnellement chacune ici de $10^6$ de données plutôt que $10^4$ pour ces graphiques) pour illustrer la convergence des estimateurs des paramètres vers les points minimisant la divergence de Kullback-Leibler entre $\pi$ et $g_\theta$, divergence qu'on a définie à l'équation 33. Les algorithmes 2.4.1, 2.5.1 et 2.6.1 sont confrontés aux mêmes données afin de comparer leur performance. Lorsqu'on passe de l'algorithme 2.4.1 en ligne à l'algorithme 2.5.1 en ligne Monte-Carlo, il y a une « couche » aléatoire supplémentaire puisque l'espérance

doit être estimée à l'aide de la méthode Monte-Carlo. Lorsque ceci n'est pas envisageable, on doit utiliser une méthode MCMC, ce qui ajoute de la variabilité puisque l'estimateur MCMC peut avoir une grande variance en plus d'être biaisé. Les algorithmes mettant à l'oeuvre des méthodes Monte-Carlo et MCMC semblent converger à la même vitesse que l'algorithme 2.4.1 (figure 3.1), ce qu'on observe aux figures 3.2 et 3.3, respectivement, et plus généralement à la figure 3.4. Toutefois, il y a des « retards » lorsqu'on passe aux versions MC et MCMC. En effet, la figure 3.4 montre que la distance moyenne entre $\mu_2$ et son estimateur courant est constamment plus petite pour l'estimateur EM en ligne que pour les autres.

### 3.1.3. Distribution des estimateurs

Les graphiques de densité empirique des estimateurs de chacun des paramètres à estimer permettent de visualiser facilement la distribution de l'estimateur. Pour étudier rigoureusement la normalité asymptotique, il faudrait utiliser la fonction de répartition, ce qu'on ne fait pas puisque ce n'est pas l'une des visées de ce mémoire. Les graphiques de densité peuvent être observés à la figure 3.5. Pour les estimateurs provenant des algorithmes 2.5.1 et 2.6.1, les estimateurs sont quelque peu plus variables que les estimateurs provenant de l'algorithme 2.4.1 comme le témoigne la hauteur des modes et la lourdeur des queues. C'est une conséquence de l'estimation supplémentaire au niveau de l'étape E d'espérance. Les densités empiriques des estimateurs trouvés par l'algorithme 2.5.1 et 2.6.1 sont également d'apparence normale. Dans Cappé et Moulines (2009), il est montré que l'algorithme EM en ligne 2.4.1 converge asymptotiquement vers une distribution normale lorsqu'on applique la moyenne de Polyak-Ruppert. Dans cet exemple, plutôt que d'imposer une tolérance comme critère d'arrêt, on limite le nombre d'itérations à $10^4$. Compte tenu de cette limite, il est probable que certaines expériences n'aient pas entièrement convergé. Finalement, l'algorithme EM trouve des estimateurs généralement peu variables et peu biaisés après 50 itérations, car la convergence est établie comme le montre la figure 3.6. Dans cet exemple, l'algorithme EM est efficace computationnellement, car il converge après un petit nombre d'itérations. L'algorithme EM pouvait être implémenté étant donné que l'espérance a posteriori était calculable. Si ce n'était pas possible, il faudrait approximer chacune des $n$ espérances à l'aide de Monte-Carlo ou de l'algorithme de Metropolis-Hastings à chaque itération. L'algorithme EM en ligne par Monte-Carlo ou MCMC implique un seul usage de l'algorithme de Metropolis-Hastings ou de Monte-Carlo par itération, ce qui est bien moins lourd en temps computationnel. Il y a effectivement $n$ espérances a posteriori à approcher à chaque itération dans la version EM de Dempster *et al.* (1977) avec un estimateur MCMC contre une seule espérance a posteriori par itération pour l'EM en ligne MCMC.

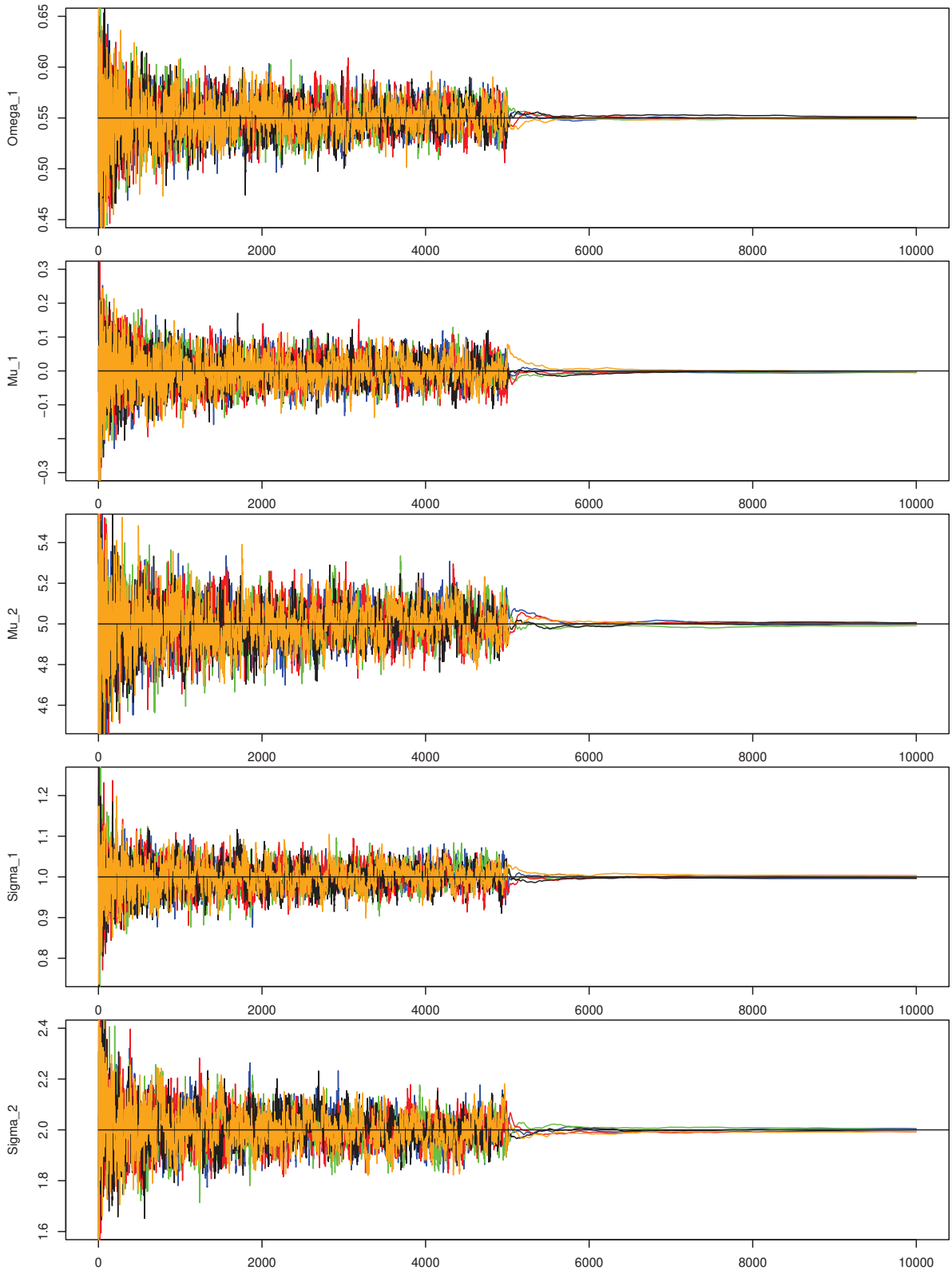


**Figure 3.1.** Illustration de la convergence des estimateurs de $\omega_1$, $\mu_1$, $\mu_2$, $\sigma_1$, $\sigma_2$ de l'algorithme en ligne 2.4.1 pour le modèle de mélange de normales. Chaque couleur correspond à un jeu de données. Le trait noir correspond à la vraie valeur du paramètre.

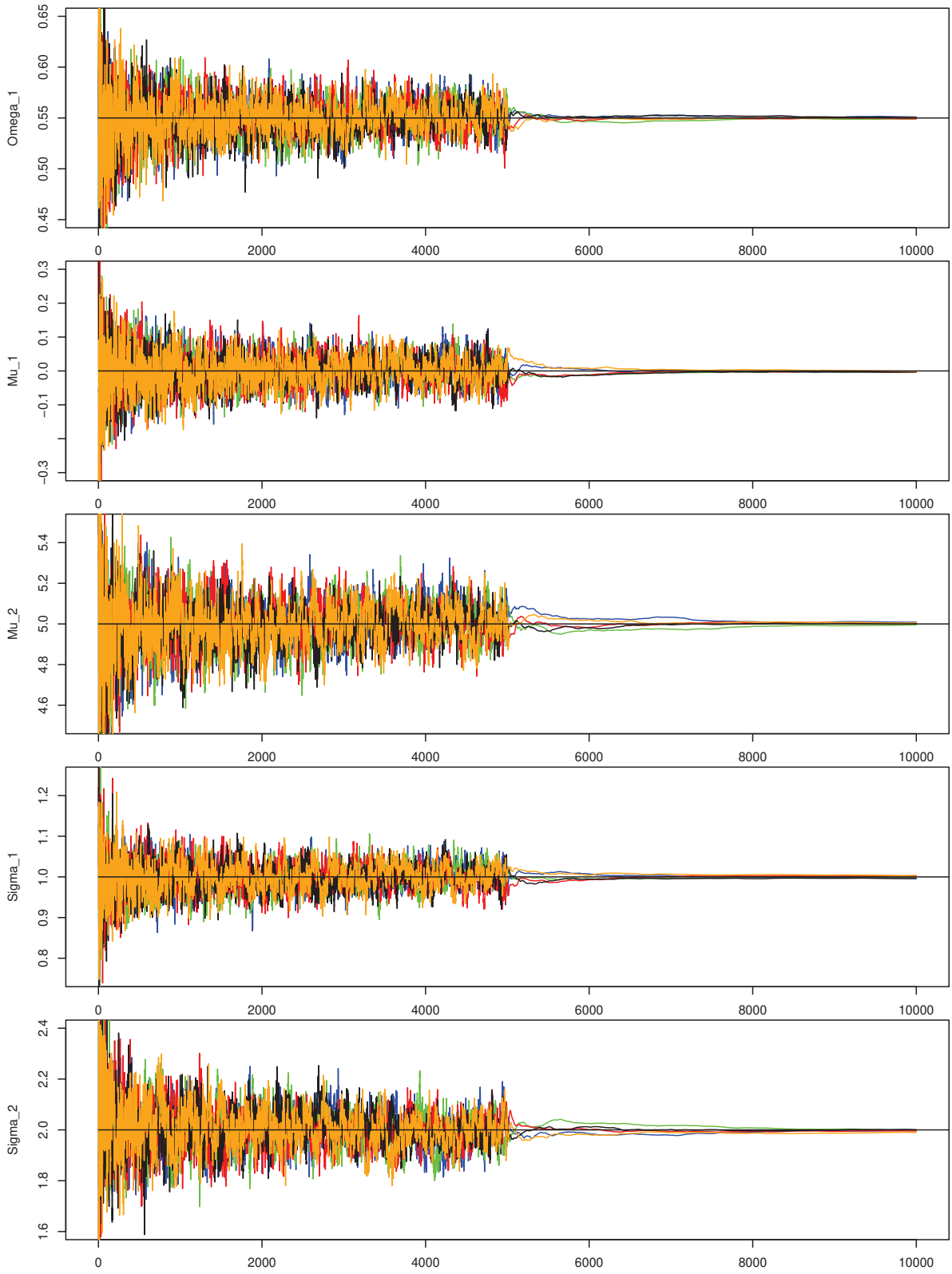


**Figure 3.2.** Illustration de la convergence des estimateurs de $\omega_1$, $\mu_1$, $\mu_2$, $\sigma_1$, $\sigma_2$ de l'algorithme en ligne Monte-Carlo 2.5.1 avec $m = 10$ réalisations. Chaque couleur correspond à un jeu de données différent. Le rythme de la convergence est similaire à la figure 3.1.

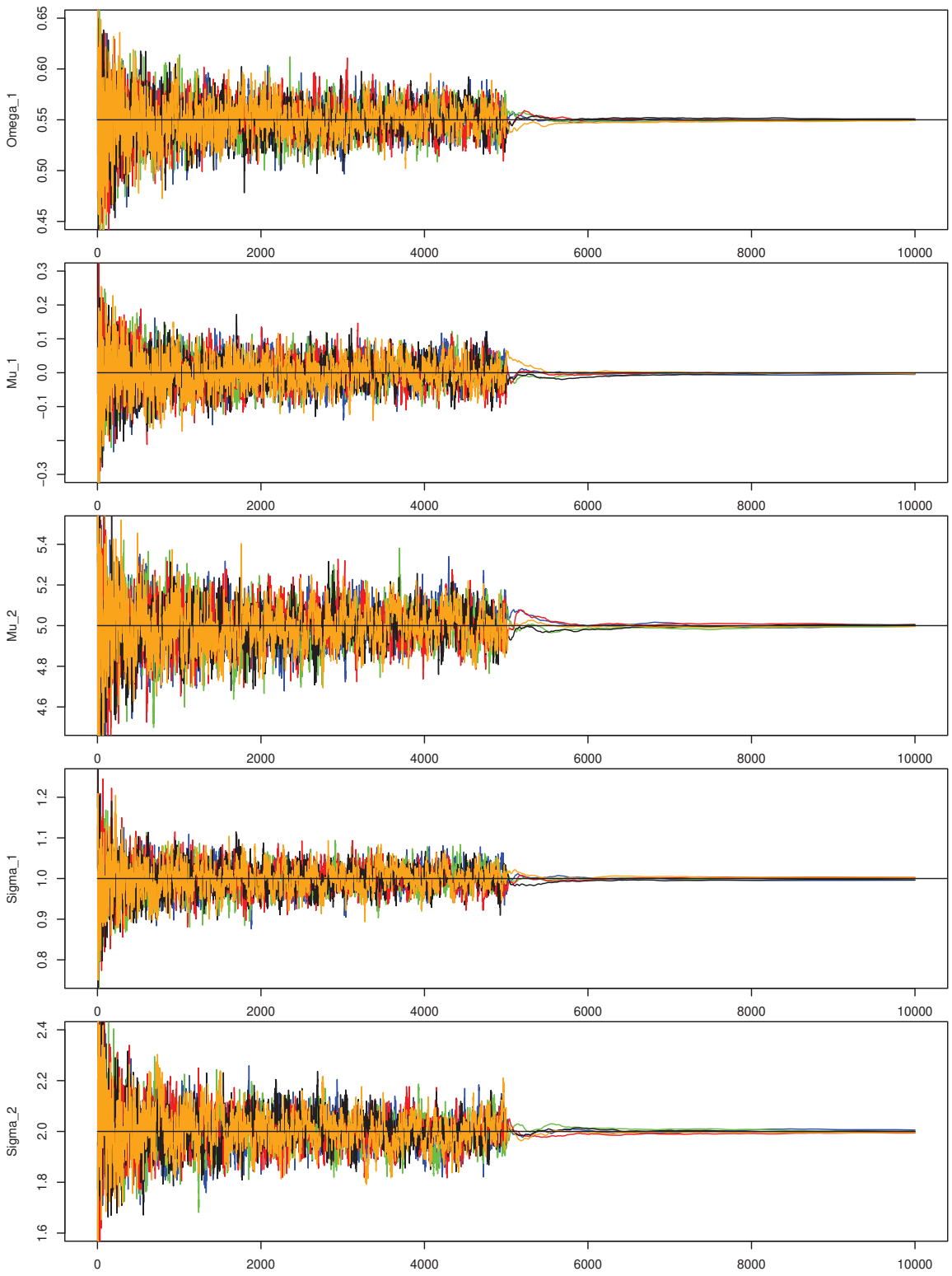


**Figure 3.3.** Illustraton de la convergence des estimateurs de $\omega_1$, $\mu_1$, $\mu_2$, $\sigma_1$, $\sigma_2$ de l'algorithme en ligne MCMC 2.6.1 avec $m - B = 50$ réalisations. Chaque couleur correspond à un jeu de données.

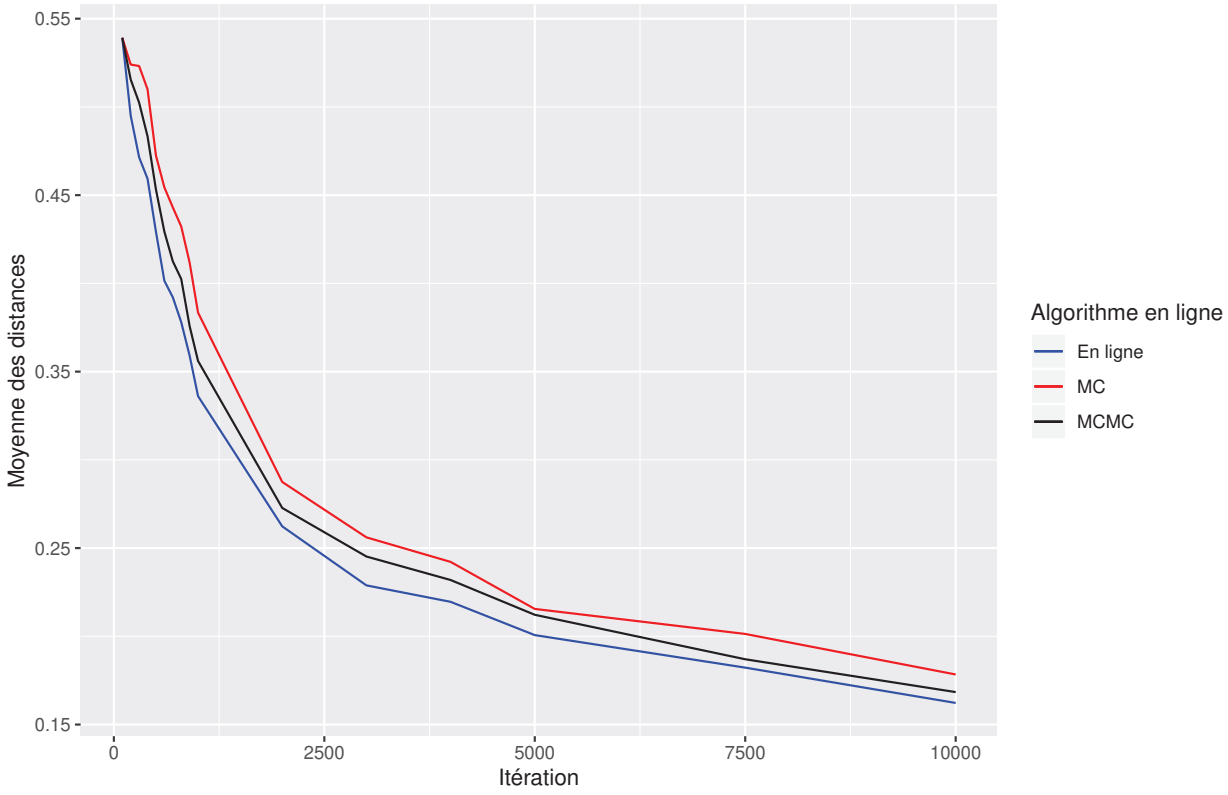


**Figure 3.4.** La distance moyenne entre $\mu_2$ et la valeur de l'estimateur courant de $\mu_2$. La convergence s'effectue au même rythme. Cependant, en moyenne, les estimateurs de l'algorithme en ligne sont plus près de la vraie valeur du paramètre que les estimateurs des algorithmes en ligne Monte-Carlo et MCMC. Il est difficile de percevoir quel algorithme produit des estimateurs plus éloignés de la vraie valeur du paramètre en observant les figures 3.1 à 3.3. Le rythme de convergence est très similaire entre l'EM en ligne et les versions Monte-Carlo et MCMC. Notons que la version MCMC a une distance moyenne plus faible que la version Monte-Carlo puisqu'elle repose sur 50 réalisations au lieu de 10, comme c'est le cas pour la version Monte-Carlo. On ne prend pas 50 réalisations MC, car sa variance n'est pas beaucoup plus grande que la version en ligne et on voudrait ultimement voir s'il est possible de réduire sa variance. De toute manière, on ne cherche pas à comparer les variances des algorithmes EM en ligne MC et MCMC.

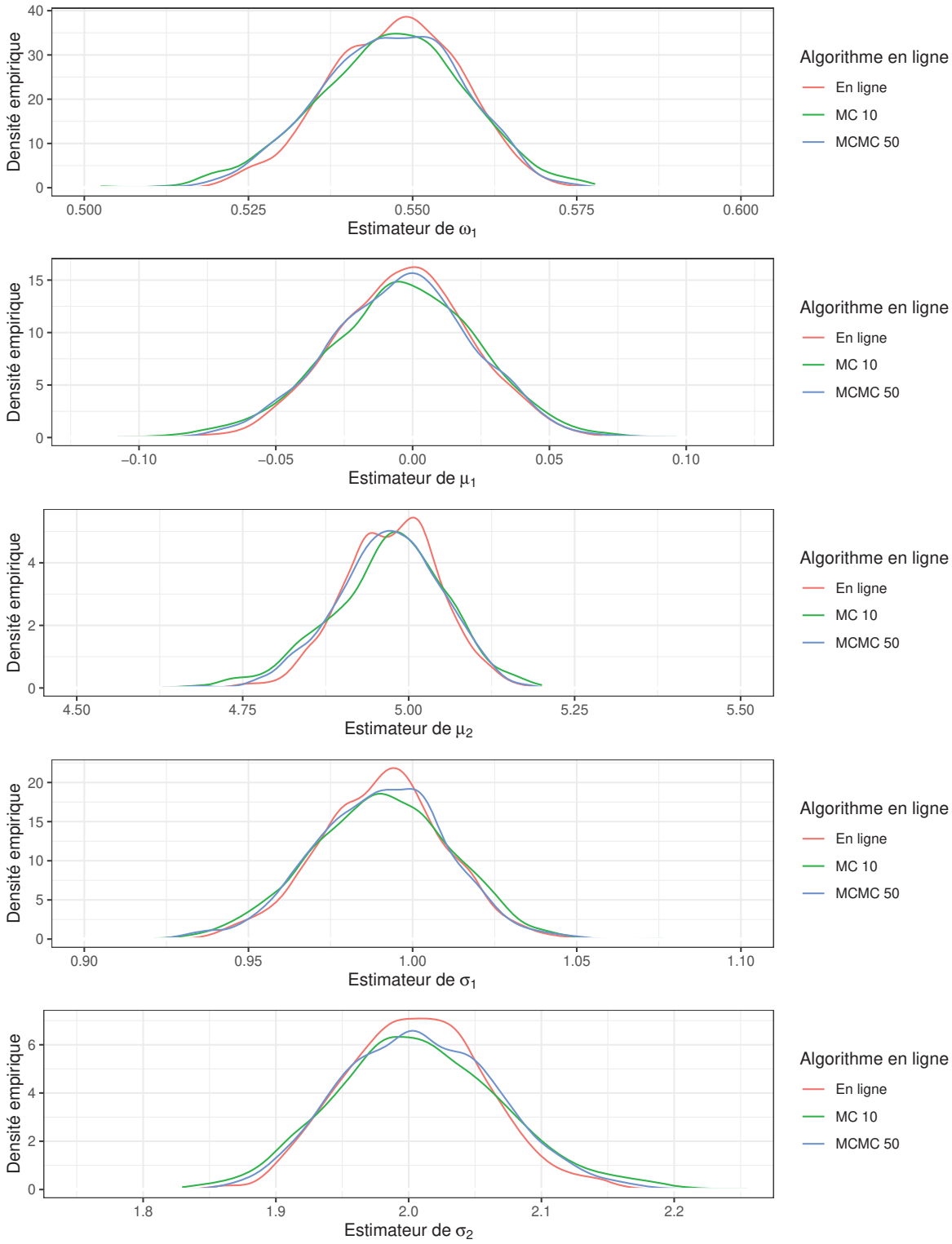


**Figure 3.5.** La densité empirique des estimateurs pour un mélange de normales est illustrée pour les algorithmes en ligne, en ligne MC 10 et en ligne MCMC 50. La normalité asymptotique n'est pas encore atteinte après $10^4$ itérations, mais la forme ne s'en éloigne pas foncièrement. L'augmentation de variance se répercute sur la hauteur du mode et la lourdeur des queues de la distribution des estimateurs provenant des algorithmes EM en ligne Monte-Carlo et MCMC.

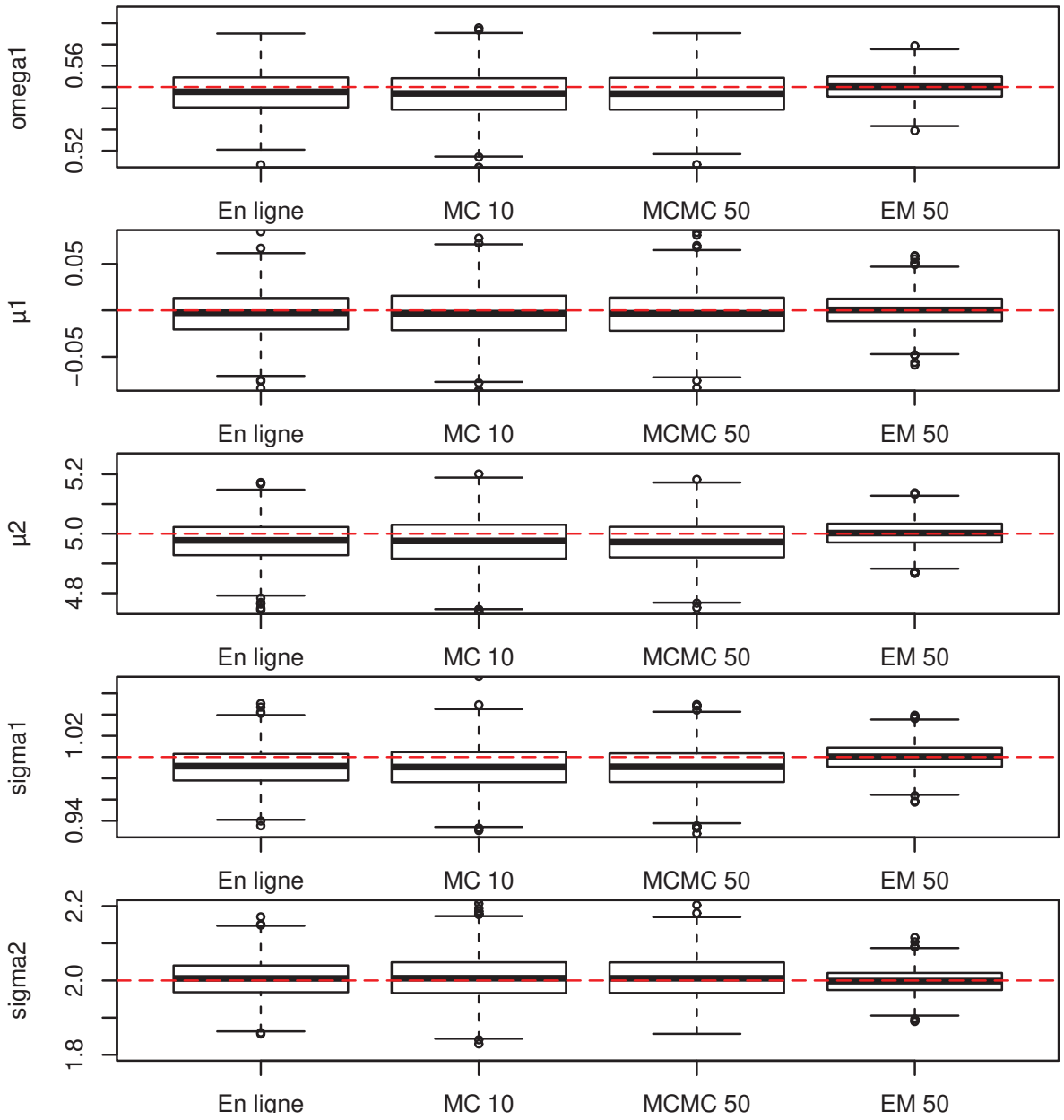


**Figure 3.6.** Boîtes à moustaches pour les paramètres $\omega_1$, $\mu_1$, $\mu_2$, $\sigma_1^2$, $\sigma_2^2$ après 1000 répétitions de $10^4$ itérations des algorithmes 2.4.1, 2.5.1, 2.6.1 et 50 itérations de l'algorithme EM dans le modèle de mélange de normales. On peut observer un léger biais chez les estimateurs des algorithmes en ligne. L'algorithme EM est limité à 50 itérations. Cela signifie qu'il a vu 50 fois toutes les données alors que les algorithmes en ligne ne les ont vues qu'une seule fois. L'écart interquartile montre que la variabilité des estimateurs augmente légèrement en passant aux versions Monte-Carlo et MCMC de l'algorithme EM en ligne.

## 3.2. Exemple d'un mélange de régression

Soit

$$Y = \mathbb{1}(X = 1)(10U - U^2) + [1 - \mathbb{1}(X \quad = 1)]\,(15 + 5U) + \epsilon,$$

la variable aléatoire suivie par les données où $U \sim$ Uniforme(0,10) est connue, $X \sim$ Bernoulli($\omega_1$) est une variable latente et $\epsilon \sim \mathcal{N}(0,\sigma^2 = 9)$. On modélise le processus par un couple $(X,Y)$. Soit $\mathbf{x}^T = [1,U,U^2]$, le modèle est le suivant :

$$Y = \sum_{i=1}^{2} \mathbb{1}(X \quad = i)\left[\beta_0^i + \beta_1^i U + \beta_2^i U^2\right] + \epsilon = \sum_{i=1}^{2} \mathbb{1}(X = i)\left[\mathbf{x}^T \beta_i\right] + \epsilon.$$

On définit $X$, la variable latente qui indique l'appartenance à une classe (1 ou 2). Les covariables $U$ et $U^2$ sont supposées connues. La densité jointe du modèle des données complètes s'écrit ainsi :

$$f(x,y;\theta) = \prod_{i=1}^{2} \omega_i^{\mathbb{1}(x=i)} \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2\sigma^2}\mathbb{1}(x \quad = i)(y - \mathbf{x}^T\beta_i)^2\right\},$$

où $\omega_2 = 1 - \omega_1$. On peut réécrire la densité jointe sous la forme standard des distributions appartenant à la famille exponentielle,

$$f(x,y;\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\sum_{i=1}^{2}\left(\frac{1}{2\sigma^2}\mathbb{1}(x \quad = i)(y - \mathbf{x}^T\beta_i)^2 - \mathbb{1}(x = i)\log(\omega_i)\right)\right\}.$$

La densité observée s'écrit de la façon suivante :

$$g(y;\theta) = \sum_{i=1}^{2} \omega_i f(y; x = i,\theta) = \sum_{i=1}^{2} \omega_i \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2\sigma^2}(y - \mathbf{x}^T\beta_i)^2\right\}.$$

La fonction de densité a posteriori de la variable latente est

$$p(x;\theta,y) = \frac{f(x,y;\theta)}{g(y;\theta)} = \frac{\prod_{i=1}^{2} \omega_i^{\mathbb{1}(x=i)} \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\{-\frac{1}{2\sigma^2}\mathbb{1}(x \quad = i)(y - \mathbf{x}^T\beta_i)^2\}}{\sum_{i=1}^{2} \omega_i \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\{-\frac{1}{2\sigma^2}(y - \mathbf{x}^T\beta_i)^2\}}.$$

La log-vraisemblance s'écrit comme

$$\log\left(f(x,y;\theta)\right) = -\log(2\pi\sigma^2) + \sum_{i=1}^{2} \mathbb{1}(x \quad = i)\left(\log\omega_i - \frac{1}{2\sigma^2}\left(y^2 + (\mathbf{x}^T\beta_i)^2 - 2y\mathbf{x}^T\beta_i\right)\right).$$

De là, on extrait les statistiques exhaustives,

$$\left.\begin{array}{rcl} S_{1,i}(X,Y) & = & \mathbb{1}(X = i) \quad (1 \times 1), \\ S_{2,i}(X,Y) & = & \mathbb{1}(X \quad = i)\mathbf{x}^T\mathbf{x} \quad (3 \times 3) \\ S_{3,i}(X,Y) & = & \mathbb{1}(X = i)\mathbf{x}^T Y \quad (3 \times 1). \end{array}\right\}$$

Les estimateurs de la quantité $\mathbb{E}_\pi[\mathbb{E}_{\theta'}[S(X,Y) \mid Y]]$ sont les suivants :

$$\left.\begin{array}{lllll} \bar{s}_{1,i}(y;\theta') & = \mathbb{E}_{\theta'}[\mathbb{1}(X=i) \mid Y=y] & = p(i;\theta',y) & (1\times 1), \\ \bar{s}_{2,i}(y;\theta') & = \mathbb{E}_{\theta'}[\mathbb{1}(X=i) \mid Y=y]\mathbf{x}^T\mathbf{x} & = p(i;\theta',y)\mathbf{x}^T\mathbf{x} & (3\times 3), \\ \bar{s}_{3,i}(y;\theta') & = \mathbb{E}_{\theta}\mathbb{1}[(X=i) \mid Y=y]\mathbf{x}^T y & = p(i;\theta',y)\mathbf{x}^T y & (3\times 1). \end{array}\right\}$$

La mise-à-jour des paramètres à l'étape de maximisation pour $\omega_i$ et $\beta_i$ dans l'algorithme EM en ligne sans approximation sont

$$\left.\begin{array}{rcl} \hat{\beta}_i & = & (s_{i,2})^{-1}s_{i,3}, \\ \hat{\omega}_i & = & s_{i,1}. \end{array}\right\}$$

En effet, $s_{i,1}$, $s_{i,2}$ et $s_{i,3}$ sont des approximations stochastiques des statistiques $\bar{s}_{i,1}(x,y)$, $\bar{s}_{i,2}(x,y)$ et $\bar{s}_{i,3}(x,y)$, les espérances a posteriori de $S_{i,1}(X,Y)$, $S_{i,2}(X,Y)$ et $S_{i,3}(X,Y)$, respectivement. À l'annexe A.3, les estimateurs de vraisemblance maximale sont écrits en fonction des statistiques exhaustives. L'estimateur du maximum de vraisemblance de $\beta_i$ avec $n=1$ est $\mathbb{1}[(X=i)\mathbf{x}^T\mathbf{x}]^{-1}\mathbb{1}(X=i)\mathbf{x}^T Y$, ce qui correspond à $(S_{i,2}(X,Y))^{-1}S_{i,3}(X,Y)$. Intuitivement, cette quantité est estimée par $(\bar{s}_{2,i}(x,y))^{-1}\bar{s}_{3,i}(x,y)$ lors de l'étape M, car $X$ n'est pas observée. Dans les algorithmes en ligne, la dernière étape consiste à effectuer une approximation stochastique, d'où $(s_{i,2})^{-1}s_{i,3}$.

### 3.2.1. Spécification des algorithmes

Les algorithmes 2.4.1 sont implémentés avec les mêmes paramètres $B$, $m$, $N$ qu'au premier exemple. Le pas d'apprentissage $\{\gamma_t\}$ vaut $\frac{0{,}99}{t^{0{,}51}}$. De plus, la quantité $s_0$ est initialisée à l'aide de statistiques exhaustives extraites de 100 données complètes qu'on aurait observées a priori. L'algorithme traite $10^4$ observations séquentiellement. La moyenne de Polyak-Ruppert est appliquée à partir de l'itération 5001 ou $5\times 10^4+1$ pour les illustrations de convergence.

### 3.2.2. Convergence des estimateurs

Tel que mentionné en début de chapitre, on prend 5 répétitions de l'expérience pour illustrer la convergence des estimateurs des paramètres vers le minimum de la divergence de Kullback-Leibler entre $\pi$ et $g_\theta$. L'algorithme est exceptionnellement implémenté pour $10^5$ itérations au lieu de $10^4$ itérations afin de mieux observer la convergence. On ne choisit pas $10^6$ itérations pour l'illustration des cinq courbes de convergence comme à l'exemple du mélange de normales puisque l'aspect computationnel devenait trop laborieux. La convergence s'effectue vers la vraie valeur des paramètres pour l'algorithme 2.4.1, comme on peut le voir à la figure 3.7 et l'algorithme 2.6.1, comme on le constate à la figure 3.9. Cependant, il arrive à l'algorithme EM en ligne Monte-Carlo 2.5.1 de converger vers d'autres valeurs, comme on le constate dans l'illustration de la convergence pour l'algorithme EM en ligne Monte-Carlo (courbe verte dans la figure 3.8). Le niveau supplémentaire d'aléatoire peut donc engendrer

des estimateurs très différents de ceux qui étaient construits par l'algorithme en ligne. Il se pourrait que l'hypothèse 3i de la section 2.4.1, $(1-\gamma)s + \gamma\bar{s}(y;\theta) \in \mathcal{S}$, ne tienne plus lorsqu'on remplace l'espérance a posteriori par un estimateur, ce qui pourrait expliquer la trajectoire de la courbe verte. Cependant, c'est une spéculation et non une certitude.

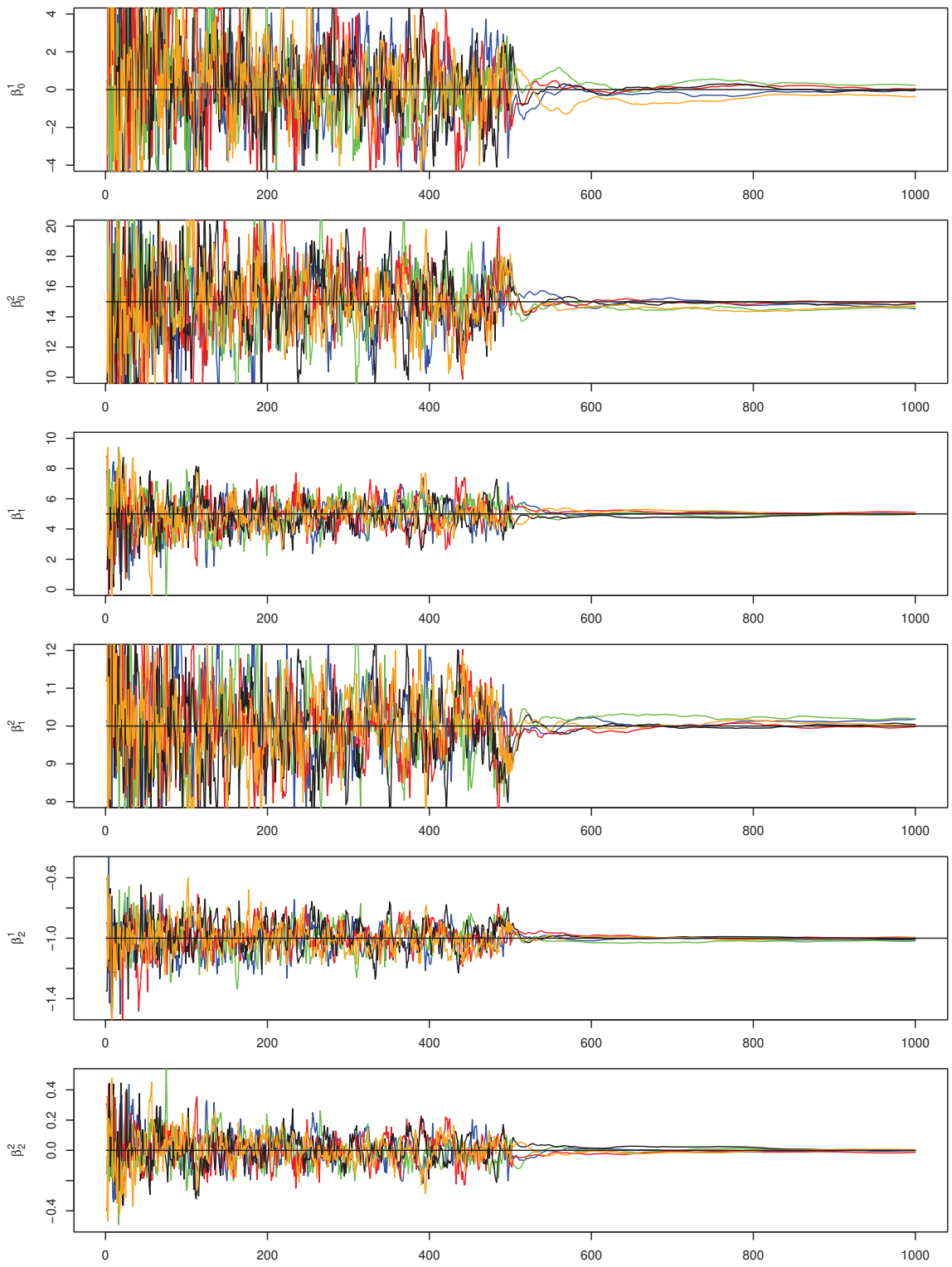


**Figure 3.7.** Illustration de la convergence des estimateurs de $\beta_0^1$, $\beta_0^2$, $\beta_1^1$, $\beta_1^2$, $\beta_2^1$, $\beta_2^2$ dans le cas de l'algorithme en ligne 2.4.1 pour l'exemple du mélange de modèles de régression. Chaque couleur représente un jeu de données différent. Le trait noir correspond à la valeur du paramètre.

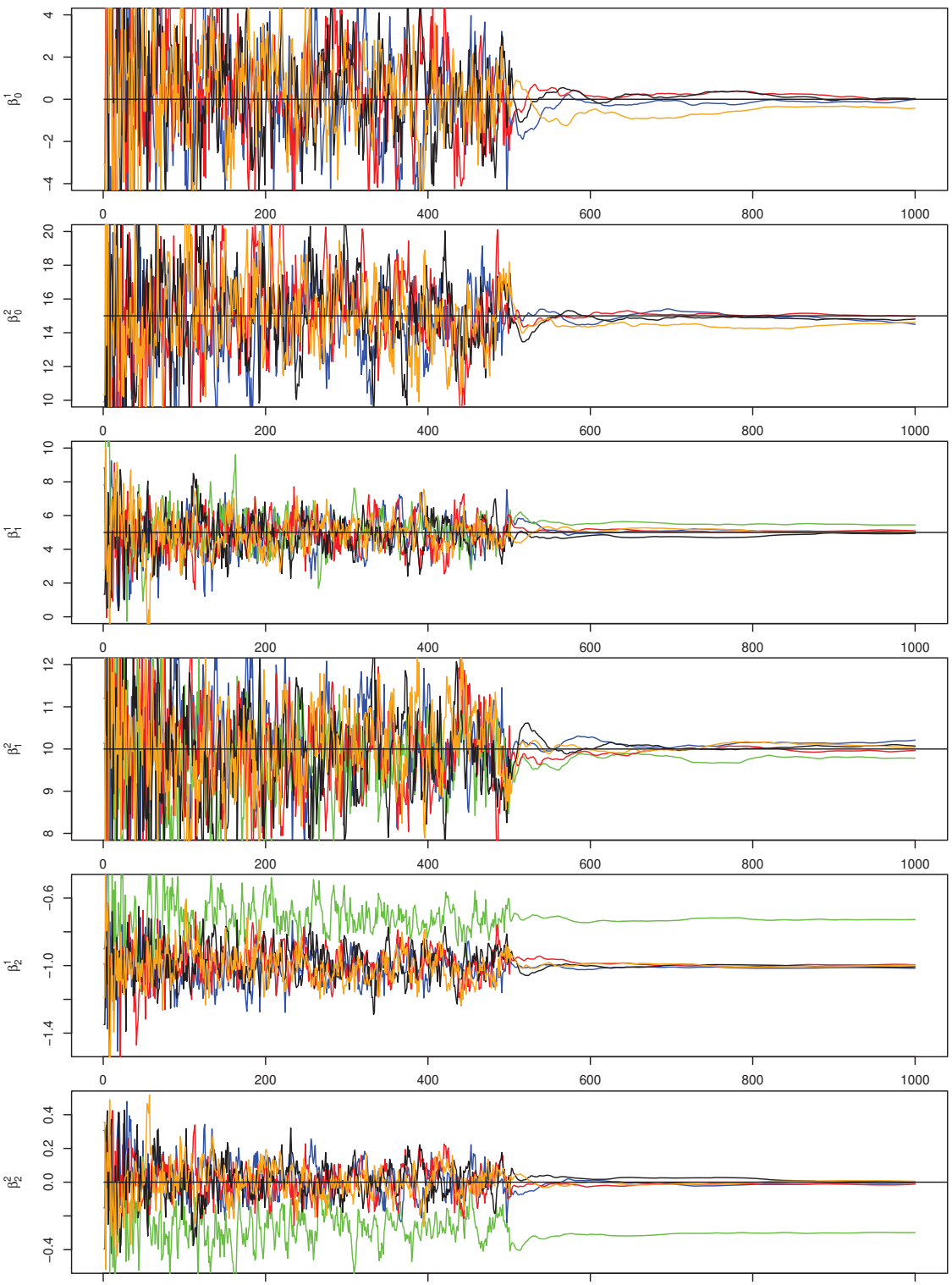


**Figure 3.8.** Illustration de la convergence des estimateurs de $\beta_0^1$, $\beta_0^2$, $\beta_1^1$, $\beta_1^2$, $\beta_2^1$, $\beta_2^2$ dans le cas de l'algorithme 2.5.1 pour l'exemple de mélange de modèles de régression. L'algorithme EM en ligne Monte-Carlo ne construit plus les mêmes estimateurs que l'algorithme EM en ligne pour le jeu de données associé à la couleur verte. Les estimateurs ne sont pas près des vrais paramètres. On spécule que dans ce cas-là, la condition $s \in \mathcal{S}$ pourrait ne plus être respectée.

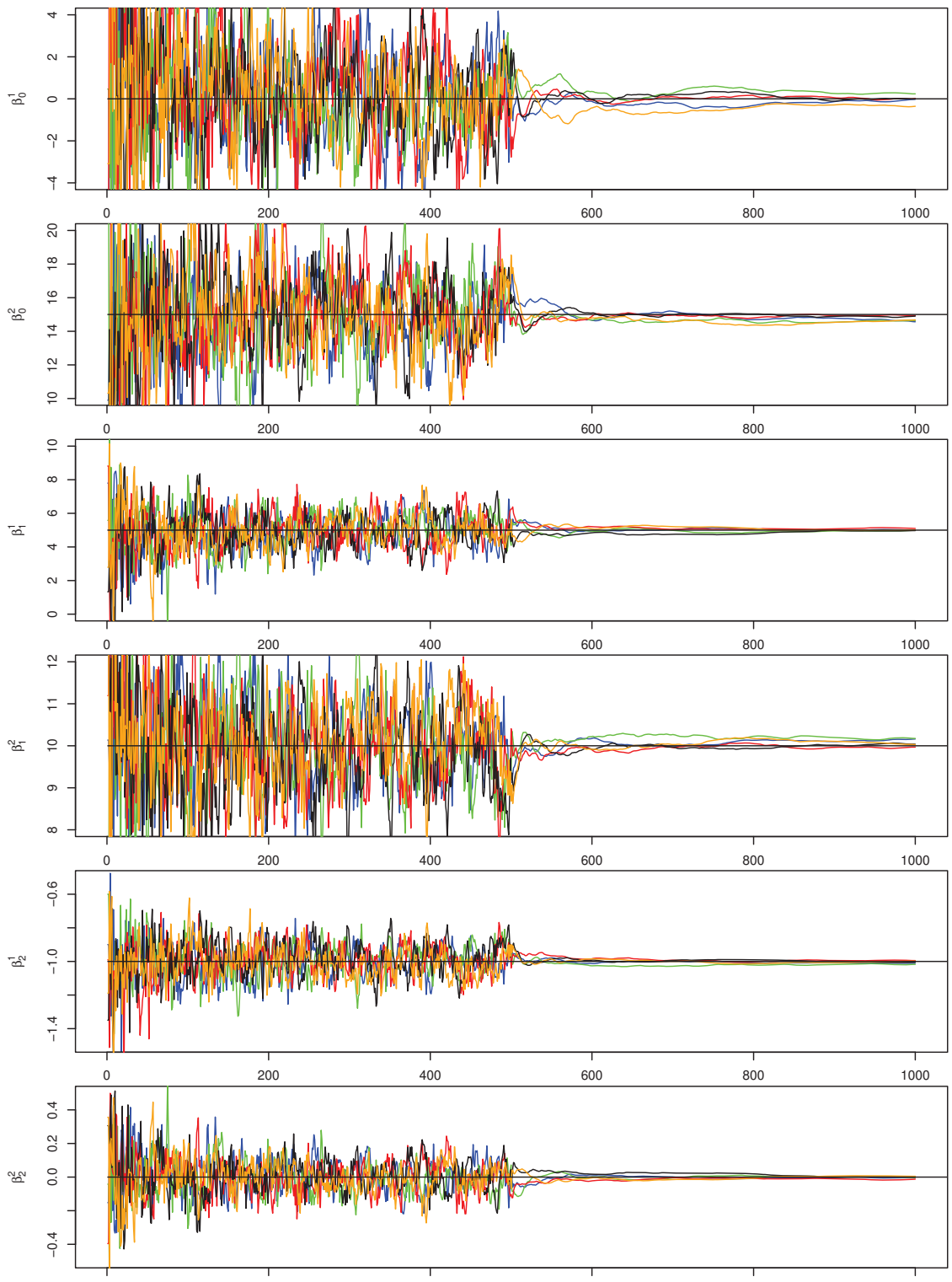


**Figure 3.9.** Illustration de la convergence des estimateurs de $\beta_0^1$, $\beta_0^2$, $\beta_1^1$, $\beta_1^2$, $\beta_2^1$, $\beta_2^2$ dans le cas de l'algorithme EM en ligne MCMC 2.6.1 pour l'exemple de mélange de modèles de régression. Contrairement à la figure 3.8, la trajectoire correspondant au jeu de données de la courbe verte converge vers les vrais paramètres.

### 3.2.3. Distribution des estimateurs

On constate en observant les densités asymptotiques que la normalité est qualitativement atteinte après $10^4$ itérations dans le problème de mélange de régression aux figures 3.10 et 3.11. Pour étudier rigoureusement la normalité asymptotique, il faut plutôt regarder la fonction de répartition. Les algorithmes contenant des estimateurs Monte-Carlo et MCMC convergent quelquefois vers des points différents que l'algorithme EM en ligne. Pour la plupart des estimateurs, le mode de la densité empirique de l'algorithme 2.4.1 est plus élevé, ce qui indique que les estimateurs sont plus concentrés autour de ce mode. Il y a un exemple de jeu de données pour lequel l'algorithme 2.4.1 et l'algorithme 2.6.1 atteignent un estimateur très près de la vraie valeur du paramètre, tandis que l'algorithme 2.5.1 converge vers un autre point critique comme le témoigne la figure 3.9. En somme, la variance augmente en utilisant des estimateurs de type Monte-Carlo et MCMC. Ceci est une conséquence de la couche supplémentaire de bruit engendrée par l'estimation de l'espérance a posteriori avec uniquement 10 réalisations Monte-Carlo ou 50 réalisations MCMC. Des boîtes à moustaches sont exhibés à la figure 3.12 dans le but de comparer la stabilité des différents algorithmes. Les écarts interquartiles augmentent en passant de l'algorithme EM en ligne à l'algorithme EM en ligne Monte-Carlo ou MCMC. Finalement, on constate que pour 50 itérations de l'EM, l'algorithme EM a convergé vers l'estimateur du maximum de vraisemblance.

## 3.3. Comparaison de l'efficacité des différents algorithmes dans des modèles à variables latentes discrètes

Dans les exemples de mélange présentés jusqu'à présent, les statistiques exhaustives d'un mélange impliquent des Bernoulli qui indiquent l'appartenance à la classe. L'espérance a posteriori de celles-ci correspond simplement à la probabilité a posteriori de $X$ conditionnellement à la plus récente mise-à-jour du paramètre. Si la densité a posteriori de $X$ est connue et simple, l'étape E est entièrement faisable. L'utilisation de méthodes Monte-Carlo et MCMC dans cet exemple permet d'évaluer l'impact de la méconnaissance de l'espérance a posteriori ; elles ne sont pas strictement nécessaires contrairement à des situations où l'espérance d'une statistique exhaustive est un problème ardu ou quand la densité est connue à une constante de normalisation près.

Comme il a été établi dans la section 1.4, la moyenne arithmétique de $m$ variables aléatoires iid suivant une loi particulière (une Bernoulli, par exemple) converge presque sûrement vers l'espérance de cette même loi. La performance de l'algorithme 2.5.1 dépend du choix de $m$. Lorsque $m$ est petit, l'algorithme 2.5.1 est moins efficace que l'algorithme 2.4.1. Il en est de même avec l'algorithme 2.6.1. Lorsque $m$ est grand, le rapport des variances s'approche de 1 puisque ces méthodes convergent vers la vraie valeur de la probabilité a posteriori à chaque itération de l'algorithme EM en ligne. En choisissant des normales à faibles variances avec deux modes éloignés, l'algorithme classe aisément la plupart des observations et les estimateurs convergent vers les vraies valeurs des paramètres.

On constate aux tableaux 3.1 et 3.2 que la variance de l'estimateur provenant de l'algorithme 2.4.1 est inférieure à celles des algorithmes EM en ligne contenant les estimations d'espérances a posteriori pour chacun des exemples de mélange. Il y a quelques cas où les algorithmes 2.5.1 et 2.6.1 ne convergent pas vers les vraies valeurs dans le mélange de régression. Cela se produit puisque les algorithmes 2.5.1 et 2.6.1 convergent parfois vers des valeurs qui ne sont pas dans le voisinage de la vraie valeur contrairement à l'algorithme 2.4.1. En retenant une statistique plus robuste comme les écarts absolus médians, le portrait demeure sensiblement le même, bien que le rapport des écarts absolus médians au carré soit plus élevé que le rapport de variance. L'algorithme 2.4.1 possède donc une meilleure efficacité relative que les autres, tel qu'attendu.

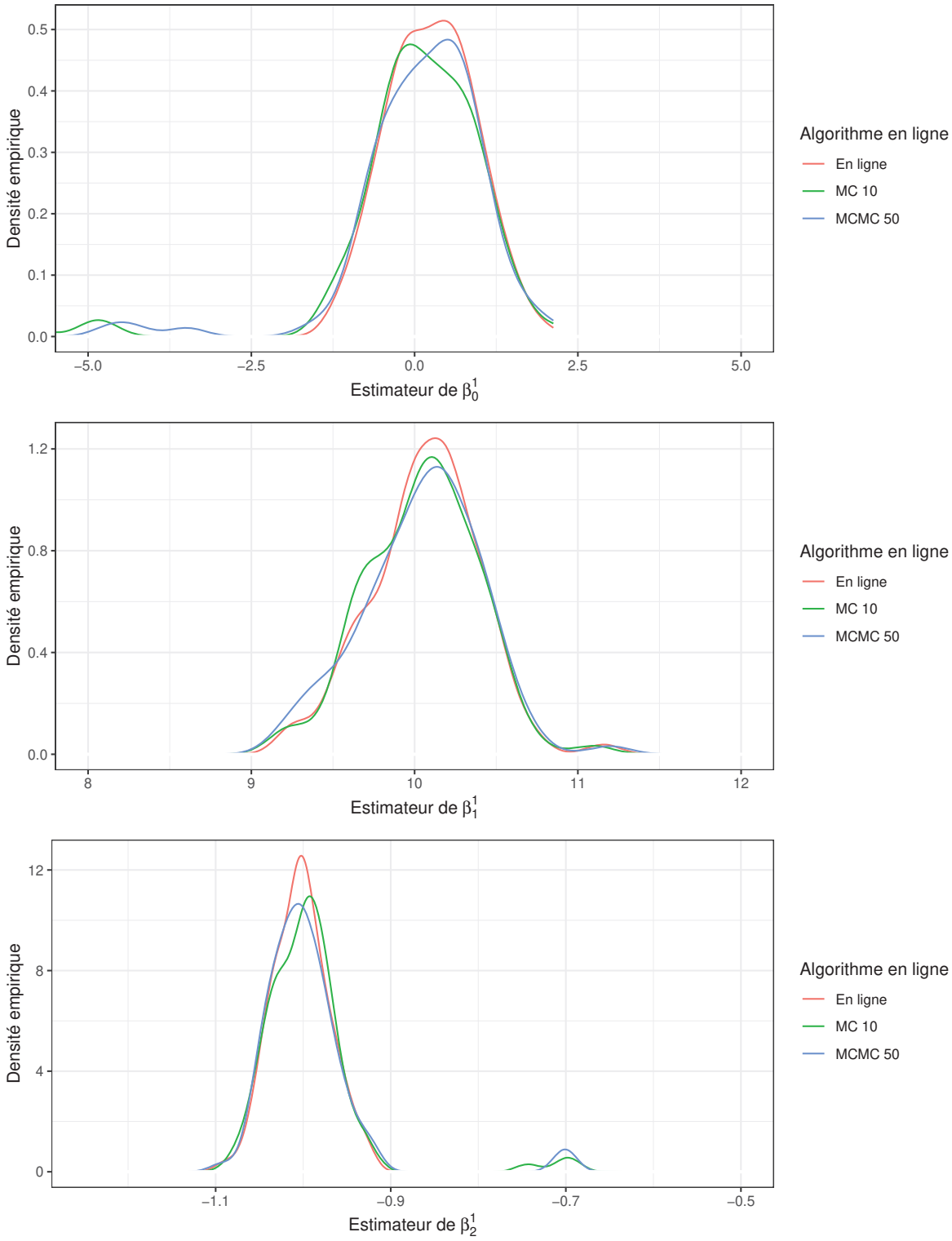


**Figure 3.10.** Densité empirique des estimateurs de $\beta_0^1$, $\beta_1^1$, $\beta_2^1$ pour l'algorithme EM en ligne, EM en ligne Monte-Carlo et EM en ligne MCMC pour l'exemple du modèle de mélange de régression. L'algorithme EM en ligne paraît un peu moins variable lorsqu'on regarde la hauteur des modes et la lourdeur des queues. De plus, il y a une chance non négligeable que les estimateurs construits par les algorithmes EM en ligne Monte-Carlo et MCMC soient loin de l'estimateur du maximum de vraisemblance, ce qui n'est pas le cas avec l'algorithme EM en ligne.

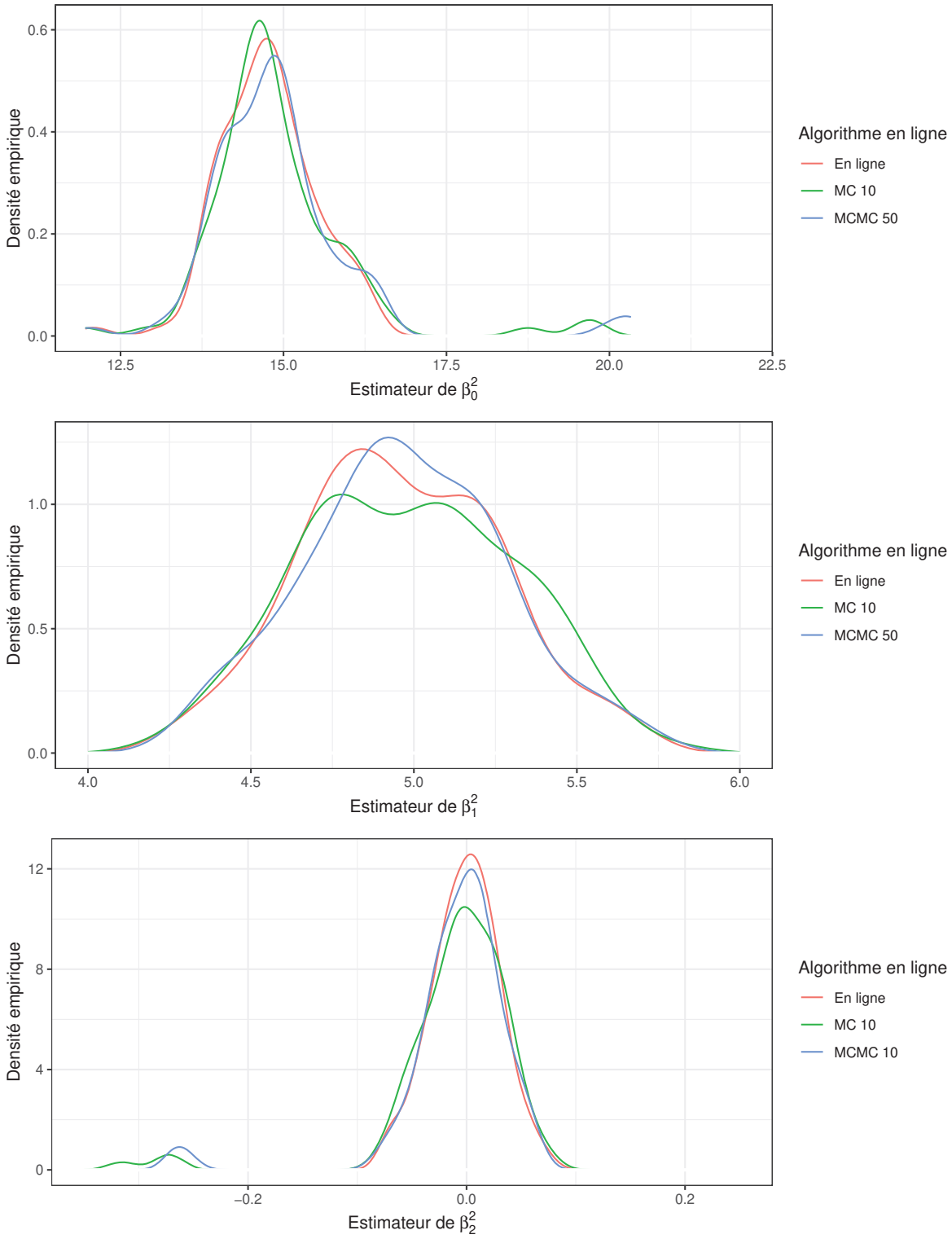


**Figure 3.11.** Densité empirique des estimateurs de $\beta_0^2$ , $\beta_1^2$, $\beta_2^2$ pour l'algorithme EM en ligne, EM en ligne Monte-Carlo et EM en ligne MCMC. Les conclusions tirées à partir de la figure 3.10 sont les mêmes pour cette présente figure.

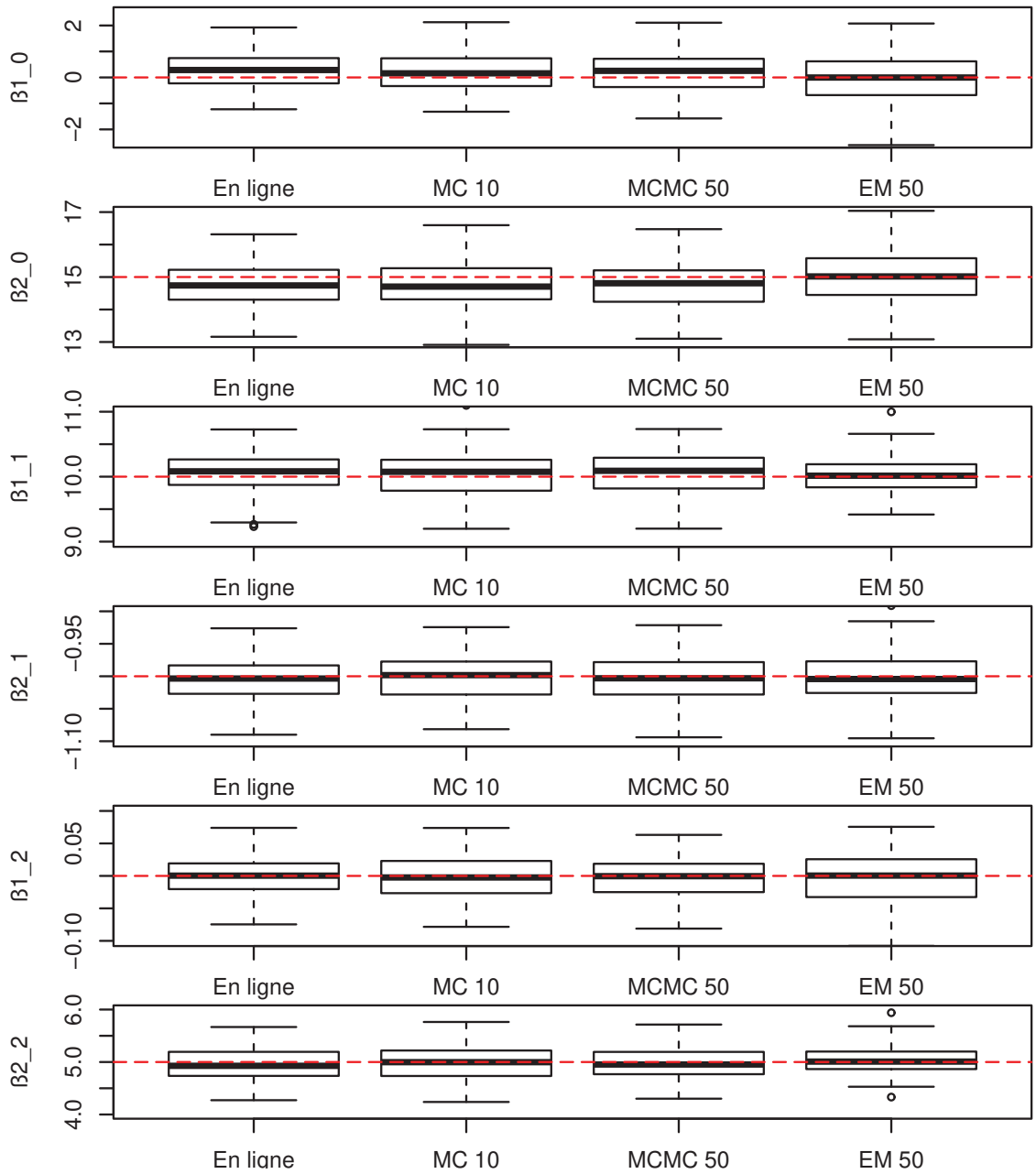


**Figure 3.12.** Boîtes à moustaches pour les paramètres $\beta_0^1$, $\beta_1^1$, $\beta_2^1$, $\beta_0^2$, $\beta_1^2$, $\beta_2^2$ après 1000 répétitions de $10^4$ itérations des algorithmes en ligne 2.4.1, 2.5.1 à 10 réalisations Monte-Carlo, 2.6.1 à 50 réalisations MCMC et 50 itérations de l'EM pour l'exemple de mélange de régression. Chacune des répétitions correspond à un nouveau jeu de données. L'algorithme EM est limité à 50 itérations (d'où EM 50). Cela signifie qu'il voit 50 fois les données tandis que les algorithmes en ligne ne les voient qu'une seule fois. En dépit de cette différence, la variance des estimateurs de l'algorithme EM en ligne n'est pas beaucoup plus grande que celle des estimateurs de l'algorithme EM standard. Les versions Monte-Carlo et MCMC de l'algorithme EM en ligne varient davantage lorsqu'on compare les écarts interquartiles.

**Tableau 3.1.** Rapport des variances et rapport des écarts médians au carré pour les algorithmes EM stochastique en ligne, MC 10 et MCMC 50 dans l'exemple de mélange de normales. La variance des estimateurs provenant de l'algorithme EM en ligne est au numérateur. La variance des estimateurs provenant de l'algorithme EM en ligne Monte-Carlo ou MCMC est au dénominateur. Ce tableau montre l'augmentation de variance causée par l'estimation de l'espérance a posteriori.

| | Var MC | MAD MC | Var MCMC | MAD MCMC |
|---|---|---|---|---|
| $\omega_1$ | 0,73 | 0,91 | 0,86 | 0,88 |
| $\mu_1$ | 0,80 | 0,82 | 0,89 | 0,93 |
| $\mu_2$ | 0,66 | 0,69 | 0,81 | 0,82 |
| $\sigma_1$ | 0,79 | 0,79 | 0,89 | 0,89 |
| $\sigma_2$ | 0,67 | 0,74 | 0,82 | 0,75 |

**Tableau 3.2.** Rapport des variances et rapport des écarts médians au carré pour les algorithmes EM stochastique en ligne, MC 10 et MCMC 50 dans l'exemple de mélange de régression. L'estimation de l'espérance a posteriori engendre une augmentation de la variance.

| | Var MC | MAD MC | Var MCMC | MAD MCMC |
|---|---|---|---|---|
| $\beta_0^1$ | 0,32 | 0,79 | 0,42 | 0,80 |
| $\beta_0^2$ | 0,43 | 0,92 | 0,36 | 0,84 |
| $\beta_1^1$ | 0,98 | 0,74 | 0,86 | 0,82 |
| $\beta_1^2$ | 0,86 | 0,66 | 0,99 | 0,97 |
| $\beta_2^1$ | 0,29 | 0,91 | 0,27 | 0,81 |
| $\beta_2^2$ | 0,27 | 0,64 | 0,32 | 0,89 |

## 3.4. Exemples de régression avec une variable latente continue

Jusqu'à présent, les cas traités impliquaient des variables latentes discrètes. Dans le cas d'une variable latente continue, la distribution a posteriori peut prendre des formes comprenant des constantes de normalisation difficiles, possiblement impossibles à intégrer lorsque la loi marginale de $X$ est compliquée. Considérons d'abord un exemple où la constante de normalisation est intégrable et la loi marginale est simple.

### 3.4.1. Exemple où la variable latente suit une distribution normale

On présente un exemple d'un modèle de régression contenant une variable latente continue dont la loi est normale. Dans cet exemple, les données proviennent de la variable aléatoire

$$Y = -20 + 10U - 5X + \epsilon, \tag{46}$$

où $\epsilon \sim \mathcal{N}(0,\frac{1}{2})$ et $U \sim$ Uniforme(0,10), de laquelle les réalisations sont observées. La variable latente $X \sim \mathcal{N}(-4,2)$ n'est pas observée. Néanmoins, sa loi et ses paramètres sont connus.

On modélise la loi conditionnelle de $Y$ par une distribution normale,

$$f(y;x,\theta) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{-\frac{(y-\beta_0+\beta_1 u+\beta_2 x)^2}{2\sigma^2}\right\} = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{-\frac{(y-\mathbf{x}^T\beta)^2}{2\sigma^2}\right\}.$$

La densité jointe s'écrit

$$f(x,y;\theta) = f(y;x,\theta)f(x) \quad = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{-\frac{(y-\beta_0+\beta_1 u+\beta_2 x)^2}{2\sigma^2}\right\} \frac{1}{(2\pi\sigma_x^2)^{\frac{1}{2}}} \exp\left\{-\frac{(x+4)^2}{2\sigma_x^2}\right\}.$$

La loi a posteriori de la variable latente suit aussi une normale,

$$\begin{aligned}
p(x;y,\theta) &= \frac{f(x,y;\theta)}{g(y;\theta)} \\
&= \frac{f(x,y;\theta)}{\int f(x,y;\theta)\nu(\mathrm{d}y)} \\
&= \frac{\frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{-\frac{(y-\beta_0+\beta_1 x+\beta_2 w)^2}{2\sigma^2}\right\} \frac{1}{(2\pi\sigma_w^2)^{\frac{1}{2}}} \exp\left\{-\frac{(x-\mu_0)^2}{2\sigma_x^2}\right\}}{\left(2\pi\left(\sigma^2+\beta_2^2\sigma_x^2\right)\right)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left[\frac{\mu_0}{\sigma_x^2}+\frac{(y-(\beta_0+\beta_1 u))^2}{\sigma^2}-\frac{(y-\beta_0-\beta_1 u)\beta_2\sigma_x^2+\mu_0\sigma^2)}{\beta_2^2\sigma_x^2+\sigma^2}\right]\right\}} \\
&= \left(2\pi\frac{\sigma_x^2\sigma^2}{\beta_2^2\sigma_x^2+\sigma^2}\right)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\frac{\left(x-\frac{\mu_0\sigma^2+\beta_2(y-\beta_0-\beta_1 u)}{\beta_2^2\sigma_x^2+\sigma^2}\right)^2}{\frac{\sigma_x^2\sigma^2}{\beta_2^2\sigma_x^2+\sigma^2}}\right\}.
\end{aligned}$$

Ceci correspond à la densité d'une loi normale. On connaît ainsi les deux premiers moments de la loi a posteriori de $X$, ce qui permet d'utiliser l'algorithme 2.4.1 étant donné que les deux premiers moments apparaissent dans les statistiques exhaustives :

$$S_1(X,Y) = \begin{bmatrix} 1 & u & X \\ u & u^2 & uX \\ X & Xu & X^2 \end{bmatrix}$$

et

$$S_2(X,Y) = \begin{bmatrix} Y \\ uY \\ XY \end{bmatrix}.$$

Les espérances de ces matrice sont :

$$\mathbb{E}_{\bar{\theta}(s)}[S_1(X,Y) \mid Y=y] \overset{\text{déf}}{=} \begin{bmatrix} 1 & u & \mathbb{E}_{\bar{\theta}(s)}[X \mid Y=y] \\ u & u^2 & u\,\mathbb{E}_{\bar{\theta}(s)}[X \mid Y=y] \\ \mathbb{E}_{\bar{\theta}(s)}[X \mid Y=y] & \mathbb{E}_{\bar{\theta}(s)}[X \mid Y=y]\,u & \mathbb{E}_{\bar{\theta}(s)}[X^2 \mid Y=y] \end{bmatrix}$$

et

$$\mathbb{E}_{\bar{\theta}(s)}[S_2(X,Y) \mid Y=y] \overset{\text{déf}}{=} \begin{bmatrix} y \\ uy \\ \mathbb{E}_{\bar{\theta}(s)}[X \mid Y=y]\,y \end{bmatrix},$$

où

$$\mathbb{E}_{\bar{\theta}(s)}\left[X \mid Y = y\right] = \frac{\mu_0\sigma^2 + \hat{\beta}_2(y - \hat{\beta}_0 - \hat{\beta}_1 u)}{\hat{\beta}_2^2\sigma_x^2 + \sigma^2}$$

et

$$\mathbb{E}_{\bar{\theta}(s)}\left[X^2 \mid Y = y\right] = \left(\frac{\mu_0\sigma^2 + \hat{\beta}_2(y - \hat{\beta}_0 - \hat{\beta}_1 u)}{\hat{\beta}_2^2\sigma_x^2 + \sigma^2}\right)^2 + \left(\frac{\sigma_x^2\sigma^2}{\hat{\beta}_2^2\sigma_x^2 + \sigma^2}\right).$$

La figure 3.13 montre la variabilité qu'ajoute l'estimation de $\mathbb{E}_{\bar{\theta}(s)}\left[X \mid Y = y\right]$ par Monte-Carlo et MCMC. Le paramètre à l'étape de maximisation des algorithmes EM en ligne est $\beta = (\mathbf{x}\mathbf{x}^T)^{-1}\mathbf{x}^T y$ qui est obtenu en maximisant $f(x,y;\theta)$. On fait l'analyse de cet exemple aux sections 4.1 et 4.2.

### 3.4.2. Cas où la vraisemblance incomplète est connue

L'EM est très utile lorsque $g_\theta$ est inconnue. Or, dans l'exemple de la section 3.4.1, $g_\theta$ est connue. Cependant, il est difficile de déterminer l'estimateur du maximum de vraisemblance analytiquement. L'algorithme EM offre une autre voie pour l'obtenir. La fonction de log-vraisemblance observée est la suivante :

$$\begin{aligned}\log\sum_{j=1}^{n} g(y_i;\theta) = &-\frac{n}{2}\log\left(2\pi\left(\sigma^2 + \beta_2^2\sigma_x^2\right)\right)\\ &-\frac{1}{2}\left(\frac{n\mu_0^2}{\sigma_x^2} + \frac{\sum_{j=1}^{n}\left(y_i - (\beta_0 + \beta_1 u_i)\right)^2}{\sigma^2} - \frac{\sum_{j=1}^{n}\left(\left(y_i - (\beta_0 + \beta_1 u_i)\right)\beta_2\sigma_x^2 + \mu_0\sigma^2\right)^2}{\left(\beta_2^2\sigma_x^2 + \sigma^2\right)\sigma^2\sigma_x^2}\right). \quad (47)\end{aligned}$$

Supposons qu'on voulait obtenir l'estimateur du maximum de vraisemblance directement sans faire intervenir les variables latentes comme c'est le cas dans le cas de l'EM avec l'espérance a posteriori. Il faudrait trouver les dérivées partielles et résoudre le système d'équation. Les dérivées partielles de la fonction de vraisemblance mènent au système d'équations suivant :

$$\beta_0 = -\beta_1\sum_{j=1}^{n} u_j + \sum_{j=1}^{n} y_j - n\beta_2\mu_0,$$

$$\beta_1 = \frac{\sum_{j=1}^{n} u_j(y_j - \beta_2\mu_0 - \beta_0)}{\sum u_j^2},$$

$$\begin{aligned}0 = &-\frac{\beta_2\sum_{j=1}^{n}\left(\sigma_x^2\left(-\beta_1 u_j + y_j - \beta_0\right)\beta_2 + \mu_0\sigma^2\right)^2}{\sigma^2\left(\sigma_x^2\beta_2^2 + \sigma^2\right)}\\ &+\frac{\sum_{j=1}^{n}\left(-\beta_1 u_j + y_j - \beta_0\right)\left(\sigma_x^2\left(-\beta_1 u_j + y_j - \beta_0\right)\beta_2 + \mu_0\sigma^2\right)}{\sigma^2} - n\sigma_x^2\beta_2. \quad (48)\end{aligned}$$

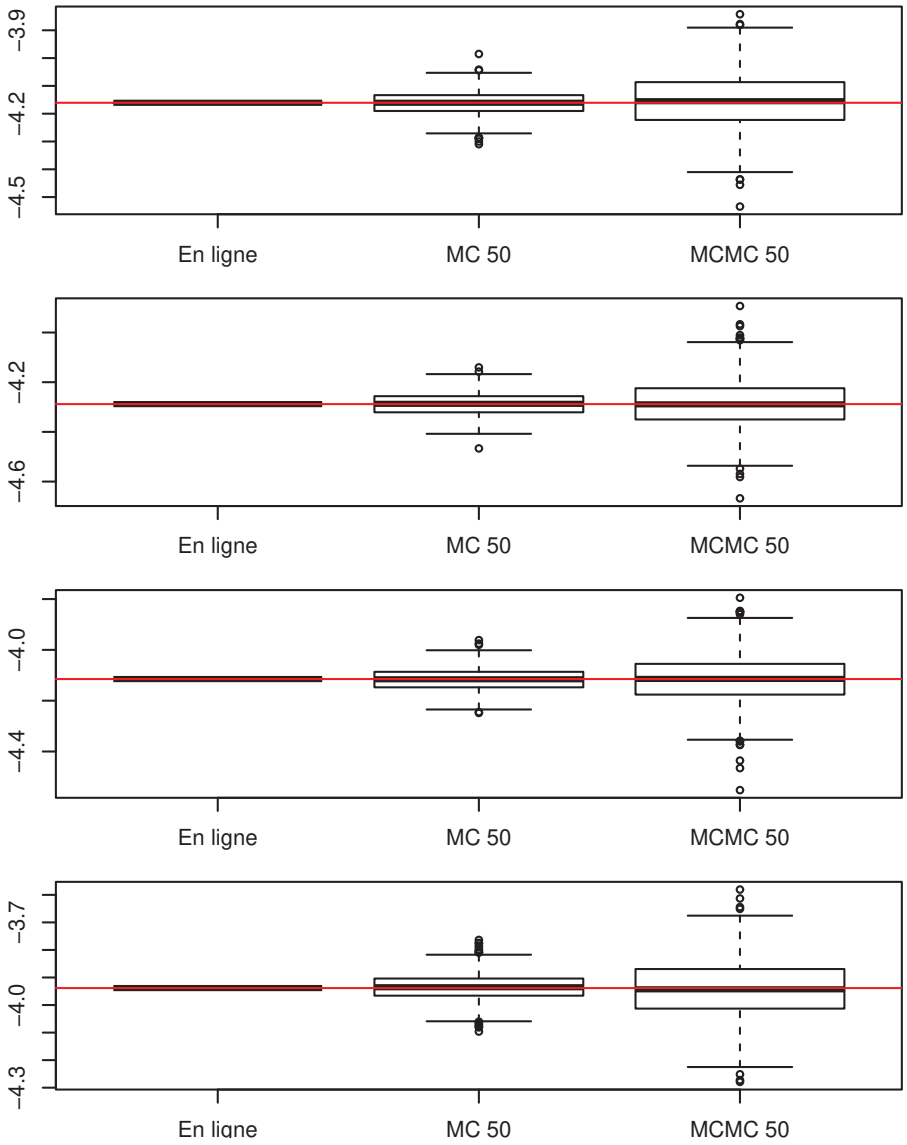


**Figure 3.13.** La boîte à moustache des estimateurs de l'espérance a posteriori où chaque ligne correspond à différents paramètres $\theta$ générés à différentes itérations de l'EM. Cette figure illustre le niveau de variabilité des estimateurs qui remplacent l'espérance a posteriori dans l'algorithme EM en ligne. Sans surprise, pour 50 réalisations Monte-Carlo et 50 réalisations MCMC, l'estimateur Monte-Carlo est moins variable puisqu'il y a de la corrélation résiduelle dans la chaîne MCMC.

Le paramètre $\beta_2$ est la solution d'un polynôme qui a possiblement plusieurs racines. Puisque les autres équations dépendent de $\beta_2$, il pourrait y exister plusieurs solutions. Comme on ne peut pas obtenir directement les estimateurs des paramètres, il faudrait procéder de manière numérique si on voulait poursuivre dans cette voie avec la méthode du gradient, par exemple. Une autre solution serait de revenir à l'EM qui n'exige ni la connaissance de $g(y;\theta)$ ni son gradient.

*Illustration de la convergence des paramètres pour différents algorithmes.* Le fait qu'il y ait seulement trois paramètres à estimer dans cet exemple permet d'illustrer la convergence des différents algorithmes abordés depuis le début du mémoire. Les trajectoires de convergence de l'algorithme 2.4.1, de l'algorithme 2.5.1, de l'algorithme du gradient ainsi que l'algorithme du gradient stochastique sont illustrées. Le même jeu de données de taille $10^5$ est traité par chacun des algorithmes qui disposent du même paramètre $\beta$ initial. On a choisi d'illustrer $\beta_0$ et $\beta_2$ dans le but de visualiser la trajectoire de la convergence en trois dimensions. On rappelle que les algorithmes de type gradient convergent vers le paramètre de vraisemblance maximale tandis que les algorithmes 2.4.1 et 2.5.1 convergent vers le paramètre qui minimise la divergence de Kullback-Leibler entre $\pi$ et $g_\theta$. À la figure 3.14, la trajectoire de l'algorithme du gradient est illustrée. La convergence se fait lentement (le pas n'est pas optimal) sans oscillation. Il s'agit de la courbe qui correspond au flot de l'équation différentielle $\frac{\mathrm{d}\theta}{\mathrm{d}t} = -\nabla\ell(\theta(t))$, avec une certaine condition initiale $\theta(0)$ imposée, discrétisée par la méthode d'Euler. L'algorithme du gradient stochastique qu'on peut observer à la figure 3.15, converge plus lentement en oscillant fortement. À la figure 3.16, la trajectoire de l'algorithme 2.4.1 est illustrée. On observe que la région finale est atteinte rapidement suivant une trajectoire similaire à celle de l'algorithme du gradient, mais qu'on y oscille grandement. À la figure 3.17, la trajectoire de l'algorithme 2.5.1 est illustrée. Chacune des trajectoires des vraisemblances en fonction de $\beta_0$ et $\beta_2$ suivent globalement la même trajectoire, avec différents niveaux de bruit, que l'algorithme du gradient dont le flot de l'équation différentielle est illustré en 3.14. Il y a de fortes oscillations pour l'algorithme du gradient stochastique et de plus petites perturbations pour les algorithmes EM en ligne. Les algorithmes en ligne ne convergent pas vers le maximum de vraisemblance, mais vers le minimum de la divergence de Kullback-Leibler entre $g_\theta$ et $\pi$. Malgré cette nuance, les trajectoires sont fortement similaires.

### 3.4.3. Exemple où la variable latente suit une distribution de Weibull

Jusqu'à présent, les données provenaient de modèles dont la loi a posteriori était connue et calculable. À présent, on se place dans la situation contraire. L'exemple suivant concerne un modèle plus complexe dont la loi a posteriori n'est pas connue exactement. Cependant, elle est connue à une constante de normalisation près. Parmi les algorithmes vus jusqu'à présent, l'algorithme 2.6.1 est le seul compatible dans cette situation. Au chapitre 4, des techniques de réduction de variance seront appliquées afin de réduire la variance de l'algorithme. Les données sont des réalisations de la variable aléatoire

$$Y = -20 + 10U - 5X + \epsilon,$$

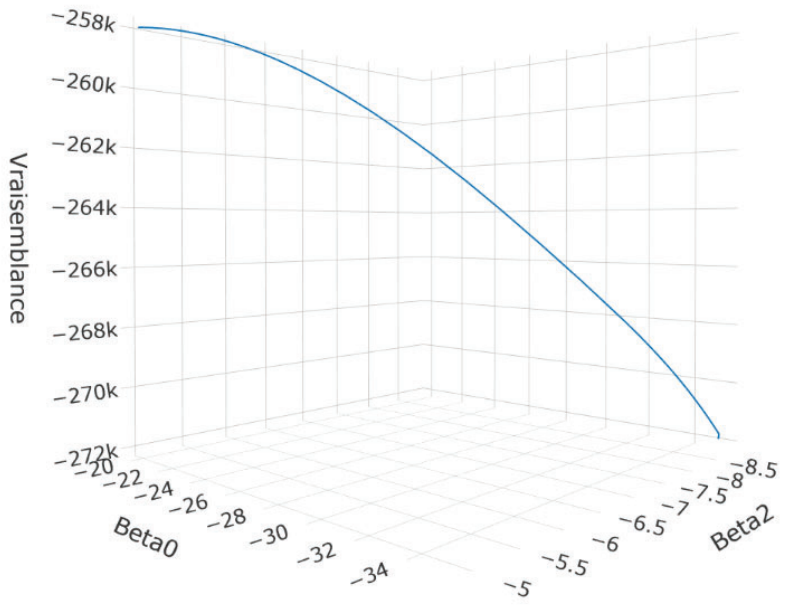


**Figure 3.14.** La vraisemblance en fonction des estimateurs des paramètres pour l'algorithme du gradient qu'on a vu en 6. La courbe bleue représente les points $\beta_0$, $\beta_2$ et la vraisemblance. L'estimateur $\beta_1$ est omis étant donné qu'il converge rapidement. La courbe représente le flot de l'équation différentielle.

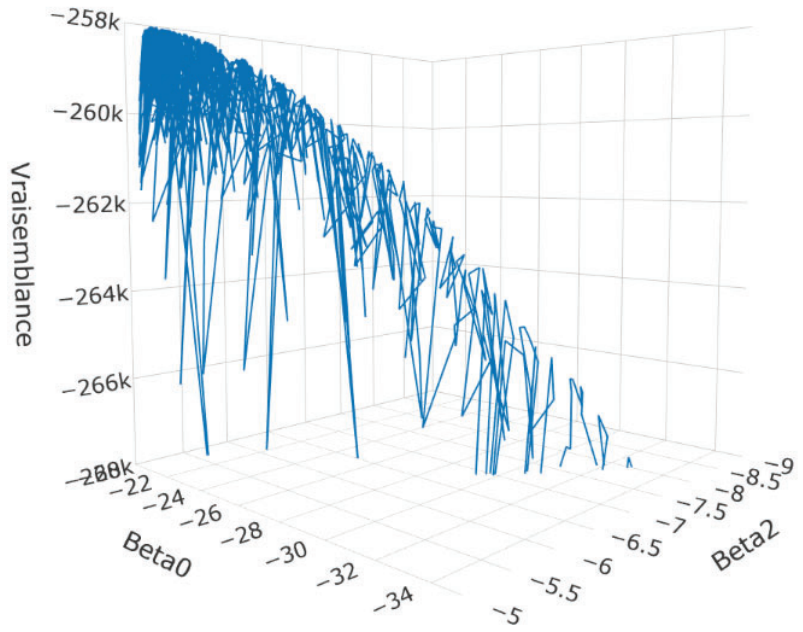


**Figure 3.15.** La vraisemblance en fonction des estimateurs des paramètres pour l'algorithme du gradient stochastique. La courbe est une version bruitée du flot de la figure 3.14.

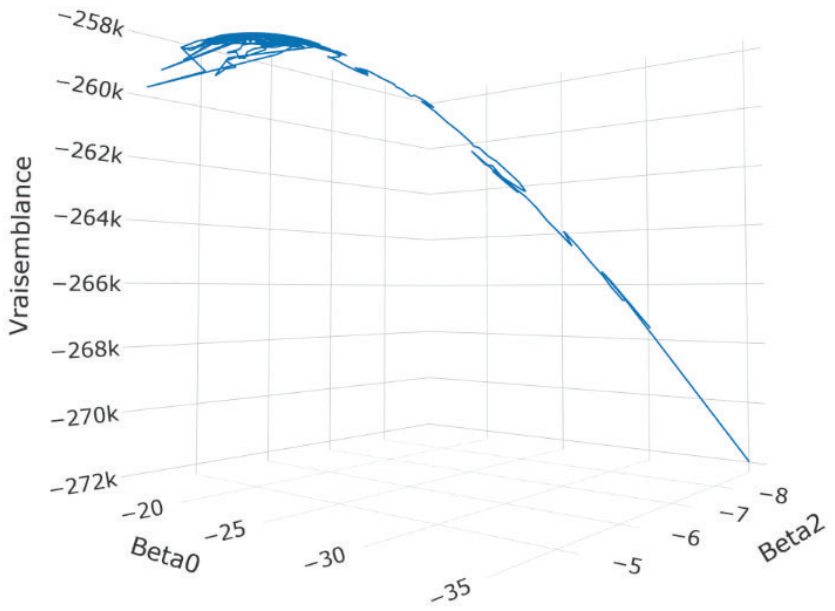


**Figure 3.16.** La vraisemblance en fonction des estimateurs des paramètres $\beta_0$ et $\beta_2$ pour l'algorithme EM en ligne. Les estimateurs oscillent autour de la valeur du paramètre.

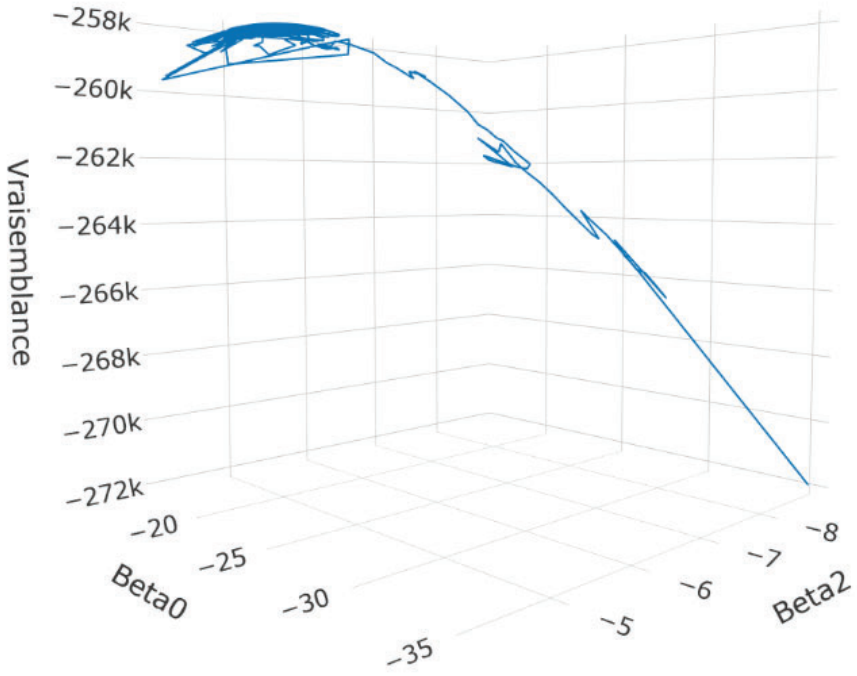


**Figure 3.17.** La vraisemblance en fonction des estimateurs des paramètres pour l'algorithme EM en ligne MC 10. La courbe ressemble fortement à celle de l'algorithme EM en ligne à la figure 3.16.

où $\epsilon \sim \mathcal{N}(0,\sigma^2 = \frac{1}{2})$ et $U \sim$ Uniforme(0,10) est observée. La variable latente $X \sim$ Weibull(6,3) n'est pas observée. Néanmoins, sa loi et ses paramètres sont connus. Sa fonction de densité, illustrée à la figure 3.18, est la suivante :

$$f(x) = 2\left(\frac{x}{3}\right)^5 \exp\left\{-\left(\frac{x}{3}\right)^6\right\}, \quad x \geq 0.$$

On modélise la loi conditionnelle de $Y \mid X$ par la densité $f(y;x;\theta) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{-\frac{(y-\beta_0+\beta_1 u+\beta_2 x)^2}{2\sigma^2}\right\}$. La loi jointe est

$$\begin{aligned} f(x,y;\theta) &= f(y;x,\theta)f(x) \\ &= \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{-\frac{(y-\beta_0-\beta_1 u-\beta_2 x)^2}{2\sigma^2}\right\} 2\left(\frac{x}{3}\right)^5 \exp\left\{-\left(\frac{x}{3}\right)^6\right\}. \quad (49) \end{aligned}$$

La loi a posteriori de $X$ est

$$p(x;y,\theta) = \frac{f(x,y;\theta)}{\int f(x,y;\theta)\nu(\mathrm{d}y)} = \frac{\frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{-\frac{(y-\beta_0-\beta_1 u-\beta_2 x)^2}{2\sigma^2}\right\} 2\left(\frac{x}{3}\right)^5 \exp\left\{-\left(\frac{x}{3}\right)^6\right\}}{\int \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{-\frac{(y-\beta_0-\beta_1 u-\beta_2 x)^2}{2\sigma^2}\right\} 2\left(\frac{x}{3}\right)^5 \exp\left\{-\left(\frac{x}{3}\right)^6\right\} \nu(\mathrm{d}y)}. \quad (50)$$

Il n'est pas aisé d'échantillonner la loi a posteriori $p(x;y)$ de la variable latente. La constante de normalisaton n'a pas de forme discernable. L'algorithme de Metropolis-Hastings est employé pour générer des réalisations de cette loi. Le paramètre à l'étape de maximisation est $\hat{\beta} = (\mathbf{x}\mathbf{x}^T)^{-1}\mathbf{x}^T y$ qui est obtenu en maximisant $f(x,y;\theta)$. Cet exemple est étudié à la section 4.3.

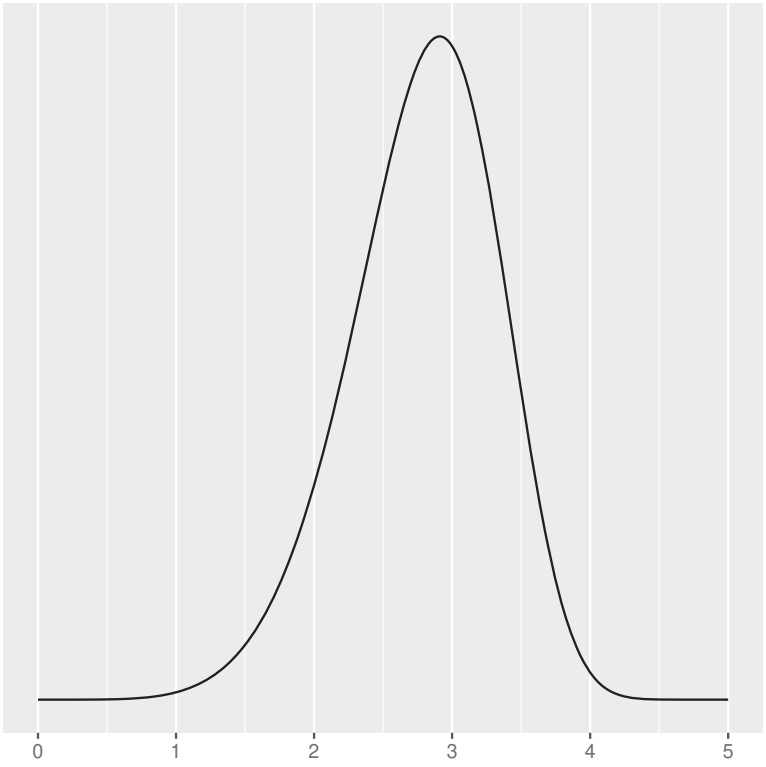


**Figure 3.18.** Représentation de la densité de la loi a priori $X$ dans l'exemple 3.4.3 est une loi de Weibull.

# Chapitre 4

# Approches permettant de réduire la variance dans les algorithmes EM en ligne

Comme montré précédemment, la variabilité des estimateurs augmente lorsque l'espérance $\mathbb{E}_{\bar{\theta}(s)}\left[S(X,Y) \mid Y\right]$ n'est pas calculable. De plus, rappelons que l'estimateur MCMC admet un biais. Que se passe-t-il lorsqu'on jumèle les méthodes de réduction de variance à une approximation stochastique de l'algorithme EM en ligne ? Plus précisément, quel est l'effet des méthodes de réduction de variance sur la convergence des estimateurs vers le minimum de la divergence de Kullback-Leibler ?

On analysera les effets de trois méthodes de réduction de variance. Premièrement, on augmentera le nombre de réalisations pour les méthodes Monte-Carlo et MCMC. Deuxièmement, on introduira une variable de contrôle pour une espérance Monte-Carlo. Troisièmement, on introduira une variable de contrôle pour une espérance MCMC.

## 4.1. Réduction de variance par l'augmentation de la taille d'échantillon

On applique 1000 fois les algorithmes 2.4.1, 2.5.1 et 2.6.1 à l'exemple de régression avec une variable latente continue, présenté à la section 3.4.1, pour différentes séquences de $10^4$ données provenant de la variable aléatoire $Y$ introduite en 46. La valeur de $\{\gamma_t\}$ est $\frac{0{,}51}{t^{0{,}51}}$. La valeur initiale de $s_0$ est obtenue en inversant le premier estimateur de $\theta_0$ obtenu à partir de l'estimateur du maximum de vraisemblance basé sur 100 observations complètes dont on aurait connaissance a priori. Pour les algorithmes comprenant des méthodes de type Monte-Carlo ou MCMC, il y a un choix à faire quant au nombre de réalisations sur lequel l'algorithme repose. Le choix par défaut demeure $m = 10$ pour la méthode de Monte-Carlo et $m = 50$ pour la méthode MCMC. Ce choix est dicté par l'efficacité computationnelle dans chacun des cas et par la nécessité d'avoir un échantillon suffisamment grand pour réduire la corrélation dans la chaîne dans le cas MCMC. Comme ces méthodes convergent quand

le nombre de réalisations augmente, on vérifie que la variance des estimateurs construits par les algorithmes 2.5.1 et 2.6.1 est réduite en faisant croître $m$ dans le cadre de l'exemple présenté à la section 3.4.1. L'estimateur Monte-Carlo a pour variance $\frac{\sigma^2}{m}$ lorsqu'elle est basée sur une collection de réalisations iid. Faire croître $m$ fait diminuer la variance de l'estimateur. L'algorithme 2.5.1 est implémenté pour $m = 1$, $m = 10$, $m = 100$ et $m = 1000$. L'algorithme 2.6.1 est mis en marche avec $B = 50$, $m = 100$ et $m = 150$. On a représenté les graphiques de densité empirique dans les figures 4.1 et 4.2 qui reposent sur 1000 réplicats de $10^4$ données. À partir de 100 réalisations, les courbes de densité empirique de 2.5.1 et 2.6.1 épousent la courbe associée à l'algorithme 2.4.1, l'algorithme dans lequel l'espérance a posteriori peut être calculée. La vraisemblance du paramètre courant en fonction de l'itération est représentée pour certains algorithmes aux figures 4.3 et 4.4. À partir de 10 réalisations Monte-Carlo, la convergence de la valeur de la vraisemblance de l'algorithme 2.5.1 est similaire à celle de l'algorithme 2.4.1. Il suffit de 50 réalisations MCMC pour avoir une convergence comparable à l'algorithme 2.4.1. Néanmoins, passer de 50 à 100 réalisations MCMC permet de réduire de quelques points de pourcentage le rapport des écarts médians comme le montre le tableau 4.1. L'écart médian absolu au carré de l'algorithme EM en ligne est placé au numérateur du rapport alors que l'écart médian absolu au carré de l'algorithme EM en ligne MC ou MCMC est placé au dénominateur. De manière similaire, passer de 10 à 100 réalisations Monte-Carlo permet de réduire complètement la variance. Par conséquent, la taille d'échantillon Monte-Carlo réduisant la totalité de la variance supplémentaire est située entre 10 et 100.

| | MC 1 | MC 10 | MC 100 | MC 1000 | MCMC 50 | MCMC 100 |
|---|---|---|---|---|---|---|
| $\beta_0$ | 0,41 | 0,83 | 0,98 | 1,00 | 0,88 | 0,91 |
| $\beta_1$ | 0,45 | 0,88 | 1,03 | 1,00 | 0,91 | 0,94 |
| $\beta_2$ | 0,46 | 0,91 | 1,00 | 0,97 | 0,93 | 0,96 |

**Tableau 4.1.** Rapport des écarts absolus médians au carré entre l'algorithme 2.4.1 et l'algorithme en ligne approximatif après $10^4$ réalisations. Dans le cas des algorithmes EM en ligne Monte-Carlo comme dans le cas des algorithmes EM en ligne MCMC, l'augmentation du nombre de réalisations sur lequel l'estimateur Monte-Carlo ou MCMC repose a pour effet de réduire le rapport de variance entre l'algorithme EM en ligne Monte-Carlo ou MCMC et l'algorithme EM en ligne habituel. À partir de 100 réalisations MC, le surplus de variance est quasiment intégralement effacé.

## 4.2. Réduction de variance dans l'algorithme 2.5.1

Pour l'exemple de modèle présenté à la section 3.4.1, une variable de contrôle pour la méthode de Monte-Carlo permet de réduire la variance lorsque l'algorithme 2.5.1 construit un estimateur du paramètre. Pour cet exemple, l'initialisation est la même qu'à la section 4.1 avec le nombre $m$ de réalisations Monte-Carlo qui est fixé à 10. On propose une construction qui pourrait être utile dans certaines circonstances. Supposons que l'espérance $\mathbb{E}_{\bar{\theta}(s)}[S(X,Y) \mid Y]$ ne soit pas facile à calculer, mais que la loi a posteriori $P_{\bar{\theta}(s)}(x;y)$ de $X \mid Y$, de même que son espérance et sa variance, soient connus. Supposons que dans le modèle de régression avec une variable latente continue présenté en 3.4.1, les espérances des statistiques exhaustives ne soient pas calculables (elles le sont en réalité, c'est un exemple joué qui sert d'illustration).

Il faudrait disposer d'une variable de contrôle fortement corrélée avec les réalisations Monte-Carlo afin de réduire la variance de l'estimateur de Monte-Carlo, s'il y en a aucune qui apparaît naturellement. La stratégie consiste à ajouter un faible bruit $\epsilon$ de loi normale d'espérance nulle et de variance faible (par exemple, on prendra une variance de $10^{-4}$ ou $10^{-6}$). Soit la variable transformée $\tilde{X} = X - c(\phi(X) - \mathbb{E}[\phi(X)])$. La variable $\epsilon$ est indépendante de la variable $X$. Les variables de contrôle notées

$$\phi(X) = X + \epsilon$$

ont une espérance identique à celle de $X$ et une variance connue. De là, il reste à trouver un coefficient $c$ près la valeur optimale $c^* = \text{Cov}[X,\phi(X)]/\text{Var}[\phi(X)]$, montrée en 26, ce qui est envisageable lorsqu'on connaît les lois de $X$ et les deux premiers moments de $\phi(X)$. Par exemple, la covariance s'écrit

$$\text{Cov}\left[X,X+\epsilon\right] = \mathbb{E}\left[X\left(X+\epsilon\right)\right] - \mathbb{E}[X]\,\mathbb{E}[X+\epsilon] = \text{Var}[X],$$

et la variance, par indépendance de $X$ et $\epsilon$, s'écrit

$$\text{Var}[\phi(X)] = \text{Var}[X+\epsilon] = \text{Var}[X] + \text{Var}[\epsilon].$$

L'algorithme qui suit ressemble à l'algorithme EM en ligne MCMC 2.5.1 dans la mesure où c'est l'EM en ligne habituel, sauf que l'espérance Monte-Carlo contient une variable de contrôle à l'étape d'espérance. On réitère que son utilité est limitée au cas où l'espérance des statistiques exhaustives serait difficile à calculer, mais que les deux premiers moments de la loi a posteriori de $X$ seraient connus.

Comme c'était le cas pour les algorithmes EM en ligne précédents, on initialise $s_0$ quelconque duquel on trouve $\hat{\theta}_0$. On connaît $Y_1$ qui est rendue disponible. On obtient des réalisations Monte-Carlo $X_{1,1},\ldots,X_{1,m}$ en échantillonnant directement la loi a posteriori de $X_1$. On calcule la quantité $c$ qui dépend de la variance de $X_1$ et de $\epsilon$ qui sont connues. À l'aide des réalisations et de la quantité $c$, on crée des variables à variance réduite en intégrant la variable de contrôle $\phi(X)$ et son espérance connue pour obtenir les réalisations $\tilde{X}_{1,1},\ldots,\tilde{X}_{1,m}$

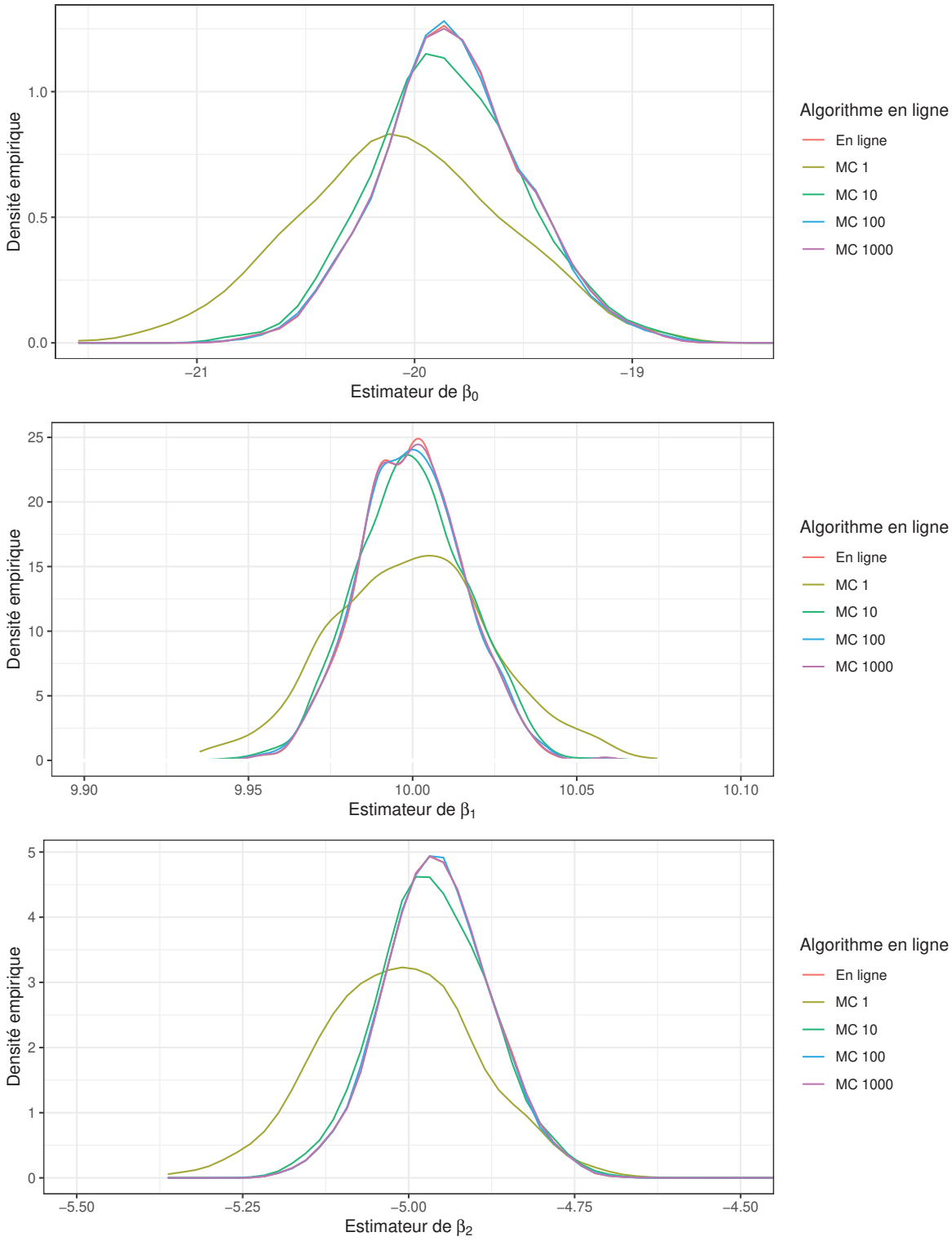


**Figure 4.1.** Densité empirique des estimateurs de $\beta_0$, $\beta_1$, $\beta_2$ dans le cas de l'algorithme EM en ligne Monte-Carlo 2.5.1 pour l'exemple de la variable latente continue suivant une normale. En augmentant le nombre de réalisations Monte-Carlo, la densité empirique des estimateurs de l'algorithme EM en ligne Monte-Carlo 2.5.1 approche la densité empirique des estimateurs de l'algorithme 2.4.1. La normalité asymptotique semble atteinte lorsque le nombre de réalisations Monte-Carlo dépasse 100.

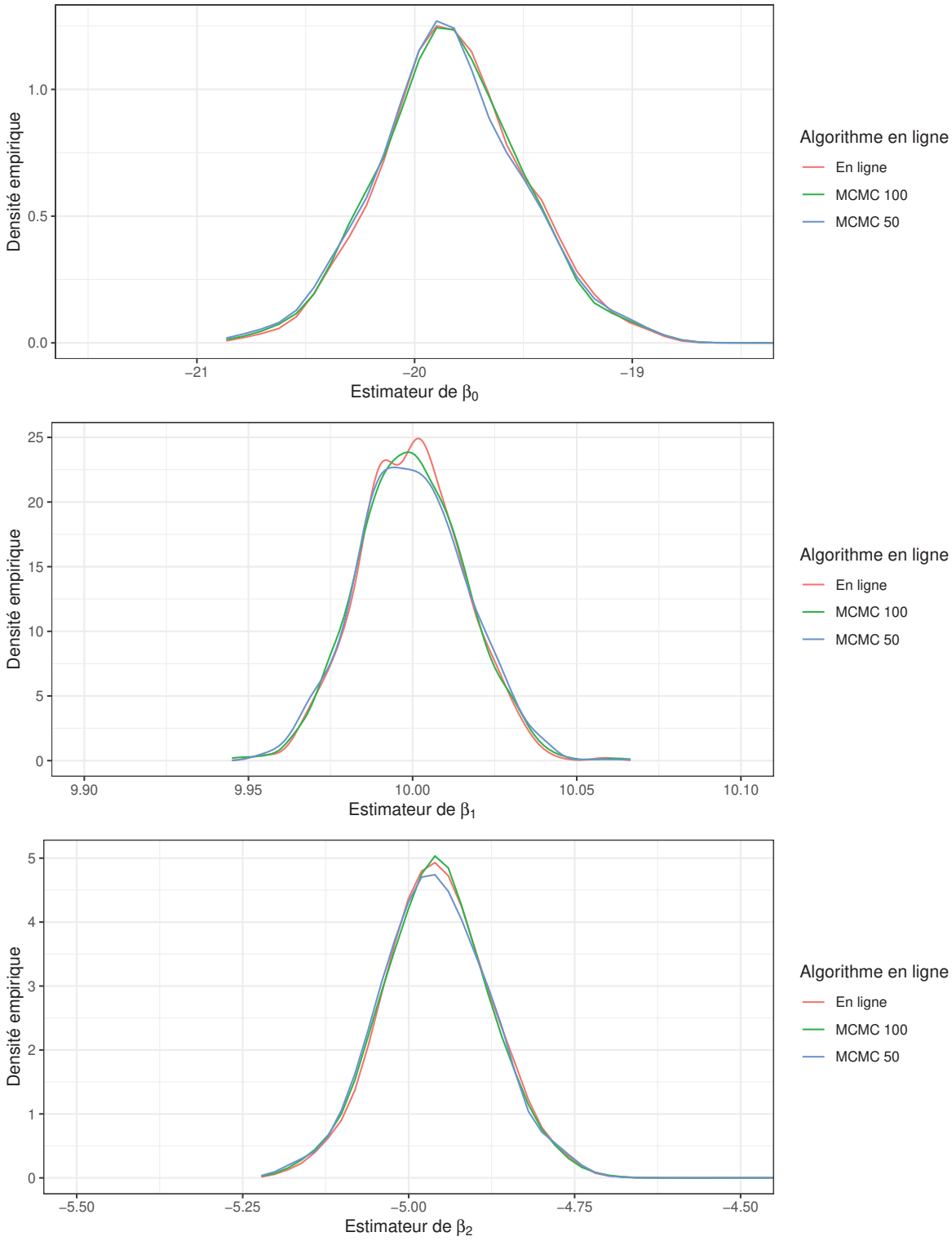


**Figure 4.2.** Densité empirique des estimateurs de $\beta_0$, $\beta_1$, $\beta_2$ dans le cas de l'algorithme EM en ligne MCMC 2.6.1 pour l'exemple de la variable latente continue suivant une normale. En augmentant le nombre de réalisations Monte-Carlo, la densité empirique des estimateurs de l'algorithme EM en ligne MCMC 2.6.1 approche la densité empirique des estimateurs de l'algorithme EM en ligne 2.4.1. La normalité asymptotique semble atteinte.

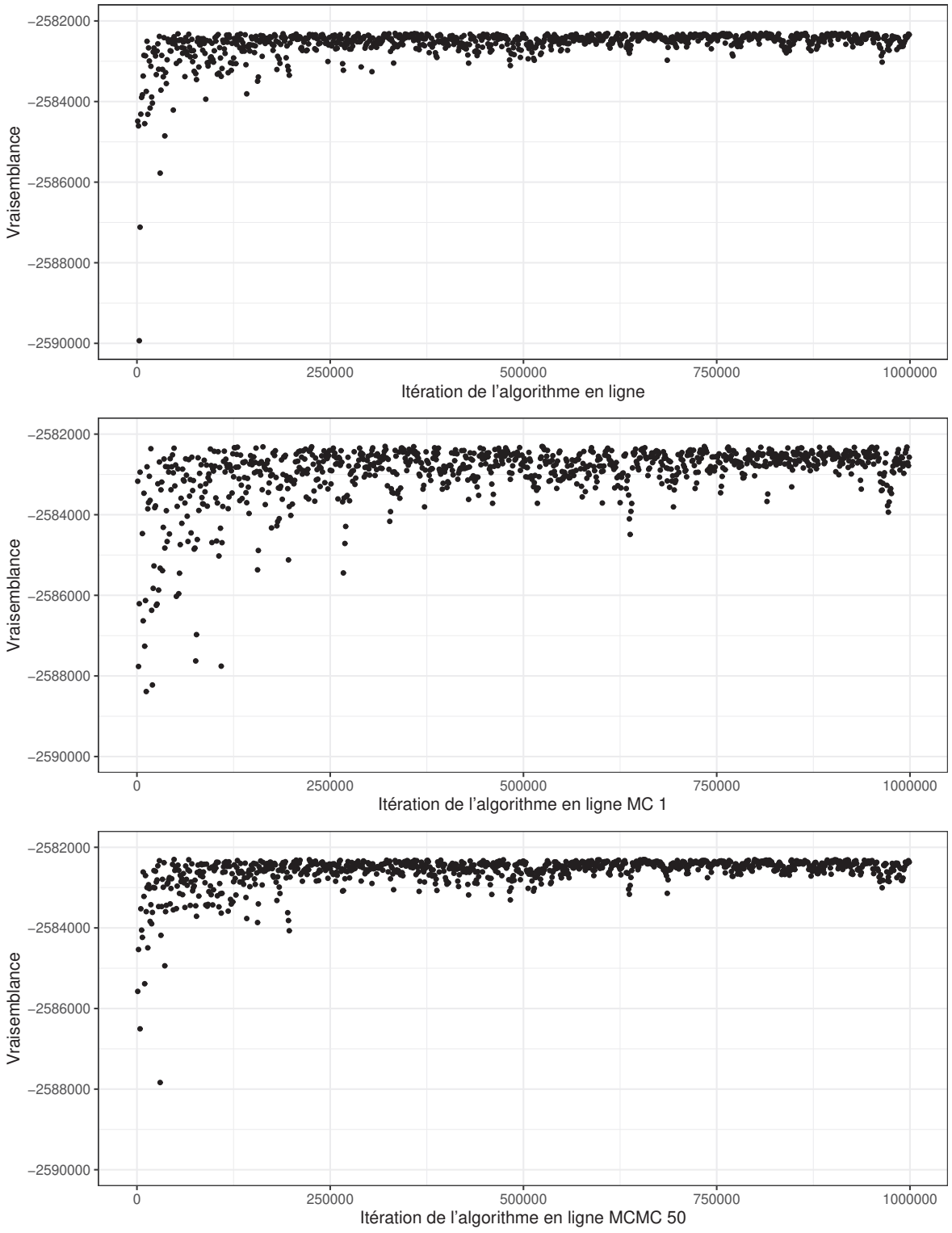


**Figure 4.3.** Évolution de la vraisemblance au paramètre courant de l'algorithme en fonction de l'itération pour différents algorithmes. Pour chacun de ceux-ci, la vraisemblance augmente rapidement pendant les premières itérations, puis semble se stabiliser, même si des variations dont la magnitude décroît subsistent. La convergence se fait de manière plus lente pour un algorithme contenant une seule réalisation Monte-Carlo. On perçoit également une convergence légèrement plus lente pour la première moitié des itérations pour un algorithme EM en ligne MCMC contenant 50 réalisations MCMC.

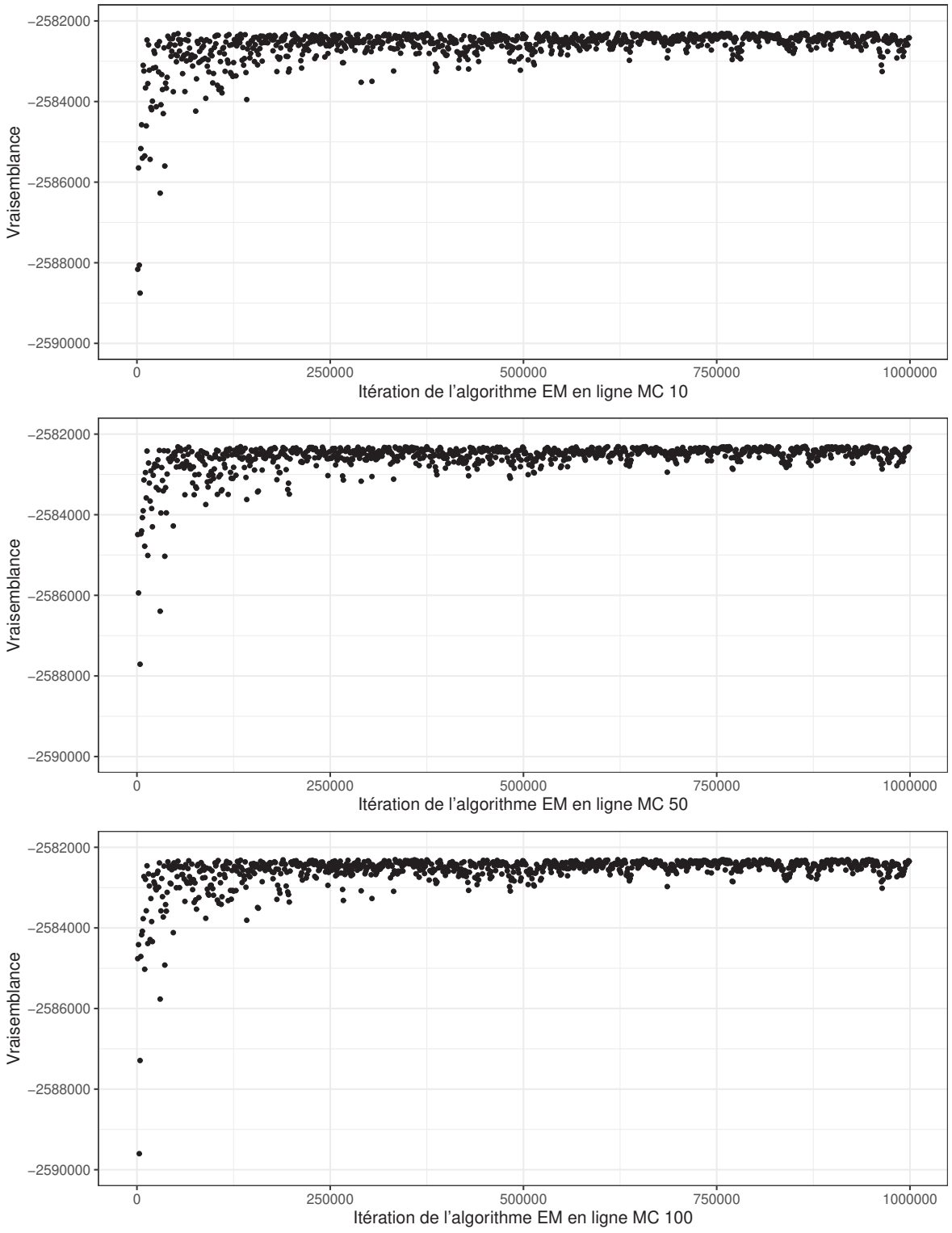


**Figure 4.4.** Évolution de la vraisemblance en fonction de l'itération pour des algorithmes en ligne Monte-Carlo 2.5.1 à différents nombres de réalisations Monte-Carlo. De façon générale, les trajectoires se ressemblent beaucoup.

à variance réduite. On peut ensuite obtenir un estimateur $\hat{\mu}^{ZV} = \frac{1}{m}\sum_{i=1}^{m} S(\tilde{X}_{k,i}, Y_k)$ de l'espérance a posteriori de la statistique exhaustive. L'étape d'appproximation stochastique est la même que pour les autres algorithmes EM en ligne, sauf que l'on insère l'estimateur $\hat{\mu}^{ZV}$ à la place de l'espérance a posteriori. À l'aide de $s_1$ obtenu à l'étape précédente, on trouve trouve $\hat{\theta}_1$. On répète le processus jusqu'à l'itération $N$.

**Algorithme 4.2.1.** L'algorithme EM par approximation stochastique en ligne MC zéro-variance.

---

**Entrées** Étant donné une suite positive décroissante $\{\gamma_t\}$, $N, m \in \mathbb{N}$ et $s_0 \in \mathcal{S}$ ;
**Initialisation ;**
$\hat{\theta}_0 = \bar{\theta}(s_0)$ ;
$k := 1$ ;
**Tant que** $k < N + 1$ **:**
*Générer* $Y_k \sim \pi$ ;
*Générer* $X_{k,1}, \ldots, X_{k,m}$ à *partir de* $P_{\hat{\theta}_{k-1}}(x_k; y_k)$ ;
*Calculer* $c = \frac{\mathrm{Var}[X_k]}{\mathrm{Var}[X_k]+\mathrm{Var}[\epsilon]}$ ;
*Calculer* $\tilde{X}_{k,1}, \ldots, \tilde{X}_{k,m}$ où $\tilde{X}_{k,m} = X_{k,m} - c(\phi(X_{k,m}) - \mathbb{E}[\phi(X_{k,m})])$ ;
*Calculer* $\hat{\mu}^{ZV} = \frac{1}{m}\sum_{i=1}^{m} S(\tilde{X}_{k,i}, Y_k)$ ;
*Calculer* $s_k = s_{k-1} - \gamma_k \left(s_{k-1} - \hat{\mu}^{ZV}\right)$ ;
$\hat{\theta}_k = \bar{\theta}(s_k)$ ;
$k \leftarrow k + 1$ ;
**fin ;**
**retourner** $\theta_N$ ;

---

Dans les conditions de cet exemple didactique, on va vérifier que la variable transformée $\tilde{X} = X - c(\phi(X) - \mathbb{E}[\phi(X)])$ permet de réduire presque complètement la variance. Pour ce faire, on emploie l'algorithme 4.2.1 sur 1000 jeux de données de taille $10^4$ afin d'obtenir 1000 estimateurs différents. La valeur de $\{\gamma_t\}$ est $\frac{0{,}51}{t^{0{,}51}}$. La valeur initiale de $s_0$ est obtenue en inversant le premier estimateur de $\theta_0$ obtenu à partir de l'estimateur du maximum de vraisemblance basé sur 10 observations complètes dont on aurait connaissance. Comme on le constate au tableau 4.2, les rapports entre l'algorithme EM en ligne Monte-Carlo à variance réduite et l'algorithme EM en ligne régulier sont près de 1, mais inférieurs à 1 entre l'algorithme EM en ligne Monte-Carlo 4.2.1 à variance réduite à 10 réalisations et l'algorithme 2.5.1 EM en ligne Monte-Carlo à 10 réalisations. Cela veut dire que la variance est la même que pour l'algorithme 2.4.1 et inférieure à celle de l'algorithme 2.5.1 à 10 réalisations Monte-Carlo. La figure 4.5 renchérit ce constat, car la courbe de l'algorithme en ligne et la courbe de l'algorithme de variance réduite (intitulé MC 10 ZV sur la légende) sont pratiquement superposées. Au niveau des temps computationnels par $N = 10^4$ itérations disponibles

au tableau 4.3, l'algorithme 2.5.1 sans réduction de variance est plus rapide que celui avec réduction de variance pour 10 réalisations. Toutefois, l'algorithme à variance réduite avec 10 réalisations est moins lourd computationnellement que l'algorithme 2.5.1 à 1000 réalisations. Il est donc plus intéressant d'opter pour la réduction de variance que de prendre une très grande taille d'échantillon Monte-Carlo.

| | En ligne | MC 10 | MC 100 | MC 1000 |
|---|---|---|---|---|
| $\beta_0$ | 1,01 | 0,84 | 1,00 | 1,01 |
| $\beta_1$ | 1,00 | 0,88 | 1,03 | 1,00 |
| $\beta_2$ | 1,02 | 0,93 | 1,01 | 0,98 |

**Tableau 4.2.** Rapport des écarts absolus médians au carré entre l'algorithme 4.2.1 et les algorithmes sans réduction de variance respectifs 2.4.1 et 2.5.1. À l'aide d'une variable de contrôle, il est possible de revenir à la variance de l'algorithme EM en ligne. En effet, le rapport des variances est près de 1 entre l'algorithme EM en ligne Monte-Carlo à variance réduite et l'algorithme EM en ligne Monte-Carlo. Le rapport entre l'algorithme EM en ligne Monte-Carlo 10 à variance réduite et l'algorithme EM en ligne Monte-Carlo 10, inférieur à 1, témoigne de l'effet de la variable de contrôle qui a permis de réduire la variance.

| | MC 1 | MC 10 | MC 100 | MC 1000 | MC ZV 10 |
|---|---|---|---|---|---|
| Temps (s) | 0,99 | 1,06 | 1,13 | 1,63 | 1,31 |

**Tableau 4.3.** Temps computationnels moyens en secondes pour les différents algorithmes. Les temps computationnels par $N = 10^4$ itérations augmentent en fonction du nombre de réalisations Monte-Carlo. Les techniques de réduction de variance exigent plus du temps pour un nombre égal de réalisations. L'algorithme EM en ligne Monte-Carlo à réduction de variance à 10 réalisations est cependant plus rapide que l'algorithme en ligne Monte-Carlo sans réduction de variance à 1000 réalisations. Ceci met en valeur l'alternative de la réduction de variance. Il vaut la peine d'appliquer une méthode de réduction de variance plutôt que de choisir une très grande taille d'échantillon Monte-Carlo.

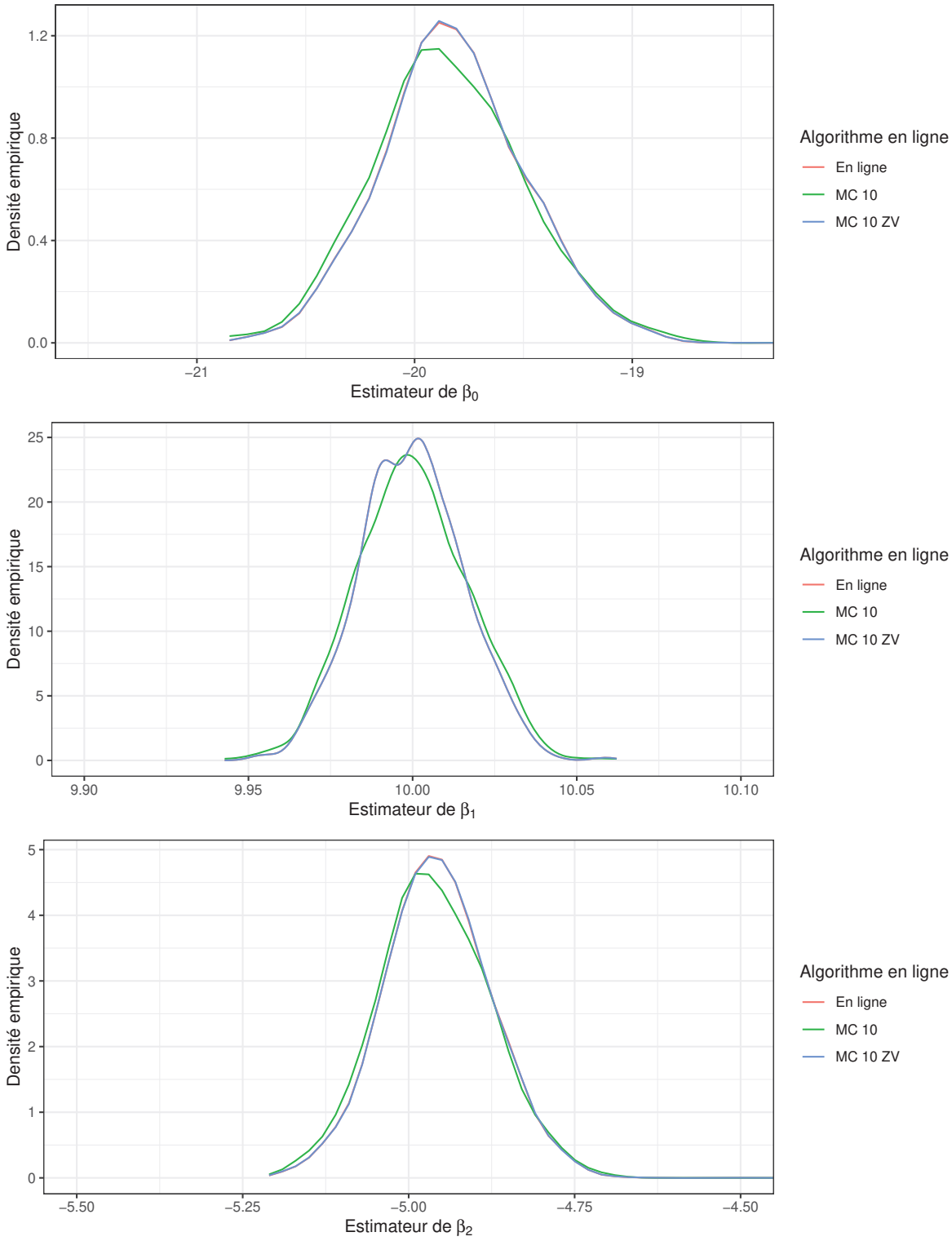


**Figure 4.5.** La densité empirique des estimateurs de $\beta_0$, $\beta_1$, $\beta_2$ dans le cas de l'algorithme 2.5.1 avec une variable de contrôle pour l'exemple de la variable latente continue. L'algorithme EM en ligne Monte-Carlo à variance réduite 4.2.1 construit des estimateurs qui ont la même distribution que l'algorithme EM en ligne. Autrement dit, on revient à l'algorithme EM en ligne.

## 4.3. Réduction de variance dans l'algorithme 2.6.1

Les exemples du chapitre 3 mettent en évidence l'augmentation de variance causée par l'estimation de $\mathbb{E}_{\tilde{\theta}(s)}\left[S(X,Y)|\, Y\right]$ par l'algorithme en ligne MCMC 2.6.1. On combine à présent la théorie de réduction de variance de Mira *et al.* (2013) aux approximations de l'algorithme EM en ligne par approximation stochastique pour créer un nouvel algorithme. Celui-ci a comme objectif d'avoir une performance comparable à l'algorithme EM en ligne où l'espérance a posteriori est calculée. Il suffit de connaître la densité $f$ proportionnelle à une constante de normalisation près de la variable que l'on veut générer et de connaître la dérivée de celle-ci. Le polynôme $P$ est optimisé à chaque itération. En pratique, il existe un package compatible avec progiciel R intitulé « ZVCV » qui permet d'obtenir l'estimateur zéro-variance en optimisant le polynôme. C'est ce qu'on a utilisé pour l'exemple de simulation.

Afin d'utiliser cette fonction, il faut spécifier les échantillons MCMC sans réduction de variance et la fonction d'intérêt qu'on écrit en fonction des échantillons. Finalement, il faut spécifier le gradient de la densité cible évalué aux points de la chaîne. Les autres arguments optionnels prennent des valeurs par défaut.

L'algorithme suivant ressemble beaucoup à l'algorithme EM en ligne MCMC 2.6.1. La différence est au niveau de la chaîne de Markov générée qui est modifiée afin d'obtenir des réalisations à variance réduite. Comme c'était le cas pour les algorithmes EM en ligne précédents, on initialise $s_0$ quelconque duquel on trouve $\hat{\theta}_0$. On connaît $Y_1$ qui est rendue disponible. On obtient des réalisations MCMC $X_{1,1}, \ldots, X_{1,m}$ à partir de l'algorithme de Metropolis-Hastings, car connaît la loi a posteriori de $X_1$ à une constante de normalisation près. À l'aide celles-ci, on emploie la méthode de Mira *et al.* (2013) pour obtenir les réalisations $\tilde{X}_{1,1}, \ldots, \tilde{X}_{1,m}$ à variance réduite. On peut ensuite obtenir un estimateur $\hat{\mu}^{ZV} = \frac{1}{m-B}\sum_{i=B+1}^{m} S(\tilde{X}_{k,i}, Y_k)$ de l'espérance a posteriori de la statistique exhaustive. L'étape d'appproximation stochastique est la même que pour les autres algorithmes EM en ligne, sauf que l'on insère l'estimateur $\hat{\mu}^{ZV}$ à la place de l'espérance a posteriori. À l'aide de $s_1$ obtenu à l'étape précédente, on trouve trouve $\hat{\theta}_1$. On répète le processus jusqu'à l'itération $N$.

**Algorithme 4.3.1.** L'algorithme EM par approximation stochastique en ligne MCMC zéro-variance.

---

**Entrées** Étant donné une suite décroissante positive $\{\gamma_t\}$, $B, N, m, \in \mathbb{N}$ et $s_0 \in \mathcal{S}$ ;
**Initialisation ;**
$\hat{\theta}_0 = \bar{\theta}(s_0)$ ;
$k := 1$ ;
**Tant que** $k < N + 1$ **:**
*Générer* $Y_k \sim \pi$ ;
*Générer* $X_{k,1}, \ldots, X_{k,m}$ à *l'aide de l'algorithme de Metropolis − Hastings* ;
Calculer $\tilde{X}_{k,1}, \ldots \tilde{X}_{k,m}$
où $\tilde{X}_{k,m} = X_{k,m} - \frac{1}{2}\Delta P(X_{k,m}) + \nabla P(X_{k,m}) \cdot \left(-\frac{1}{2}\nabla \log\left(f\left(X_{k,m}\right)\right)\right)$ ;
*Calculer* $\hat{\mu}^{ZV} = \frac{1}{m-B}\sum_{i=B+1}^{m} S(\tilde{X}_{k,i}, Y_k)$ ;
*Calculer* $s_k = s_{k-1} - \gamma_k \left(s_{k-1} - \hat{\mu}^{ZV}\right)$ ;
$\hat{\theta}_k = \bar{\theta}(s_k)$ ;
$k \leftarrow k + 1$ ;
**fin ;**
**retourner** $\theta_N$ **;**

---

Les estimateurs MCMC sont moins stables puisque l'estimateur n'est pas sans biais et a potentiellement une plus grande variance. Les techniques de réduction de variance pourraient permettre de réduire la variance causée par l'estimation de l'espérance a posteriori de la variable latente. La loi a posteriori de la forme $p(x; y,\theta) = \frac{f(x,y;\theta)}{\int_0^\infty f(x,y;\theta)\nu(\mathrm{d}y)}$ peut être échantillonnée par MCMC. L'espérance sous la distribution a posteriori sera donc remplacée par un estimateur MCMC. Étant donné que l'espérance a posteriori exacte de la variable latente n'est pas disponible, on ne peut pas implémenter l'algorithme EM en ligne habituel. Ainsi, l'augmentation de variance causée par l'usage de la méthode MCMC n'est pas connue. Cela signifie qu'il est impossible de quantifier l'effet de la méthode de réduction de variance. Elle pourrait avoir comme effet de donner la même variance qu'on aurait obtenue avec l'espérance exacte ; ou bien quelque part entre les deux. Pour remédier à cette situation, on implémente l'algorithme EM en ligne MCMC à 500 réalisations dont les estimateurs servent de valeurs de référence. Il est aussi possible de comparer la distribution, le biais et la variance des estimateurs fournis par l'algorithme 2.6.1 (EM en ligne MCMC sans réduction de variance) et l'algorithme 4.3.1 (EM en ligne MCMC avec réduction de variance). On peut maintenant évaluer la performance de l'algorithme contenant une composante de réduction de variance.

Pour l'exemple de modèle de régression avec une variable latente de Weibull, présenté à la section 3.4.3, on prend 1000 réplicats distincts de $10^4$ données, c'est-à-dire qu'on applique 1000 fois l'algorithme 4.3.1 sur $10^4$ données. La valeur de $\{\gamma_t\}$ est $\frac{0{,}51}{t^{0{,}51}}$. La valeur initiale

de $s_0$ est obtenue en inversant le premier estimateur de $\theta_0$ obtenu à partir de l'estimateur du maximum de vraisemblance basé sur 10 observations complètes dont on aurait connaissance. La loi $\pi$ n'étant pas connue, il faut un estimateur sans biais de $\mathbb{E}_\pi[\mathbb{E}_{\bar{\theta}(s)}[S(X,Y)|Y]]$, $\mathbb{E}_{\bar{\theta}(s)}[S(X,Y)|Y=y]$, par exemple. Lorsqu'on dispose de réalisations provenant d'un algorithme MCMC, il est possible d'estimer l'espérance a posteriori $\mathbb{E}_{\bar{\theta}(s)}[S(X,Y)|Y=y]$ dont on a besoin afin de pouvoir implémenter l'algorithme EM en ligne par approximation stochastique.

On tente de réduire la variance pour les estimateurs basés sur 50 et 100 réalisations MCMC. Lorsque les estimateurs MCMC sont remplacés par les estimateurs de Mira *et al.* (2013), on obtient une réduction de variance de l'ordre de ce que l'on rapporte dans le tableau 4.4. Pour des échantillons MCMC basés sur 50 et 100 réalisations, l'efficacité relative de l'algorithme 4.3.1 varie entre 0,85 et 0,95. Les rapports de variance par rapport à l'algorithme 2.6.1 avec 500 réalisations sont un peu inférieurs à 1, comme on le voit à la figure 4.5. On suppose que l'algorithme 2.6.1 avec 500 réalisations MCMC s'approche de ce qu'on aurait avec l'algorithme EM en ligne 2.4.1, ce qui servira de référence pour calculer le rapport des écarts médians. On voit également que l'algorithme de réduction de variance à 50 réalisations MCMC a une variance fortement semblable à celle à 100 réalisations. De plus, il est moins lourd en temps computationnel tel qu'indiqué au tableau 4.6. La figure 4.6 montre que les courbes de densité empirique associées à la réduction de variance sont presque superposées malgré la différence du nombre de réalisations. La réduction de variance élève légèrement le mode par rapport aux densités associées aux algorithmes qui ne mettent pas en oeuvre des techniques de réduction de variance. La figure 4.6 suggère que les courbes de densité associées aux estimateurs construits par les algorithmes à variance réduite sont presque superposées. Il semblerait plus avantageux en matière d'efficacité computationnelle de choisir $m = 50$ que $m = 100$. Le tableau 4.6 donne une approximation du temps computationnel par itération. L'algorithme 4.3.1 (de réduction de variance) pour 50 et 100 est certes plus lourd en temps computationnel que l'algorithme 2.6.1. On voit cependant qu'il est plus judicieux de recourir à une approche de réduction de variance que de choisir un très grand nombre de réalisations MCMC, par exemple 500. L'enjeu plus général consiste à trouver un juste milieu entre efficacité computationnelle et efficacité relative des estimateurs.

| | Rapport des variances | | Rapport des écarts absolus médians au carré | |
|---|---|---|---|---|
| | MCMC50 | MCMC100 | MCMC50 | MCMC100 |
| $\beta_0$ | 0,89 | 0,94 | 0,96 | 0,92 |
| $\beta_1$ | 0,88 | 0,92 | 0,92 | 0,89 |
| $\beta_2$ | 0,91 | 0,94 | 0,87 | 0,88 |

**Tableau 4.4.** Rapport des variances et des écarts absolus médians au carré entre l'algorithme 4.3.1 de réduction de variance avec 50 ou 100 réalisations et l'algorithme 2.6.1 avec 50 ou 100 réalisations. Les rapports de variance entre l'algorithme EM en ligne MCMC avec réduction de variance et l'algorithme EM en ligne MCMC à réduction de variance sont inférieurs 1 par quelques ou plusieurs points de pourcentage tout dépendant du paramètre estimé. Ainsi, la réduction de variance est avantageuse en termes de variance pour un algorithme ayant le même nombre de réalisations MCMC.

| | Rapport des variances | | Rapport des écarts absolus médians au carré | |
|---|---|---|---|---|
| | MCMC 50 | MCMC 100 | MCMC 50 | MCMC 100 |
| $\beta_0$ | 0,98 | 0,98 | 0,96 | 0,96 |
| $\beta_1$ | 0,98 | 0,98 | 0,97 | 0,97 |
| $\beta_2$ | 0,98 | 0,98 | 0,95 | 0,92 |

**Tableau 4.5.** Rapport des variances et des écarts absolus médians au carré entre l'algorithme 4.3.1 de réduction de variance à 50 ou 100 réalisations et l'algorithme en ligne ou l'algorithme de référence 2.6.1 avec 500 réalisations MCMC. L'algorithme EM en ligne n'est pas applicable à cet exemple complexe. L'algorithme EM en ligne MCMC, quant à lui, est applicable. En choisissant un nombre de réalisations de grande taille, on suppose que l'on approche l'algorithme EM en ligne. On voit que l'algorithme EM en ligne à variance réduite est très comparable, le rapport de variable étant environ 0,98 à l'algorithme EM en ligne MCMC.

| | MCMC 50 | MCMC 100 | MCMC 500 | MCMC ZV 50 | MCMC ZV 100 |
|---|---|---|---|---|---|
| Temps (s) | 12,6 | 18,7 | 77,0 | 28,6 | 36,2 |

**Tableau 4.6.** Temps computationnels moyens en secondes pour les différents algorithmes. Les temps computationnels par $10^4$ itérations augmentent en fonction du nombre de réalisations Monte-Carlo ou MCMC. Les techniques de réduction de variance exigent plus du temps. Ils sont cependant plus rapides que l'algorithme en ligne sans réduction de variance à 500 réalisations. Ceci met en valeur l'alternative de la réduction de variance. Il vaut la peine d'appliquer une méthode de réduction de variance plutôt que de faire augmenter sans cesse la longeur de la chaîne de Markov.

## 4.4. Impacts de la réduction de variance

Il est toujours possible de réduire la variance des estimateurs EM en ligne en construisant des estimateurs de l'espérance a posteriori basés sur plus d'échantillons iid ou des chaînes de Markov plus longues. Ceci entraîne un coût computationnel pour l'algorithme, qui peut être important, en particulier quand une méthode MCMC est utilisée. Réduire la variance de

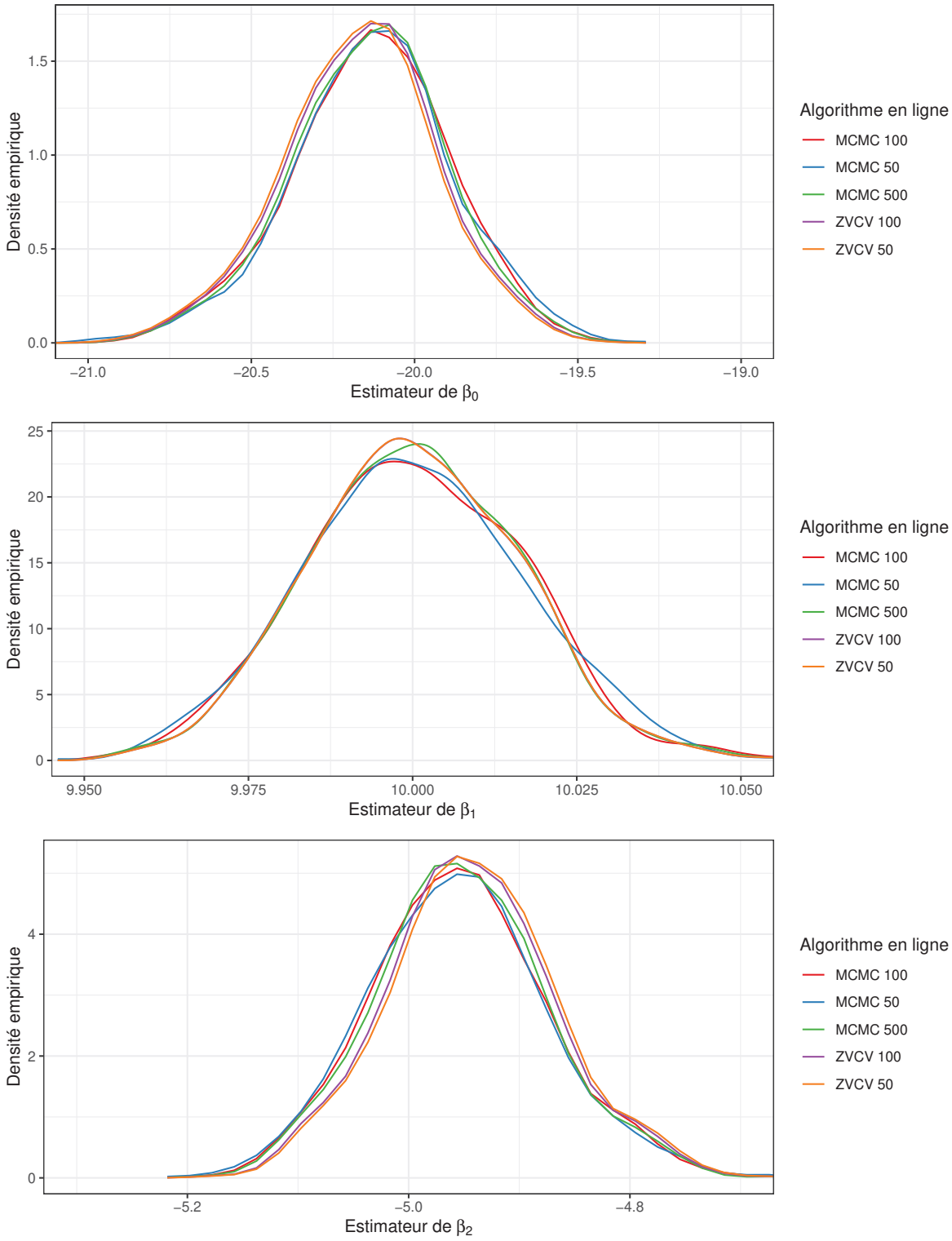


**Figure 4.6.** Densité empirique des estimateurs de $\beta_0$, $\beta_1$, $\beta_2$ dans le cas de l'algorithme 4.3.1 en ligne pour l'exemple de la variable latente continue suivant une Weibull. Les courbes orange et mauve correspondent aux densités empiriques des algorithmes de réduction de variance. Leurs modes sont plus élevés, ce qui suggère que les distributions de leurs estimateurs sont concentrées autour de ces points. La courbe verte correspond au cas où le nombre de réalisations MCMC est plus élevé. À l'exception de la figure du milieu, la distribution associée à la courbe verte semble plus variable que les algorithmes de réduction de variance reposant sur moins de réalisations MCMC.

façon plus astucieuse est possible dans certains cas en considérant des variables de contrôle. On observe que pour des échantillons iid Monte-Carlo, une simple variable de contrôle permet de trouver presque exactement les estimateurs EM en ligne sans approximation.

On observe que l'algorithme EM en ligne MCMC à variance réduite à 100 réalisations est plus précis que l'algorithme EM en ligne MCMC à 500 réalisations (qui fait office de valeur de référence, car les algorithmes EM et EM en ligne ne peuvent pas être implémentés dans ce contexte). Or, l'algorithme EM en ligne MCMC à 500 réalisations a un coût computationnel beaucoup plus grand que l'algorithme EM en ligne MCMC à 100 réalisations, mais aussi que l'algorithme EM en ligne MCMC 100 à variance réduite (plus du double).

Cette expérience montre qu'en contexte réaliste, c'est-à-dire lorsque l'estimateur du maximum de vraisemblance n'est pas calculable directement, lorsque l'algorithme EM en ligne n'est pas implémentable à cause de l'espérance a posteriori ou lorsqu'une transition MCMC est lente (car la loi a posteriori est compliquée et lente à évaluer computationnellement), l'algorithme EM en ligne MCMC est implémentable pour tous, mais nécessairement computationnellement lent pour le dernier. Cependant, on peut avoir la même efficacité que l'algorithme EM en ligne MCMC lorsqu'on introduit une variable de contrôle, c'est-à-dire lorsqu'on utilise l'algorithme EM en ligne MCMC à variance réduite, mais la complexité computationnelle sera plus faible quand l'algorithme EM en ligne MCMC utilise une longue chaîne de Markov. Bien sûr, ce constat repose sur le fait que la variable de contrôle soit relativement simple à calculer computationnellement.

# Discussion et conclusion

L'objectif de ce mémoire était d'étudier l'efficacité des algorithmes EM en ligne dans les contextes de données massives et en présence de modèles statistiques complexes. Les algorithmes en ligne sont avantageux puisqu'ils peuvent traiter les données lorsque celles-ci deviennent disponibles séquentiellement et ne sont pas gardées en mémoire par la suite. Ceci offre un avantage majeur par rapport à l'algorithme EM traditionnel qui nécessite d'avoir toutes les données disponibles en tout au long de l'apprentissage.

Dans les sections précédentes, on a fait l'étude de différents algorithmes. La raison d'être, les forces et faiblesses de ceux-ci ont été évoquées. Au chapitre 2, on a présenté brièvement différentes variantes de l'algorithme EM. De plus, on a mis en lumière comment la technique d'approximation de Robbins-Monro sous-tend tous ces algorithmes et occupe donc une place centrale de l'analyse de ceux-ci. Au chapitre 3, on a présenté quelques exemples d'application de l'algorithme EM en ligne où on observe que le fait de remplacer l'espérance a posteriori exacte par une approximation engendre une augmentation de la variance des estimateurs des paramètres dans les problèmes à variables manquantes. Au chapitre 4, on a montré que le problème de variance subsiste lorsque la variable latente est continue dans un modèle de régression, problème qui est largement corrigé par l'augmentation de la taille de l'échantillon Monte-Carlo. L'usage judicieux d'une variable de contrôle permet de réduire la variance lorsqu'on fait appel à un estimateur de Monte-Carlo à l'étape d'espérance dans des modèles simples ou à un estimateur MCMC dans des modèles à variables latentes continues plus complexes. Les problèmes précédents de mélange et de régression à variable latente normale reposaient sur des modèles statistiques relativement simples dans la mesure où il est aisé d'obtenir explicitement la loi a posteriori et d'échantillonner directement celle-ci. L'exemple de modèle dont la loi a priori de la variable latente suit une loi de Weibull est qualifié de complexe dans le sens qu'il n'est ni possible d'obtenir directement l'espérance a posteriori ni possible d'échantillonner directement la loi a posteriori. Pour cet exemple, on a proposé un algorithme en ligne MCMC qui emploie les techniques de réduction de variance pour MCMC de Mira *et al.* (2013). De manière générale, prendre des échantillons iid ou prendre une chaîne de Markov générée par MCMC confèrent une légère diminution de variance par rapport aux estimateurs de l'algorithme EM en ligne Monte-Carlo et MCMC, respectivement.

La question subsidiaire est de savoir combien de réalisations MC ou MCMC sont nécessaires afin de revenir à la distribution des estimateurs provenant de l'algorithme en ligne de Cappé et Moulines (2009). L'algorithme en ligne à variance réduite MC à 10 réalisations a un temps computationnel inférieur par itération que l'algorithme EM en ligne MC à 1000 réalisations. L'algorithme à variance réduite à 100 réalisations MCMC a un temps computationnel inférieur par itération à celui l'algorithme en ligne MCMC à 500 itérations. Les algorithmes à variance réduite ont des variances pratiquement équivalentes aux algorithmes sans réduction de variance reposant sur un grand nombre d'échantillons iid ou sur une longue chaîne de Markov comme le témoigne les tableaux 4.2 et 4.4. Les estimateurs de chaque paramètre ont un rapport de variance près de 1, ce qui rend la réduction de variance intéressante d'un point de vue computationnel.

L'emploi de méthodes de MCMC dans les contextes d'approximation stochastique occasionne des questions théoriques supplémentaires. Les algorithmes d'approximation stochastique exigent des estimateurs sans biais. L'estimateur MCMC qu'on a utilisé n'est pas sans biais, toutefois, sous des conditions relativement élémentaires concernant la chaîne de Markov sous jacente au MCMC, il est asymptotiquement sans biais quand la longueur de la chaîne augmente. Que se passe-t-il au niveau de la convergence dans un cas comme celui-là ? Ces considérations sont l'objet d'un papier récent de Tadić *et al.* (2017) dont l'objectif est de faire l'étude des propriétés asymptotiques d'un estimateur biaisé du gradient. L'estimateur biaisé entraîne une convergence qui ne se fait pas vers le minimum de la fonction à minimiser, mais vers un point dans le voisinage du minimum qui dépend du biais asymptotique (voir le théorème 2.1 de Tadić *et al.* (2017)). Cet exercice pourrait être repris pour un algorithme d'approximation stochastique plus général avec un estimateur MCMC. Il faudrait que les mêmes hypothèses et conditions évoquées dans le papier de Tadić *et al.* (2017) soient remplies, ce que l'on n'a pas vérifié. Une autre difficulté d'application de cette approche à notre situation est que la fonction $M$ qui caractérise l'approximation stochastique 3 est le gradient de la fonction à minimiser alors que pour les algorithmes EM en ligne, on a $M(s) = \mathbb{E}_{\pi}[\mathbb{E}_{\bar{\theta}(s)}[S(X,Y)|Y]] - s$. On peut toutefois observer que les résultats contenus dans ce mémoire relatifs à l'algorithme EM en ligne avec MCMC ne sont pas en contradiction avec ce que Tadić *et al.* (2017) ont montré.

La convergence non-asymptotique de l'algorithme EM en ligne pour des modèles comportant une constante de régularisation est étudiée dans Karimi *et al.* (2019b). Dans cette situation, la convergence est établie sous certaines conditions générales pour le cas où la collection $\{\xi_t\}$ est iid et dans le cas où $\{\xi_t\}$ est une chaîne de Markov qui caractérise le bruit de l'approximation stochastique. Une borne sur $\mathbb{E}[\|M(s_t)\|]$ est donnée au théorème 1 de Karimi *et al.* (2019b). Cependant, cela ne correspond pas à la situation de l'algorithme EM en ligne MCMC 2.6.1 étant donné que le biais est situé au niveau de l'étape d'espérance plutôt qu'à l'étape de maximisation. La chaîne $\{\xi_t\}$ ne converge pas vers une

loi stationnaire dans l'algorithme EM en ligne MCMC 2.6.1. En effet, la loi de $\xi_k$ ne dépend pas de $\xi_{k-1}$ par le biais d'un noyau explicite. La loi de $\xi_k$ est celle de la chaîne de Markov de loi stationnaire $p_{\theta_k}(\cdot; Y_k)$ c'est-à-dire que $\xi_k$ est une collection de variables qui dépend de $\xi_{k-1}$ uniquement à travers $\theta_k$. Plus précisément, $\xi_k$ est une chaîne de Markov de loi $(\xi_{k,1}, \ldots, \xi_{k,m}) \sim \int \pi(\mathrm{d}y) \int q_0(\mathrm{d}\xi_{k,0}) K_{\theta_k,y}(\xi_{k,0}, \mathrm{d}\xi_{k,1}) \ldots K_{\theta_k,y}(\xi_{k,m-1}, \mathrm{d}\xi_{k,m})$ où $K_{\theta,y}$ est le noyau de la chaine Markov ayant $p(x; y, \theta)$ pour loi stationnaire et $q_0$ est la loi initiale de la chaîne en question et $m$ la taille de la chaîne. On voit donc que la chaîne de Markov $\xi_k$ dépend de $\xi_{k-1}$ à travers $\theta_k$. La situation est donc beaucoup plus complexe que celle de Karimi.

La robustesse n'est pas étudiée dans ce mémoire. Est-ce que les algorithmes en ligne tolèrent des valeurs aberrantes de $Y$ ? En théorie, oui, puisque le modèle $g_\theta$ n'est pas forcément équivalent à loi de Y, $\pi$. En pratique, on peut supposer que le pas $\{\gamma_t\}$ pourrait occuper un rôle important dans la mesure où cette suite décroît quand $t$ augmente. Par conséquent, l'influence d'une valeur aberrante ou d'une valeur dans les ailes de $\pi$ diminue lorsqu'elle survient quand $t$ est grand. Une méthode *ad hoc* qu'on a employée, suggérée dans Cappé et Moulines (2009), est d'attendre un certain nombre d'itérations avant de mettre à jour les estimateurs des paramètres. Cela permet d'assurer que $s \in \mathcal{S}$, ce qui est une condition nécessaire pour la convergence de l'algorithme EM en ligne. Toutefois, cette heuristique n'est pas forcément suffisante pour assurer que $s \in \mathcal{S}$ dans l'algorithme EM en ligne MC et MCMC dans la mesure où, à cause de l'approximation de l'espérance, le vecteur de statistique exhaustive construit par l'algorithme Robbins-Monro pourrait ressortir de $\mathcal{S}$. En particulier, lorsqu'une donnée inhabituelle est traitée. Il faudrait donc réfléchir à rendre plus robuste l'algorithme EM en ligne lorsqu'une approximation de cette nature est utilisée. Une avenue potentielle serait de créer un mécanisme de rejet à partir d'estimateurs initiaux de la variance de la moyenne, ce qui permettrait d'écarter les valeurs très extrêmes de $y$ en fonction d'une règle. La première donnée traitée par l'algorithme est celle ayant le plus grand poids, la dernière donnée serait alors celle ayant le plus petit poids, de manière analogue à une fonction de Huber. Les données arrivent dans un ordre aléatoire. Bien sûr, en procédant de la sorte, on s'éloigne du cadre en ligne de l'algorithme initial vu qu'il faut avoir mis un poids sur toutes les données pour former le premier estimateur, donc avoir vu et conservé toutes les données.

# Références bibliographiques


Christophe Andrieu, Éric Moulines et Pierre Priouret : Stability of stochastic approximation under verifiable conditions. *SIAM Journal on Control and Optimization*, 44 (1):283–312, 2005. URL https://doi.org/10.1137/S0363012902417267.

Léon Bottou : Large-scale machine learning with stochastic gradient descent. *In Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

Stephen Boyd, Stephen P Boyd et Lieven Vandenberghe : *Convex optimization*. Cambridge university press, 2004.

Olivier Cappé : Online expectation-maximisation. *Mixtures : Estimation and applications*, pages 31–53, 2011.

Olivier Cappé et Eric Moulines : On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 71(3):593–613, 2009.

George Casella et Roger L Berger : *Statistical inference*, volume 2. Duxbury Pacific Grove, CA, 2002.

Jianfei Chen, Jun Zhu, Yee Whye Teh et Tong Zhang : Stochastic expectation maximization with variance reduction. *In Advances in Neural Information Processing Systems*, pages 7967–7977, 2018.

Bernard Delyon, Marc Lavielle et Eric Moulines : Convergence of a stochastic approximation version of the EM algorithm. *Annals of statistics*, pages 94–128, 1999.

A. P. Dempster, N. M. Laird et D. B. Rubin : Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. ISSN 00359246. URL http://www.jstor.org/stable/2984875.

T.S. Ferguson : *A Course in Large Sample Theory*. Chapman & Hall Texts in Statistical Science Series. Springer US, 1996. ISBN 9780412043710. URL https://books.google.ca/books?id=DDh_OiTw9agC.

W. K. Hastings : Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 04 1970. ISSN 0006-3444. URL https://doi.org/10.1093/biomet/57.1.97.

Rie Johnson et Tong Zhang : Accelerating stochastic gradient descent using predictive variance reduction. *In* C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani et K. Q. Weinberger, éditeurs : *Advances in Neural Information Processing Systems 26*, pages 315–323. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/4937-accelerating-stochastic-gradient-descent-using-predictive-variance-reduction.pdf.

Belhal Karimi, Marc Lavielle et Éric Moulines : On the convergence properties of the mini-batch EM and MCEM algorithms. 2019a.

Belhal Karimi, Blazej Miasojedow, Éric Moulines et Hoi-To Wai : Non-asymptotic analysis of biased stochastic approximation scheme. *arXiv preprint arXiv :1902.00629*, 2019b.

Estelle Kuhn et Marc Lavielle : Coupling a stochastic approximation version of EM with an MCMC procedure. *ESAIM : Probability and Statistics*, 8:115–131, 2004.

Estelle Kuhn, Catherine Matias et Tabea Rebafka : Properties of the stochastic approximation EM algorithm with mini-batch sampling. *arXiv preprint arXiv :1907.09164*, 2019.

Harold Kushner et G George Yin : *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media, 2003.

Kenneth Lange : A gradient algorithm locally equivalent to the EM algorithm. *Journal of the Royal Statistical Society : Series B (Methodological)*, 57(2):425–437, 1995.

Erich L Lehmann et George Casella : *Theory of point estimation*. Springer Science & Business Media, 2006.

Rémi Leluc et François Portier : Towards asymptotic optimality with conditioned stochastic gradient descent. *arXiv preprint arXiv :2006.02745*, 2020.

Mary J Lindstrom et Douglas M Bates : Newton—raphson and em algorithms for linear mixed-effects models for repeated-measures data. *Journal of the American Statistical Association*, 83(404):1014–1022, 1988.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller et Edward Teller : Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953. URL https://doi.org/10.1063/1.1699114.

S.P. Meyn et R.L. Tweedie : *Markov Chains and Stochastic Stability*. Springer-Verlag, London, 1993. URL /brokenurl#probability.ca/MT.

A Mira, P Tenconi et D Bressanini : Variance reduction for MCMC. *No. qf0310. Department of Economics, University of Insubria*, 2003.

Antonietta Mira, Reza Solgi et Daniele Imparato : Zero variance Markov chain Monte Carlo for bayesian estimators. *Statistics and Computing*, 23(5):653–662, 2013.

Radford M Neal et Geoffrey E Hinton : A view of the EM algorithm that justifies incremental, sparse, and other variants. *In Learning in graphical models*, pages 355–368. Springer, 1998.

Boris Polyak : New method of stochastic approximation type. *Automation and Remote Control*, 1990, 01 1990.

Herbert Robbins et Sutton Monro : A stochastic approximation method. *Ann. Math. Statist.*, 22(3):400–407, 09 1951. URL https://doi.org/10.1214/aoms/1177729586.

Christian Robert et George Casella : *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

Gareth O. Roberts et Jeffrey S. Rosenthal : General state space Markov chains and MCMC algorithms. *Probab. Surveys*, 1:20–71, 2004. URL https://doi.org/10.1214/154957804100000024.

Jeffrey S Rosenthal : Parallel computing and Monte Carlo algorithms. *Far east journal of theoretical statistics*, 4(2):207–236, 2000.

David Ruppert : Efficient estimations from a slowly convergent Robbins-Monro process. Rapport technique, Cornell University Operations Research and Industrial Engineering, 1988.

Vladislav B Tadić, Arnaud Doucet *et al.* : Asymptotic bias of stochastic gradient search. *The Annals of Applied Probability*, 27(6):3255–3304, 2017.

D Michael Titterington : Recursive parameter estimation using incomplete data. *Journal of the Royal Statistical Society : Series B (Methodological)*, 46(2):257–267, 1984.

M. S. Wolynetz : Algorithm as 139 : Maximum likelihood estimation in a linear model from confined and censored normal data. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(2):195–206, 1979. ISSN 00359254, 14679876. URL http://www.jstor.org/stable/2346749.

CF Jeff Wu : On the convergence properties of the EM algorithm. *The Annals of statistics*, pages 95–103, 1983.

Sinan Yildirim, Sumeetpal S. Singh et Arnaud Doucet : An online expectation–maximization algorithm for changepoint models. *Journal of Computational and Graphical Statistics*, 22(4):906–926, 2013. URL https://doi.org/10.1080/10618600.2012.674653.

# Annexe A

# Démonstrations

## A.1. Démonstrations de résultats concernant la convergence de l'algorithme 2.4.1

### A.1.1. Démonstration de la proposition 2.4.3

Démonstration. Soit $s^* \in \Gamma$, par définition :

$$\mathbb{E}_\pi \left[\bar{s}\left(Y,\theta\left(s^*\right)\right)\right] = s^*. \tag{51}$$

Notons que

$$\begin{aligned}
\nabla_\theta K(\pi,g_\theta) &= \nabla_\theta \mathbb{E}_\pi \left[\log \pi\left(Y\right)\right] - \nabla_\theta \mathbb{E}_\pi \left[\log g(Y;\theta)|\theta\right] && (52)\\
&= -\mathbb{E}_\pi \left[\mathbb{E}_\theta \left[\nabla_\theta \log f\left(X,Y;\theta\right)|Y\right]\right] && (53)\\
&= -\mathbb{E}_\pi \left[\mathbb{E}_\theta \left[\nabla_\theta \ell\left(S\left(X,Y\right),\theta\right)|Y\right]\right] && (54)\\
&= \nabla_\theta \phi(\theta) - \nabla_\theta \psi^T \mathbb{E}_\pi \left[\bar{s}\left(Y,\theta\right)\right] && (55)
\end{aligned}$$

où la seconde égalité découle de l'identité de Fisher et $\bar{s}(Y,\theta) = \mathbb{E}_\theta[s(X,Y)|Y]$. Ceci implique pour tout $s \in \mathcal{S}$

$$\nabla_\theta K(\pi,g_\theta)|_{\bar{\theta}(s)} = \nabla_\theta \phi(\bar{\theta}(s)) - \nabla_\theta \psi^T(\bar{\theta}(s))\, \mathbb{E}_\pi[s(Y,\bar{\theta}(s))]. \tag{56}$$

En particullier, pour $s^*$. À partir de 51, on a

$$\nabla_\theta K\left(\pi \| g_\theta\right)|_{\bar{\theta}(s^*)} = \nabla_\theta \phi\left(\bar{\theta}\left(s^*\right)\right) - \nabla_\theta \psi^T\left(\bar{\theta}\left(s^*\right)\right) s^*$$

Comme $\bar{\theta}$ est une solution de $\nabla_\theta \ell(s,\bar{\theta}(s)) = 0$, pour tout $s \in \mathcal{S}$, $s^*$ remplit aussi la condition. Ainsi, $s \in \Gamma \Rightarrow \theta^*$ est une solution stationnaire de $\nabla_\theta K(\pi \| g_\theta) = 0$.
$\Leftarrow$
Supposons que $\nabla_\theta K(\pi \| g_\theta) = 0$ et supposons qu'il existe $s_0 = \mathbb{E}_\pi[\bar{s}(Y,\bar{\theta}(s_0))] \in \mathcal{S}$ t.q $\bar{\theta}(s_0) = 0$. À l'aide de 52, on obtient

$$\nabla_\theta \phi\left(\bar{\theta}\left(s_0\right)\right) - \nabla_\theta \psi^T\left(\bar{\theta}\left(s_0\right)\right) \mathbb{E}_\pi\left[\bar{s}\left(Y,\bar{\theta}\left(s_0\right)\right)\right] = 0$$

Pour tout $s \in \mathcal{S}$, en particulier $s_0$, la fonction $\ell(s,\theta)$ possède un maximum unique, dans ce cas-ci en $\bar{\theta}(s_0)$, ce qui complète la preuve.

□

## A.1.2. Démonstration de la proposition 2.4.4

Démonstration. Notons que

$$\nabla_s w(s) = \nabla_s \bar{\theta}^T(s) \nabla_\theta K(\pi \| g_\theta)|_{\bar{\theta}(s)}, \tag{57}$$

où $\nabla_s \bar{\theta}(s) \in M_{p,d}$ avec $\{\nabla_s \bar{\theta}(s)\}_{i,j} = \frac{\partial \bar{\theta}_i}{\partial s_j(s)}$.
Ensuite, à partir de 56 et l'identité $\nabla_\theta \phi(\bar{\theta}(s)) = \nabla_\theta \psi^T(\bar{\theta}(s))$ qu'on obtient en posant $\nabla_\theta \ell(s,\bar{\theta}(s)) = 0$, on a :

$$\nabla_\theta K(\pi \| g_\theta)|_{\bar{\theta}(s)} = \nabla_\theta \psi^T\left(\bar{\theta}\left(s\right)\right)\left\{s - \mathbb{E}_\pi\left[s\left(Y,\bar{\theta}\left(s\right)\right)\right]\right\} = -\nabla_\theta \psi^T\left(\bar{\theta}\left(s\right)\right) M\left(s\right). \tag{58}$$

À présent, définissons les fonctions $\Phi : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^p$ avec $\Phi(s,\theta) = \nabla_\theta \ell(s,\theta) = \nabla_\theta[\phi(\theta) - \psi^T(\theta)s]$ et $u : \mathbb{R}^d \to \mathbb{R}^d \times \mathbb{R}^p$ avec $u(s) = [s,\bar{\theta}(s)]$. Le gradient partiel par rapport à $s$ de $\Phi^T$ est une matrice $\nabla_s \Phi^T \in M_{d,p}$ avec

$$\{\nabla_s \Phi^T\}_{i,j} = \frac{\partial \Phi_j}{\partial s_i} = \frac{\partial^2 \ell}{\partial s_i \partial \theta_j}.$$

Notons que

$$\begin{aligned}
\nabla_s \Phi^T \circ u(s) &= \nabla_s u(s) \nabla_\theta \Phi^T|_{u(s)} \\
&= [I_d, \nabla_s \bar{\theta}^T(s)][\nabla_s \Phi|_{u(s)}, \nabla_\theta \Phi|_{u(s)}]^T \\
&= \nabla_s \Phi|_{u(s)} + \nabla_s \bar{\theta}^T(s) \nabla_\theta \Phi|_{u(s)} \\
&= \left\{\nabla_\theta \psi^T\left(\bar{\theta}\left(s\right)\right)\right\}^T + \nabla_s \bar{\theta}^T\left(s\right) \nabla_\theta^2 \ell|_{u(s)}
\end{aligned}$$

La fonction $\Phi \circ u(s)$ est identiquement nulle. Son gradient par rapport à $s$ l'est aussi. On obtient :

$$\{\nabla_\theta \psi^T(\bar{\theta}(s)\}^T + \nabla_s \bar{\theta}^T(s) \nabla^2_\theta \ell|_{u(s)} = 0 \iff \nabla_s \bar{\theta}^T(s) = -\{\nabla_\theta \psi^T(\bar{\theta}(s))\}^T \{\nabla^2_\theta \ell|_{u(s)}\}^{-1}. \quad (59)$$

En combinant 57 et 59, on trouve :

$$M(s)^T \nabla_s w(s) = M(s)^T \{\nabla_\theta \psi^T(\bar{\theta}(s))\}^T \{\nabla^2_\theta l|_{u(s)}\}^{-1} \nabla_\theta \psi^T(\bar{\theta}(s)) M(s). \quad (60)$$

L'hypothèse 1 c dans Cappé et Moulines (2009), permet de déduire que $\{\nabla^2_\theta l|_{u(s)}\}$ est définie négative et inversible. Ainsi, pour tout $s \in \mathcal{S}$, $\langle \nabla_s w(s), M(s) \rangle < 0$.

Dans le cas où il y a égalité, soit $s \in \mathcal{S}$, on a que $\nabla_\theta \psi(\bar{\theta}(s)) M(s) = 0$ (ou sa transposée). Ceci entraîne que

$$\nabla_\theta \psi^T(\bar{\theta}(s)) s = \nabla_\theta \psi^T(\bar{\theta}(s))\, \mathbb{E}_\pi[\bar{s}(Y, \bar{\theta}(s))] \iff \nabla_\theta \phi(\bar{\theta}(s)) = \nabla_\theta \psi^T(\bar{\theta}(s))\, \mathbb{E}_\pi[\bar{s}(Y, \bar{\theta}(s))] \iff s \in \Gamma$$

ce qui achève la preuve.

□

### A.1.3. Démonstration du théorème 2.4.6

Démonstration. Il suffit d'appliquer le théorème 2.3 de Andrieu *et al.* (2005). Les hypothèses A1 de (Andrieu *et al.*, 2005) sont les suivantes :

Supposons qu'il existe $\Theta$, un sous-ensemble ouvert de $\mathbb{R}^m$, $M : \Theta \to \mathbb{R}^m$ est continue, et qu'il existe une fonction continue et différentiable $w : \Theta \to [0, \infty)$ t.q

(1) Il existe $O_0 > 0$ t.q

$$\mathcal{L} := \{\theta \in \Theta, \langle \nabla w(\theta), h(\theta) \rangle = 0\} \subset \{\theta \in \Theta, w(\theta) < O_0\} \quad (61)$$

(2) Il existe $O_1 \in (O_0, \infty]$ t.q $W_{O_1}$ est un ensemble compact,

(3) Pour tout $\theta \in \Theta\mathcal{L}, \langle \nabla w(\theta), M(\theta) \rangle < 0$,

(4) L'intérieur de la fermeture de $w(\mathcal{L})$ est vide.

Compte tenu des conditions évoquées ci-dessus, il suffit d'appliquer le théorème 2.3 dans Andrieu *et al.* (2005) à la séquence $\{s_n\}$.

□

## A.2. Démonstration de la proposition 1.6.2

$$\begin{aligned}
\mathbb{E}_f\left[\frac{H\phi}{\sqrt{f}}\right] &= \int_\Omega (H\phi)\sqrt{f} \\
&= \int_\Omega \left(V\phi - \frac{1}{2}\Delta\phi\right)\sqrt{f} \\
&= \int_\Omega \left(V\phi - \frac{1}{2}\Delta\phi\right)\sqrt{f} + \int_\Omega \left(H\sqrt{f}\right)\phi - \int_\Omega \left(V\sqrt{f} - \frac{1}{2}\Delta\sqrt{f}\right)\phi \\
&= -\int_\Omega \frac{1}{2}\Delta\phi\sqrt{f} + \int_\Omega \left(H\sqrt{f}\right)\phi + \int_\Omega \frac{1}{2}\Delta\sqrt{f}\phi \\
&= \int_\Omega \left(H\sqrt{f}\right)\phi - \int_\Omega \frac{1}{2}\Delta\phi\sqrt{f} + \int_\Omega \frac{1}{2}\Delta\sqrt{f}\phi \\
&= \int_\Omega \left(H\sqrt{f}\right)\phi - \frac{1}{2}\int_{\partial\Omega} \sqrt{f}\nabla\phi \cdot \mathbf{n}dS + \frac{1}{2}\int_{\partial\Omega} \phi\nabla\sqrt{f} \cdot \mathbf{n}dS \\
&= \int_\Omega \left(H\sqrt{f}\right)\phi + \frac{1}{2}\int_{\partial\Omega} \left(\phi\nabla\sqrt{f} - \sqrt{f}\nabla\phi\right) \cdot \mathbf{n}dS \\
&= \int_\Omega \left(V\sqrt{f} - \frac{1}{2}\Delta\sqrt{f}\right)\phi + \frac{1}{2}\int_{\partial\Omega} \left(\phi\nabla\sqrt{f} - \sqrt{f}\nabla\phi\right) \cdot \mathbf{n}dS \\
&= \int_\Omega \left(\frac{1}{2\sqrt{f}}(\Delta\sqrt{f})\sqrt{f} - \frac{1}{2}\Delta\sqrt{f}\right)\phi + \frac{1}{2}\int_{\partial\Omega} \left(\phi\nabla\sqrt{f} - \sqrt{f}\nabla\phi\right) \cdot \mathbf{n}dS \\
&= \frac{1}{2}\int_{\partial\Omega} \left(\phi\nabla\sqrt{f} - \sqrt{f}\nabla\phi\right) \cdot \mathbf{n}dS
\end{aligned}$$

Ainsi, $\mathbb{E}_f\left[\frac{H\phi}{\sqrt{f}}\right] = 0$ si $\phi\nabla\sqrt{f} = \sqrt{f}\nabla\phi$ sur $\partial\Omega$.

Dans le cas où $\phi = P\sqrt{f}$, les égalités

$$\nabla\phi = \sqrt{f}\nabla P + \frac{P}{2\sqrt{f}}\nabla f$$

et

$$\phi\nabla\sqrt{f} - \sqrt{f}\nabla\phi = fP,$$

entraînent que $\mathbb{E}_f\left[\frac{H\phi}{\sqrt{f}}\right] = 0$ si $f(x)\frac{\partial P(x)}{\partial x_j} = 0$, $\forall x \in \partial\Omega$, $j = 1,2,\ldots,d$.

## A.3. Démonstrations concernant les estimateurs du maximum de vraisemblance

L'estimateur du maximum de vraisemblance du paramètre $\omega_i$ est trouvé dans le mélange de normales, mais le raisonnement est très similaire pour l'estimateur dans le mélange de régression.

### A.3.1. Estimateur $\omega_i$, $\mu_i$ et $\sigma_i^2$

$$\begin{aligned}\ell_n(\theta; y,x) &= \sum_{i=1}^{2}\sum_{j=1}^{n} I(x_j = i)\log\left(\omega_i \frac{1}{(2\pi\sigma_i^2)^{\frac{n}{2}}}\exp\left\{-\sum_{j=1}^{n}\frac{1}{2\sigma_i^2}(y_j-\mu_i)^2\right\}\right)\\ &= \sum_{i=1}^{2}\sum_{j=1}^{n} I(x_j = i)\log\left(\omega_i\sqrt{2\pi\sigma_i^2}\right) - \sum_{i=1}^{2}\sum_{j=1}^{n} I(x_j = i)\frac{1}{2\sigma_i^2}\left(y_j-\mu_i\right)^2\end{aligned}$$

$$\frac{\partial}{\partial\omega_i}\ell_n(\theta; y,x) = \sum_{j=1}^{n} I(x_j = i)\frac{1}{\omega_i} - \sum_{j=1}^{n}\frac{1}{1-\omega_i}\left(1 - I\left(x_j = i\right)\right) \tag{62}$$

$$\frac{\partial}{\partial\omega_i}\ell(\theta; y,x) = 0 \tag{63}$$

$$\sum_{j=1}^{n} I(x_j = i)\frac{1}{\omega_i} = \sum_{j=1}^{n}\frac{1}{1-\omega_i}(1 - I(x_j = i)) \tag{64}$$

$$\sum_{j=1}^{n} I(x_j = i)\left(\frac{1}{\omega_i} - 1\right) = \sum_{j=1}^{n}\left(1 - I\left(x_j = i\right)\right) \tag{65}$$

$$\sum_{j=1}^{n} I(x_j = i)\left(\frac{1}{\omega_i}\right) = n \tag{66}$$

$$\omega_i = \frac{\sum_{j=1}^{n} I(x_j = i)}{n} \tag{67}$$

$$\begin{aligned}\frac{\partial}{\partial\mu_i}\ell(\theta; y,x) &= 0\\ -2\sum_{j=1}^{n} I(x_j = i)\left(y_j - \mu_i\right) &= 0\\ \sum_{j=1}^{n} I(x_j = i)y_j &= \sum_{j=1}^{n} I(x_j = i)\mu_i\\ \mu_i &= \frac{\sum_{j=1}^{n} I(x_j = i)y_j}{\sum_{j=1}^{n} I(x_j = i)}\end{aligned}$$

$$\frac{\partial}{\partial \sigma_i^2}\ell(\theta; y,x) = 0$$

$$\sum_{i=1}^{2}\sum_{j=1}^{n} I(x_j = i)\log\left(\omega_i\sqrt{2\pi\sigma_i^2}\right) - \sum_{i=1}^{2}\sum_{j=1}^{n} I(x_j = i)\frac{1}{2\sigma_i^2}\left(y_j - \mu_i\right)^2 = 0$$

$$\sum_{j=1}^{n} I(x_j = i)\frac{1}{(\sigma_i^2)} - \sum_{j=1}^{n} I(x_j = i)\frac{1}{\sigma_i^4}\left(y_j - \mu_i\right)^2 = 0$$

$$\sigma_i^2 = \frac{\sum_{j=1}^{n} I(x_j = i)\left(y_j - \mu_i\right)^2}{\sum_{j=1}^{n} I(x_j = i)}$$

$$= \frac{\sum_{j=1}^{n} I(x_j = i)y_j^2}{\sum_{j=1}^{n} I(x_j = i)} - \frac{\sum_{j=1}^{n} I(x_j = i)2y_j\mu_i - \sum_{j=1}^{n} I(x_j = i)\mu_i^2}{\sum_{j=1}^{n} I(x_j = i)}$$

$$= \frac{\sum_{j=1}^{n} I(x_j = i)y_j^2}{\sum_{j=1}^{n} I(x_j = i)} - \mu_i^2$$

### A.3.2. Estimateur de régression $\beta_i$

$$\ell_n(\beta,\omega_i; y,x) = \sum_{i=1}^{2}\sum_{j=1}^{n} I(x_j = i)\log\left(\omega_i\frac{1}{(2\pi\sigma_i^2)^{\frac{1}{2}}}\right) + \sum_{i=1}^{2}\sum_{j=1}^{n} I(u_j = i)\left\{-\frac{1}{2\sigma_i^2}(y_j - \mathbf{x_j}^T\beta_i)^2\right\}$$

$$\frac{\partial}{\partial \beta_i}\ell(\theta; y,x) = -I(x = i)\frac{1}{2\sigma_i^2}2\mathbf{X}'(\mathbf{Y} - \mathbf{X}\beta_i)$$

$$\frac{\partial}{\partial \beta_i}\ell(\theta; y,x) = 0$$

$$-I(x = i)\frac{1}{2\sigma_i^2}2\mathbf{X}'(\mathbf{Y} - \mathbf{X}\beta_i) = 0$$

$$\mathbf{X}'\mathbf{Y} = \mathbf{X}'\mathbf{X}\beta_i)$$

$$\beta_i = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$$