Lecture d'article

–

Efficient preconditioned stochastic gradient descent for estimation in latent variable models

Université Paris-Saclay

Novembre 2024

–

Résumé

Nous présentons dans ce rapport une relecture de l'article *Efficient preconditioned stochastic gradient descent for estimation in latent variable models* par Baey et al. (2023) [1].Cette publication met en avant une nouvelle façon d'optimiser les méthodes de descente de gradient pour l'estimation de paramètres dans des modèles à variables latentes. Nous verrons dans ce rapport la place de l'article dans la bibliographie et nous ferons une relecture de ses éléments théoriques et de ses implémentations avant de soumettre notre critique personelle.

1 Introduction

1.1 Présentation

L'article a été publié en 2023 par Charlotte Baey *(Université de Lille, CNRS)*, Maud Delattre, Estelle Kuhn *(Université Paris-Saclay, INRAE, MaIAGE)*, Jean-Benoist Leger *(Université de technologie de Compiègne, CNRS)* et Sarah Lemler *(Université Paris-Saclay, CentraleSupélec)*. Il s'intéresse à l'estimation de modèles paramétriques à variables latentes et propose une nouvelle stratégie d'optimisation pour les algorithmes de descente de gradient stochastique.

L'introduction présente l'historique des travaux qui ont pu être realisé pour l'estimation de ces modèles afin d'amener les enjeux et la problématique à laquelle les auteurs s'intéressent ici : nous avons d'une part les algorithmes de type EM (*Dempster 1977*[2]) et ses variantes SAEM, VEM (*Delyon et al., 1999*[3], *Bernardo et al., 2003*[4]) qui sont plus anciens mais toujours très fréquemment utilisés pour l'estimation des modèles à variables latentes. D'autre part, il existe également des algorithmes d'optimisation par descente de gradient stochastique qui, comme indiqué par les auteurs, ont l'avantage d'être plus simples à implémenter en pratique (puisque l'expression du gradient de la log-vraisemblance est généralement plus accessible et peut être calculé par différenciation automatique, tandis que l'expression du maximum de vraisemblance utilisé dans les algorithmes EM peut être difficile à établir pour certains modèles). Cependant leurs garanties théoriques de convergence et les choix pratiques du taux d'apprentissage sont encore peu présents dans la littérature (*Cai, 2010*[5] et *Fang&Li, 2021*[6]).

L'article présente ici une variante de l'algorithme de descente de gradient stochastique (SGD) qui permet d'accélérer l'optimisation du paramètre en réutilisant d'une part une estimation de l'information de Fisher que les auteurs ont étudié sur des travaux antérieurs *(Delattre & Kuhn [7])* et qui permet de conditionner l'étape de descente jusqu'à une approximation d'ordre 2 de la log-vraisemblance. D'autre part ils font ici évoluer le taux d'apprentissage γ sur trois phases en s'inspirant de travaux similaires réalisés dans d'autres contextes d'optimisation [8], [9].

1.2 Comparaisons par rapport aux autres méthodes

L'article met également en avant le fait que, contrairement aux algorithmes EM, les convergences par descente de gradient peuvent être assurées pour des modèles n'appartenant pas à la famille des modèles de courbe exponentielle. C'est notamment le cas pour les modèles à effets mixtes sur lesquels les auteurs réalisent leurs expériences et comparent les résultats obtenus à une implémentation spécifique de l'algorithme SAEM pour ce modèle (*Comets et al. 2017* [10], librairie R `saemix`).

Sous certaines hypothèses sur les modèles étudiés, les auteurs cherchent à fournir quelques arguments de convergence pour leur algorithme généralisables à un grand nombre de familles de modèles : *Cai 2010* [5] ne possède pas de preuve théorique concernant la convergence de l'algorithme de descente de gradient. Celui de *Fang&Li 2021* [11] possède un théorème de convergence mais seulement pour des modèles qui n'ont pas de paramètres à estimer dans les distributions des variables latentes. Cela l'empêche de fonctionner sur la plupart des modèles, et notamment pour ceux choisis par les auteurs du Fisher-SGD pour illustrer la performance de leurs algorithmes.

Il existe également d'autres algorithmes de descente de gradient stochastique (modèles latents hierarchiques, Hong. 2024 [12]) mais qui sont dédiés à des familles de modèles toujours très spécifiques.

2 Étude des arguments théoriques

2.1 Rappels sur les modèles à variables latentes

Les modèles à variables latentes sont des modèles sur lesquels on suppose qu'une variable observée Y_i peut être conditionné par un phénomène latent non observé Z_i. L'objectif est d'estimer la distribution $p_Y(y;\theta)$ paramétrisée par θ à partir des valeurs de y observées, en tenant compte du phénomène latent Z.

Quelques exemples de modèles à variables latentes

Gaussian mixtures :

Le modèle de mélange gaussien est défini par rapport à une variable latente Z que l'on suppose suivre une loi multinomiale de vecteur de probabilité π et tel que la densité conditionnelle de Y par rapport à la réalisation Z = k suit une loi normale centrée en μ_k et de variance σ_k^2 :

$$\begin{cases} Z_i \sim \mathcal{M}(1,\pi), \quad \sum\limits_{k=1}^{K} \pi_t = 1 \\ Y_i | Z_i = k \sim \mathcal{N}(\mu_k, \sigma_k^2) \end{cases} \tag{1}$$

Dans ce modèle, les paramètres à estimer sont les proportions π, et les paramètres des lois normales μ, σ^2. On réalisera notre propre implémentation et analyse du Fisher-SGD pour ces modèles en partie 4.

Mixed-effect models :

Les modèles *mixed-effect* permettent de généraliser des modèles linéaires où l'on suppose que plusieurs observations sont corrélées à la présence d'un individu i parmi les données :

$$\begin{cases} Z_i \sim \mathcal{N}(\beta, \Sigma) \\ Y_{ij}|Z_i \sim h(\alpha, Z_i, X_{ij} + \varepsilon_{ij}), \quad \varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2) \end{cases} \tag{2}$$

où, $Z_i, i \in \{1 \dots J\}$ désigne les effets latents liés à l'individu i et suit une loi gaussienne. Y_{ij} est la j-ième variable observée pour l'individu i, dont on suppose que la distribution conditionnelle à l'effet Z_i est une fonction de Z_i, d'une variable aléatoire explicative X_{ij}, d'effets fixes α et d'un bruit aléatoire gaussien ε_{ij}. Dans ce modèle, les paramètres à estimer sont α, σ^2, β et Σ.

SBM (Stochastic Block Models) :

Les modèles de blocs stochastiques reprennent les structures de graphe, où l'objectif est de regrouper des noeuds par similarité sur un graphe orienté de N noeuds. On suppose ainsi qu'il existe une classification inconnue des noeuds (variable latente Z) sur le graphe avec une distribution conditionnelle des arêtes dépendant uniquement des noeuds du cluster. En notant (Y_{ij}) la matrice d'adjacence du graphe, K le nombre de groupes et Z_i l'effet latent du i-ème individu, on pose :

$$\begin{cases} Z_i \sim \mathcal{M}(1, \alpha), \quad \sum_{k=1}^{K} \alpha = 1 \\ Y_{ij}|Z_{ik}Z_{jl} = 1 \sim \mathcal{B}(p_{kl}), \quad p_{kl} \in]0, 1[\end{cases} \tag{3}$$

2.2 Estimation par maximum de vraisemblance

On souhaite estimer θ en maximisant la vraisemblance sur la densité jointe du couple $f(y, z; \theta)$:

$$g(y; \theta) = \int_{\mathcal{Z}} f(y, z; \theta) dz$$

$$\hat{\theta} = \text{argmax}_\theta \, g_\theta(y)$$

Puisque g_θ n'est pas connu directement et que les variables latentes Z_i ne sont pas observées, l'algorithme EM (estimation/maximisation) procède en deux étapes : l'estimation a posteriori des données latentes z_i par règle de Bayes pour un paramètre θ_t donné puis l'optimisation par maximum de vraisemblance sur la loi du couple (Y, Z) pour obtenir un nouveau paramètre θ_{t+1}. Comme on le note sur l'article, l'algorithme est désormais relativement ancien (présenté pour la première fois par *Dempster et al., 1977* [2] puis sous différentes variantes : SAEM *Delyon et al., 1999* [3], VEM *Bernardo et al., 2003* [4]). Les auteurs se concentrent ici sur un procédé alternatif pour l'estimation de la densité en s'appuyant sur les travaux de *Cappé et al., 2005* [13]. L'algorithme consiste ici à passer par le gradient de la log-vraisemblance de g_θ et de réaliser une descente de gradient stochastique pour récupérer une estimation d'un maximum local du paramètre :

Algorithm 1: Algorithme de descente de gradient stochastique **(SGD)**

Data: $\gamma, \theta_0, y_1 ... y_N$
Result: Estimation $\hat{\theta}$
for $t \leftarrow 1$ **to** n_{iter} **do**
  $z_i^k = p_{\theta_{t-1}}(z_i|y_i)$ *(estimation a posteriori)*;
  $v_t = \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \log\, f(y_i, z_i^k; \theta_{t-1})$;
  $\theta_t = \theta_{t-1} + \gamma v_t$ *(étape de descente)*;
end

L'étude réutilise des travaux antérieurs *(Delattre & Kuhn., 2019)* [7] sur l'estimation de l'information de fisher. Celle-ci permet de conditionner l'étape de descente à l'ordre 2 sur la log-vraisemblance du modèle :

Algorithm 2: Conditionnement de l'étape de descente par information de fisher (Fisher-SGD)

Data: $\theta_0, y_1 ... y_N, \gamma_0, K_{ph}, K_h, \alpha$
Result: Estimation $\hat{\theta}$
for $t \leftarrow 1$ **to** n_{iter} **do**
  if $t < K_{ph}$ **then**
    $\gamma_t = \exp((1 - \frac{t}{K_{ph}}) \log\, \gamma_0)$;
  end
  else if $t < K_h$ **then**
    $\gamma_t = 1.0$;
  end
  else
    $\gamma_t = (t - K_h)^{-\alpha}$;
  end
  $z_i^k = p_{\theta_{t-1}}(z_i|y_i)$ *(estimation a posteriori)*;
  $\Delta_i^{(t)} = (1 - \gamma_t)\Delta_i^{(t-1)} + \gamma_t \nabla_\theta \log\, f(y_i, z_i^k; \theta_{t-1})$;
  $v_t = \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \log\, f(y_i, z_i^k; \theta_{t-1})$;
  $I_t = \frac{1}{N} \sum_{i=1}^{N} \Delta_i^t (\Delta_i^t)^T$ *(estimation de l'information de fisher)*;
  if $k < K_{ph}$ **then**
    $P_t = (1 - \gamma_t).max(1, Tr(I_t))Id + \gamma_t I_t$ *(pre-heating)*;
  end
  else
    $P_t = I_t$;
  end
  $\theta_t = \theta_{t-1} + \gamma_t P_t^{-1} v_t$ *(étape de descente)*;
end

En plus de cette estimation, le taux d'apprentissage γ_t évolue en trois phases durant l'optimisation :

— *pre-heating* : γ_t évolue de façon exponentielle pour atteindre exactement 1.0 à l'étape de *heating*
— *heating* : γ_t est maintenu à 1.0
— *decreasing step* : diminution progressive de γ_t en puissance $-\alpha$ pour un paramètre α à calibrer (les auteurs proposent 2/3 par défaut).

Cette séparation de phases vise à augmenter progressivement le conditionnement de l'algorithme (jusqu'à la phase de heating) en stabilisant l'estimation de fisher durant l'étape de pre-heating.

2.3 Estimation de l'information de fisher et propriété de convergence

L'élément principal utilsé pour l'optimisation de l'algorithme de descente de gradient stochastique est l'utilisation de l'information de fisher du modèle qui est définie par :

$$I(\theta) = \mathbb{E}[\frac{\partial}{\partial \theta^2} \log f(x, y; \theta)] = \mathbb{E}[(\nabla_\theta \log f(x, y; \theta))(\nabla_\theta \log f(x, y; \theta))^T]$$

Dans de précédents travaux *(Delattre & Kuhn, [7])*, les auteurs proposent d'estimer l'information de Fisher par approximation stochastique de manière similaire à ce qui peut être réalisé dans des approches de type EM ou sur l'algorithme de descente de gradient que l'on étudie ici : les données latentes z_i sont estimées a posteriori pour un paramètre θ donné. Puis l'information de Fisher est estimé par :

$$\hat{I}_t(\theta) = \frac{1}{n} \sum (\Delta_i^{(t)} (\Delta_i^{(t)})^T)$$

par un lissage progressif du gradient estimé :

$$\Delta_i^{(t)} = (1 - \gamma_t)\Delta_i^{(t-1)} + \gamma_t \nabla_\theta \log f(y_i, z_i; \theta_t)$$

Pour soutenir leur algorithme, les auteurs proposent un théorème qui donne une borne sur l'esperance du gradient minimal (en norme) de la log-vraisemblance au cours des itérations, qui tend vers 0 lorsque le nombre d'itérations tend vers $+\infty$. Le théorème est fondé sur les hypothèses suivantes :

(i) La densité jointe f de (Y, Z) est deux fois différentiable pour tout $\theta \in \Theta$

(ii) $\forall y \in \mathcal{Y}$, la log-vraisemblance observée $\log\ g(y, \theta)$ est continuement différentiable sur Θ et :

$$\nabla_\theta g(y, \theta) = \int_{\mathcal{Z}} \nabla_\theta f(y, z; \theta) dz$$

(iii) $\forall k \geq 0$, on a $\gamma_t \in [0,1]$, $\sum_{k=1}^{+\infty} \gamma_t = +\infty$ et $\sum_{k=1}^{+\infty} \gamma_t^2 < +\infty$

(iv) $\nabla_\theta F$ est L-lipshitz

(v) $\exists C > 0 : \mathbb{E}[\|\nabla_\theta \log f(y, Z; \theta)\|^2] \leq C$

(vi) $\exists \lambda_m > 0$ et $\lambda_M > 0 : \forall 1 \leq t \leq K, \forall \lambda \in \mathrm{Sp}(I_t), \lambda_m < \lambda < \lambda_M$

Comme le précise l'article, les trois premières hypothèses sont des hypothèses classiques dans l'estimation des modèles à variables latentes où l'on exige un minimum de régularité sur les fonctions de vraisemblance. Les hypothèses (iv) et (v) sont utilisées pour de nombreuses preuves de convergence d'algorithme de descente de gradient stochastique. L'hypothèse (vi) est la plus restrictive puisqu'elle nécessite en pratique d'être vérifiée pour chaque itération de l'algorithme et pour chaque estimation de l'information de fisher. Les auteurs précisent toutefois que, si l'information de fisher I^* du modèle étudié est définie positive, l'estimation de I_t lorsque k et N tendent vers l'infini converge vers I^* et est donc définie positive pour k et N suffisamment grands. C'est en particulier l'étude qu'ils ont réalisé sur leurs précédents travaux (*Delattre et Kuhn* [7])

En pratique l'étape de pre-heating que l'on utilise dans l'algorithme de descente permet de laisser l'information de fisher se stabiliser au cours des itérations avant de pouvoir l'utiliser efficacement pour l'optimisation. Durant cette phase, des poids sont ajoutés sur la diagonale de la matrice pour éviter un mauvais conditionnement de l'information de fisher estimée aux premières itérations. Les valeurs de γ_t utilisées sur les trois phases de l'algorithme permettent de vérifier l'hypothèse (iii). Sous ces hypothèses et en notant $F := -\log f(y, z; \theta)$, on peut montrer le résultat suivant :

$$\mathbb{E}[\min_{0 \leq l \leq k} \|\nabla_\theta F(\theta_l)\|^2] \leq \frac{2(F(\theta_0) - minF)}{2\mu_m \sum_{l=0}^{k} \gamma_l} + \frac{\mu_M^2 CL \sum_{l=0}^{k} \gamma_l^2}{2\mu_m \sum_{l=0}^{k} \gamma_l}$$

avec une borne qui tend vers 0 lorsque k tend vers $+\infty$ par (iii). La preuve de ce résultat peut être réalisée par inégalité de Taylor-Lagrange à l'ordre 2 sur $F(\theta_t)$. la dérivée seconde est bornée (le gradient de F est L-lipshitz par (iv)). Après passage à l'esperance, les hypothèses (v) et (vi) permettent de trouver la borne ci-dessus.

3 Résultats obtenus par les auteurs

L'article fournit un premier lien (FSGD-mixedeffects) destiné à l'implémentation des *mixed-effect models*. Pour ce modèle, on génère 1000 datasets de 1000 individus sur lesquels sont effectués les analyses comparatives entre les algorithmes Fisher-SGD et SAEM. Nous avons pu retrouver les analyses de convergence des estimateurs qui sont fournies en annexe A. Cependant, on constate que l'estimation n'a pas convergé pour l'ensemble des datasets générés. On retrouve certains

échantillons pour lesquelles l'estimation dégénère après la phase de heating et où l'estimation avant et pendant la phase de heating n'est pas satisfaisante. La table d'analyse donnant les erreurs en RMSE est ici mauvaise, en particulier à côté des résultats de l'application de l'algorithme SAEM sur les mêmes données. Un autre test a été réalisé sur des données réelles accessibles publiquement (data). On retrouve pour ces données les mêmes résultats que ceux fournis par l'article. A noter que l'algorithme SAEM n'a pas été utilisé sur ces données.

Sur un deuxième lien (FSGD-SBM) l'algorithme de descente de gradient conditionné est réalisé pour les modèles de blocs stochastiques avec 2000 essais pour des graphes de 100 ou 200 neouds. De manière similaire, on retrouve en sortie d'exécution les figures de convergence des estimateurs fournis par l'article. Pour la table d'analyse qui fournit les erreurs en RMSE des estimateurs par rapport aux paramètres réels, il est important de noter que dans l'article, l'algorithme est lancé 20 fois pour chaque jeu de données et conserve uniquement l'estimateur qui maximise la log-vraisemblance. Nos résultats sont légèrements moins bons car Fisher-SGD n'a été lancé qu'une fois par dataset. Toutes les figures retrouvées sont fournies en annexe A.

4 Implémentation pour le modèle de mélange gaussien

Nous avons réalisé une implémentation (github) pour les modèles de mélange gaussien afin de comparer les résultats obtenus par descente de gradient par rapport aux algorithmes EM (package Rmixmod). On se donne un n-échantillon $(Y_1, ..., Y_n)$ que l'on suppose conditionné par un phénomène latent non observé $(Z_1, ..., Z_n)$ avec :

$$\begin{cases} Y_i \mid Z_i = k \sim \mathcal{N}(\mu_k, \sigma_k^2) \\ Z_i \sim \mathcal{M}(1, \pi) \end{cases}$$

où π est un vecteur de probabilités sur K classes possibles pour Z_i, avec $\sum_k \pi_k = 1$

On veut estimer le paramètre $\theta = (\pi, \mu, \sigma^2)$ qui maximise la log-vraisemblance :

$$\log\ f(\theta; y_i, z_{ik}) = \sum_{k=1}^{3} z_{ik} \log(\pi_t \phi(yi; \mu_k, \sigma_k^2))$$

où $\phi(y; \mu_k, \sigma_k^2)$ désigne la densité de la loi normale de la k-ième composante centrée en μ_k et de variance σ_k^2.

La loi a posteriori $p_{Z_i=k|Y_i}$ est obtenue par formule de Bayes sur le paramètre θ en cours :

$$z_{ik} = p_{Z_i=k|Y_i}(y, z; \theta) = \frac{p_{Y_i|Z_i=k}(y_i, z_i; \theta) p_{Z_i=k}(z_i; \theta)}{p_{Y_i}(y; \theta)} = \frac{\pi_t \phi(y_i; \mu_k, \sigma_k^2)}{\sum_j \pi_j \phi(y_i; \mu_j, \sigma_j^2)}$$

A partir de l'estimation a posteriori z_{ik}, on cherche ensuite à maximiser l'espérance de la log-vraisemblance par descente de gradient. On relâche dans notre cas le paramètre π_3 conditionné par la contrainte $\sum_j \pi_j = 1$. Dans notre cas, on utilisera directement l'expression analytique du gradient. Les auteurs précisent cependant que cette expression analytique n'est pas nécessaire et peut être calculée durant l'optimisation par différenciation automatique :

$$\nabla_\theta \log\, f(\theta; y_i, z_{ik})$$

$$= \begin{cases} \frac{\partial}{\partial \pi_t} \log\, f(\theta; y_i, z_{ik}) = \frac{z_{ik}}{\pi_t} - \frac{z_{i3}}{(1-\pi_1-\pi_2)}, \quad k \in \{1, 2\} \\ \frac{\partial}{\partial \mu_k} \log\, f(\theta; y_i, z_{ik}) = z_{ik} \frac{y-\mu_k}{\sigma_k^2} \\ \frac{\partial}{\partial \sigma_k^2} \log\, f(\theta; y_i, z_{ik}) = z_{ik} (\frac{(y-\mu_k)^2}{2\sigma_k^4} - \frac{1}{2\sigma_k^2}) \end{cases}$$

Pour notre simulation, nous avons tiré 25 échantillons de 5000 données avec les paramètres $\mu = (-6, -1, 3)$, $\sigma = (1.25, 1., 0.5)$ et $\pi = (1/5, 7/15, 1/3)$. On utilise pour cette expérimentation 5 paramètres initiaux tirés de la façon suivante : le paramètre μ_0 est tiré uniformément sur trois plages équivalentes entre le minimum et le maximum observé de l'échantillon et σ_0^2 est tiré en proportion de la variance globale de l'échantillon. A partir de ces deux paramètres, on réalise ensuite une classification naïve des éléments sur les trois composantes pour déterminer les proportions initiales. La méthode de descente de gradient stochastique par conditionnement et sans conditionnement est évaluée pour les mêmes paramètres initiaux. Les deux algorithmes sont effectués sur 8000 itérations et l'étape de pre-heating est suspendue après 3000 itérations pour l'algorithme conditionné. Les taux d'apprentissage initiaux sont fixés à $\gamma = 0.01$ (constant pour SGD) et $\gamma_0 = 10^{-6}$ (SGD-Fisher). Pour les algorithmes SGD et Fisher-SGD on compare également le nombre de convergences réussies pour ce modèle et trois légères variations :

— *model (base)* :
$\theta^*_{base} = (1/5, 7/15, 1/3, -6, -1, 3, 1.25, 1., 0.5)$
— *model (2)* :
$\theta^*_2 = (1/5, 7/15, 1/3, -4, 0, 2, 1.25, 1., 0.5)$ (faible distinction)
— *model (3)* :
$\theta^*_3 = (1/15, 13/15, 1/15, -6, -1, 3, 1.25, 1., 0.5)$ (proportions déséquilibrées)
— *model (4)* :
$\theta^*_4 = (1/5, 7/15, 1/3, -6, -1, 3, 1., 1., 1.)$ (même variance)

	Rmixmod (EM) RMSE	SGD RMSE	SGD-Fisher RMSE
π_1	0.013	0.006	0.002
π_2	0.016	0.007	0.002
μ_1	0.101	0.096	0.039
μ_2	0.049	0.029	0.018
μ_3	0.028	0.013	0.012
σ_1^2	0.080	0.416	0.072
σ_2^2	0.042	0.144	0.038
σ_3^2	0.021	0.009	0.008

TABLE 1 – Analyse comparative des erreurs d'estimation (base)

	model (base)	model (2)	model (3)	model (4)
SGD	100%	94%	100%	84%
Fisher-SGD	99%	91%	77%	57%

TABLE 2 – Taux de convergences réussies par modèle

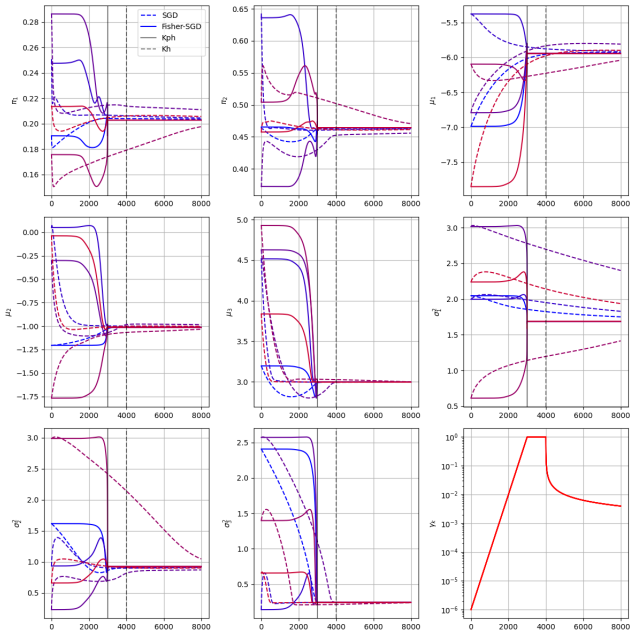
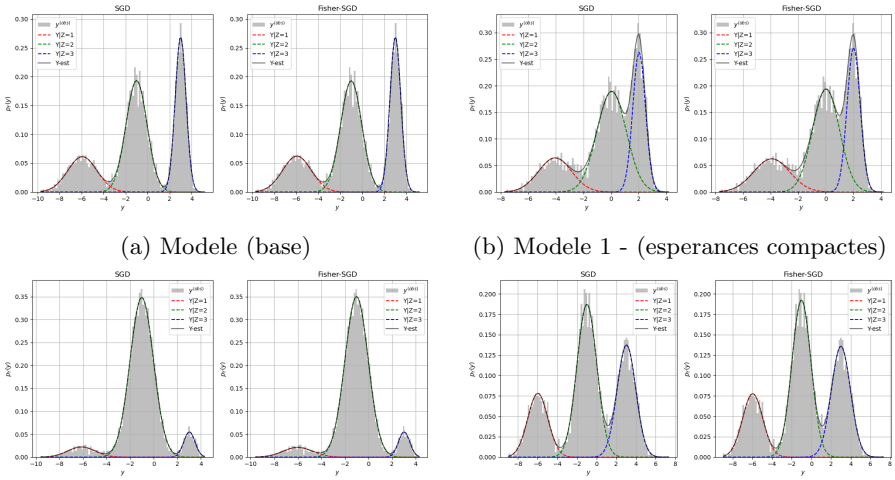


FIGURE 1 – Convergence du paramètre après 8000 itérations sur le premier échantillon du modèle (base) pour cinq paramètres initiaux aléatoires. Convergence du SGD en pointillés et du Fisher-SGD en traits pleins. Evolution du taux d'apprentissage γ_t sur la dernière figure.



(a) Modele (base)

(b) Modele 1 - (esperances compactes)

(c) Modele 2 - (déséquilibre)

(d) Modele 3 - (même variance)

FIGURE 2 – Densités obtenues après 8000 itérations

5 Conclusion et critique générale sur l'article

L'article présente très bien les enjeux de l'étude des problèmes d'estimation des modèles à variables latentes en rappelant notamment tous les travaux qui ont pu être réalisés dans ce domaine au cours du temps. En se concentrant ici aux méthodes de descente de gradient qui ont été étudié plus récemment dans ce domaine (en comparaison des algorithmes de type EM), les auteurs apportent de manière efficace une réponse à leur problématique en rappelant le schéma théorique des modèles à variables latentes, en expliquant par la suite comment l'on peut optimiser la descente de gradient stochastique pour ces modèles avant de présenter leurs résultats expérimentaux.

On peut regretter que l'étude ne se compare ici qu'aux algorithmes SAEM (variante stochastique de l'algorithme EM) et pas à des méthodes de descente de gradient sans conditionnement et heating (puisque ces deux facteurs sont censés améliorer l'efficacité de l'optimisation par descente). On note par notre expérimentation sur les modèles de mélange gaussiens que la convergence de l'algorithme, bien que plus efficace en nombre d'itérations, semble ici parfois plus sensible à l'initialisation du paramètre que l'on cherche à estimer, avec de nombreux cas où nous n'avons pas pu observer la convergence de l'algorithme contrairement à une méthode de descente classique. A noter qu'il est ici nécessaire de callibrer des paramètres d'optimisation supplémentaires pour pouvoir aider la convergence de l'algorithme (durée de pre-heating, taux d'apprentissage initial)

Références

[1] Charlotte Baey, Maud Delattre, Estelle Kuhn, Jean-Benoist Leger, and Sarah Lemler. Efficient preconditioned stochastic gradient descent for estimation in latent variable models. June 2023.

[2] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society Series B : Statistical Methodology*, 39(1) :1–22, September 1977.

[3] Bernard Delyon, Marc Lavielle, and Eric Moulines. Convergence of a stochastic approximation version of the em algorithm. *The Annals of Statistics*, 27(1), March 1999.

[4] Matthew J Beal and Zoubin Ghahramani. *The Variational Bayesian EM Algorithm for Incomplete Data : With Application to Scoring Graphical Model Structures*, pages 453–463. Oxford University PressOxford, July 2003.

[5] Li Cai. Metropolis-hastings robbins-monro algorithm for confirmatory item factor analysis. *Journal of Educational and Behavioral Statistics*, 35(3) :307–335, June 2010.

[6] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider : Near-optimal non-convex optimization via stochastic path integrated differential estimator. July 2018.

[7] Maud Delattre and Estelle Kuhn. Estimating fisher information matrix in latent variable models based on the score function. September 2019.

[8] Ilya Loshchilov and Frank Hutter. Sgdr : Stochastic gradient descent with warm restarts. August 2016.

[9] Leslie N. Smith and Nicholay Topin. Super-convergence : Very fast training of neural networks using large learning rates. August 2017.

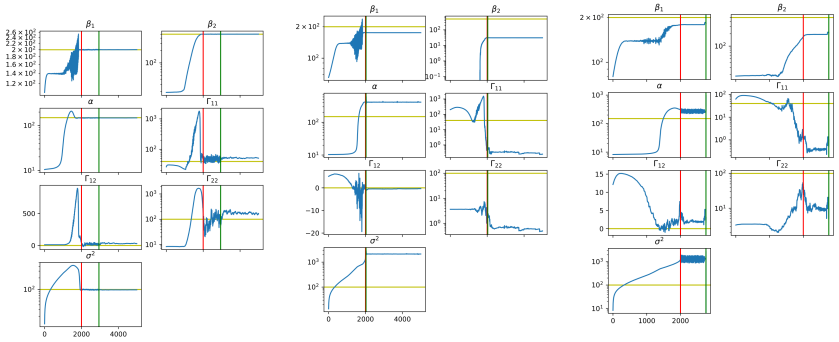
[10] Emmanuelle Comets, Audrey Lavenu, and Marc Lavielle. Parameter estimation in nonlinear mixed effect models using saemix, an r implementation of the saem algorithm. *Journal of Statistical Software*, 80(3), 2017.

[11] Guanhua Fang and Ping Li. On estimation in latent variable models. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3100–3110. PMLR, 18–24 Jul 2021.

[12] Johnny Hong, Sara Stoudt, and Perry de Valpine. Fast maximum likelihood estimation for general hierarchical models. *Journal of Applied Statistics*, pages 1–29, July 2024.

[13] Olivier Cappé. *Inference in Hidden Markov Models*. Springer Series in Statistics Ser. Springer New York, New York, NY, 2007. Description based on publisher supplied metadata and other sources.

A Réexécution du code fourni par les auteurs



(a) Convergence de l'estimateur

(b) Non convergence de l'estimateur avec phase de heating très courte

(c) Non convergence de l'estimateur après la phase de heating

FIGURE 3 – Exemples de convergence de quelques estimateurs du *mixed-effect model* (en bleu). Le paramètre réel est identifié en jaune et les phases de heating et de decreasing step sont affichées par les lignes verticales.

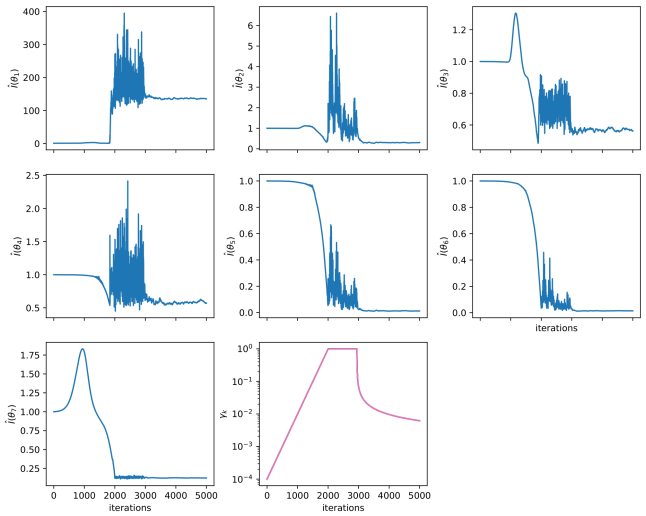


FIGURE 4 – Évolution de la diagonale de l'information de Fisher du *mixed-effect model* (dans un cas où la convergence est correcte).

	Fisher-SGD		SAEM	
Params	**RMSE**	**Coverage**	**RMSE**	**Coverage**
θ	560.10	0. ± 0.	14.9911	1 ± 0
β_1	19.75	0.019 ± 0.008	0.1764	1.00 ± 0.000
β_1	95.83	0.001 ± 0.0019.	0.6286	0.92 ± 0.016
α	10.64	0.115 ± 0.0197	0.3956	1.00 ± 0.000
Σ_{11}	490.10	0.147 ± 0.0219	2.7185	0.84 ± 0.022
Σ_{12}	132.68	0.191 ± 0.0243	4.041	1.00 ± 0.000
Σ_{22}	206.96	0.167 ± 0.0231	14.4005	0.92 ± 0.016
σ^2	58.13	0.102 ± 0.0187	0.9207	0.96 ± 0.012

TABLE 3 – Comparaison entre Fisher-SGD (sans NaN) et SAEM pour le *mixed-effect model*. Le coverage est la proportion de datasets pour lesquels la vrai valeur générant les données est tombée dans un intervalle de confiance à 95% de l'estimateur. Pour l'algorithme SAEM, nous n'avons réalisé le test que sur 25 échantillons (contre 1000 sur l'article).

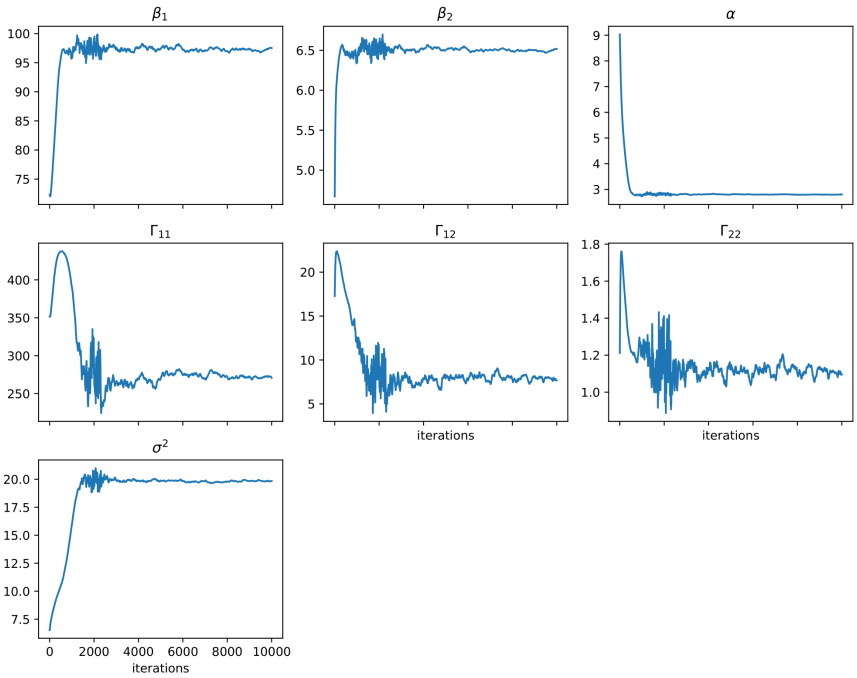


FIGURE 5 – convergence des paramètres du Fisher-SGD sur données réelles pour le *mixed-effect model*. Egalement fourni dans l'article original

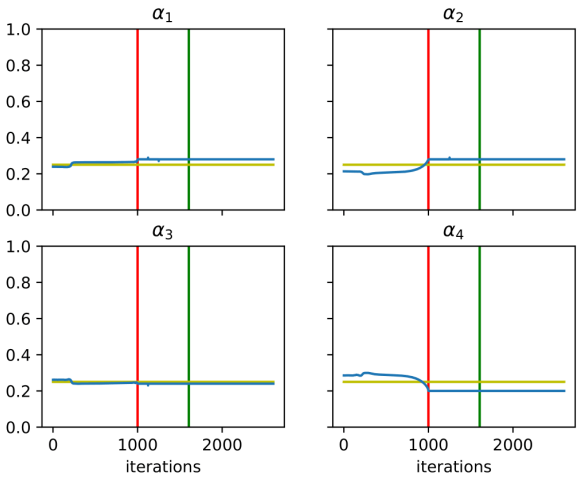


FIGURE 6 – Evolution des estimations du paramètre α pour le *SBM*

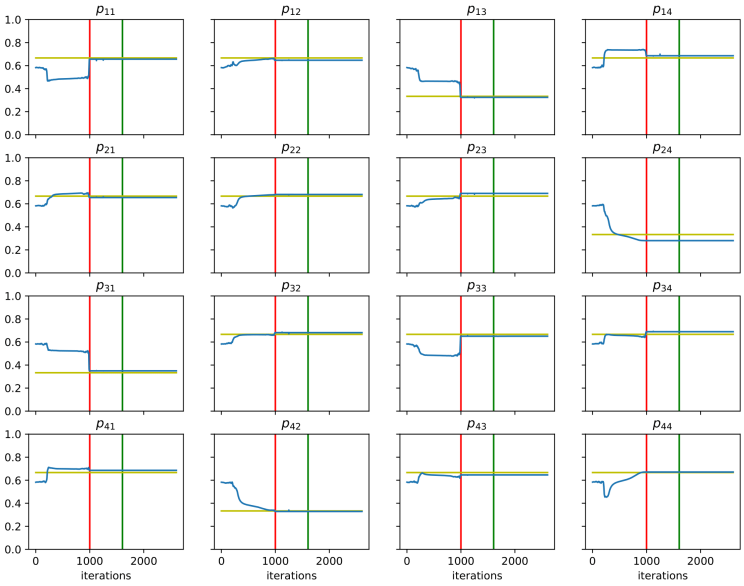


FIGURE 7 – Exemple de l'évolution des estimations des p du *SBM*. Les résultats sont similaires à ceux de l'article.

		N=100		N=200	
	Simulated val	**RMSE**	**Empirical cov**	**RMSE**	**Empirical cov**
θ	.	1.073	0.880 ± 0.064	0.886	0.880 ± 0.064
α_1	0.250	0.051	0.940 ± 0.047	0.043	0.870 ± 0.066
α_2	0.250	0.051	0.900 ± 0.059	0.053	0.910 ± 0.056
α_3	0.250	0.055	0.920 ± 0.053	0.047	0.900 ± 0.059
α_4	0.250	0.057	0.910 ± 0.056	0.041	0.910 ± 0.056
$p_{1,1}$	0.667	0.057	0.920 ± 0.053	0.031	0.910 ± 0.056
$p_{1,2}$	0.667	0.033	0.940 ± 0.047	0.025	0.930 ± 0.050
$p_{1,3}$	0.333	0.050	0.910 ± 0.056	0.081	0.850 ± 0.070
$p_{1,4}$	0.667	0.030	0.960 ± 0.038	0.017	0.910 ± 0.056
$p_{2,1}$	0.667	0.029	0.900 ± 0.059	0.030	0.890 ± 0.061
$p_{2,2}$	0.667	0.035	0.900 ± 0.059	0.026	0.920 ± 0.053
$p_{2,3}$	0.667	0.026	0.900 ± 0.059	0.032	0.900 ± 0.059
$p_{2,4}$	0.333	0.060	0.890 ± 0.061	0.061	0.910 ± 0.056
$p_{3,1}$	0.333	0.052	0.900 ± 0.059	0.064	0.850 ± 0.070
$p_{3,2}$	0.667	0.047	0.890 ± 0.061	0.032	0.900 ± 0.059
$p_{3,3}$	0.667	0.068	0.950 ± 0.043	0.028	0.940 ± 0.047
$p_{3,4}$	0.667	0.029	0.900 ± 0.059	0.019	0.940 ± 0.047
$p_{4,1}$	0.667	0.033	0.930 ± 0.050	0.019	0.920 ± 0.053
$p_{4,2}$	0.333	0.045	0.930 ± 0.050	0.059	0.910 ± 0.056
$p_{4,3}$	0.667	0.030	0.930 ± 0.050	0.019	0.920 ± 0.053
$p_{4,4}$	0.667	0.069	0.900 ± 0.059	0.028	0.890 ± 0.061

TABLE 4 – Table des résultats du Fisher-SGD sur le *SBM* dans les cas avec 100 et 200 nœuds. Dans les deux cas, nous avons lancé 100 expériences (au lieu de 2000 dans l'article). Par ailleurs, en raison de contraintes calculatoires, nous n'avons effectué qu'une seule estimation par modèle (au lieu de 20) ce qui peut amener à des résultats moins bons que ceux présentés dans l'article