Ryan Boustany

Paul Couairon

Algorithmes MCMC & MCMC accéléré pour des modèles de régression bayésienne en grande dimension

Cours enseigné par :
Stephan Clémençon

Année scolaire 2020-2021

Table des matières

Introduction

Ce projet est inspiré de l'article de recherche de Rui Jin et Aixin Tan : *Fast Markov chain Monte Carlo for high dimensional Bayesian regression models with shrinkage priors* [1], Journal of Computational and Graphical Statistics, 1-36, 2020.
Nous nous intéressons au modèle général de la régression linéaire à p variables et n exemples, qui est représenté comme suit

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta}^* + \boldsymbol{\varepsilon} \tag{1}$$

où $\boldsymbol{\beta}^*$ représente le vecteur des coefficients de la régression de taille $\boldsymbol{p \times 1}$, que nous souhaitons approcher, $\boldsymbol{X}$ notre matrice de features, de taille $\boldsymbol{n \times p}$, et enfin $\boldsymbol{\varepsilon}$ un bruit gaussien, de taille $\boldsymbol{n \times 1}$. Dans notre cas, nous nous intéressons à un modèle **en grande dimension**, c'est-à-dire que $\boldsymbol{p}$ est très grand, supérieur à $\boldsymbol{n}$, voire grandement supérieur à $\boldsymbol{n}$. L'approche standard pour résoudre ce genre de problème est d'utiliser la méthode de régression pénalisée LASSO, qui nous permet d'estimer de nombreuses valeurs de $\boldsymbol{\beta}_j^*$ à des valeurs nulles. Il est intéressant de quantifier l'incertitude autour de l'estimation de ces coefficients. Une alternative est de proposer des modèles bayésiens, qui nous permettent de quantifier les incertitudes des paramètres et des variables latentes à travers les distributions *à posteriori*. Des méthodes classiques de Monte-Carlo comme celle du Gibbs Sampling nous permettent, grâce à ces distributions, d'approcher le vecteur d'intérêt $\boldsymbol{\beta}^*$. Cependant, en grande dimension, la convergence de l'algorithme de Gibbs est peu satisfaisante. L'idée de R. Jin et A. Tan est de proposer un algorithme de Gibbs alternatif, requérant beaucoup moins de stockage à chaque itération. Dans ce projet, nous nous intéresserons à la comparaison de cet algorithme avec celui de Gibbs classique sur un modèle de **régression groupée**, c'est-à-dire que la matrice de features $\boldsymbol{X}$ présente des groupes de colonnes avec de fortes corrélations [2].
L'étude de ce type de problème n'est pas anodine : par exemple, en sciences sociales, si nous nous intéressons à plusieurs catégories de variables (liées à l'éducation etc.), un phénomène de corrélation intra-classes apparaîtra. Pour un problème de machine learning, le recodage de certaines variables qualitatives à l'aide d'un One-Hot Encoding fera apparaître de la corrélation dans chaque colonne liée à la variable catégorielle concernée. Dans le cadre de la régression groupée, l'estimateur LASSO est défini par :

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{group}} = \boldsymbol{\arg\min_{\beta}} \left\| \boldsymbol{Y} - \sum_{\boldsymbol{k=1}}^{\boldsymbol{K}} \boldsymbol{X}_{\boldsymbol{G_k}} \boldsymbol{\beta}_{\boldsymbol{G_k}} \right\|^{\boldsymbol{2}} + \boldsymbol{\lambda} \sum_{\boldsymbol{k=1}}^{\boldsymbol{K}} \|\boldsymbol{\beta}_{\boldsymbol{G_k}}\|_{\boldsymbol{2}} \tag{2}$$

1. Article disponible sur `https://doi.org/10.1080/10618600.2020.1864383`

2. Les auteurs de l'article appliquent leur modèle pour deux autres cas de grande dimension, que l'on ne traitera pas ici.

où $\boldsymbol{K}$ correspond au nombre de classes (ou groupes) de variables de notre jeu de données, et $\boldsymbol{\beta_{G_k}}$ et $\boldsymbol{X_{G_k}}$ représentent respectivement les sous-matrices de coefficients et de features pour les classes concernées.

1 Framework et objectifs

1.1 Formalisation bayésienne

En régression, l'approche bayésienne est généralement la même : elle consiste à proposer une distribution pour la vraisemblance du modèle, ainsi que des spécifications pour les distributions *à priori* afin d'en déduire les distributions *à posteriori* pour approcher le vecteur cible $\boldsymbol{\beta^*}$. Pour les modèles en grande dimension, la spécificité que l'on souhaite préciser et qui est liée au *shrinkage* (rétrecissement, contraction) fait l'objet d'un paramètre pour lequel on définit une loi *à priori*. Le modèle décrit dans (3) correspond au modèle général utilisé dans le cadre d'estimation bayésienne pour la régression.

$$\begin{aligned} \boldsymbol{Y}|\beta, \boldsymbol{\sigma^2} &\sim \boldsymbol{\mathcal{N}_n}(\boldsymbol{X}\beta, \boldsymbol{\sigma^2 I_n}) \\ \beta|\boldsymbol{\eta}, \boldsymbol{\sigma^2} &\sim \boldsymbol{\mathcal{N}_p}(\boldsymbol{0}, \boldsymbol{\sigma^2 \Sigma_\eta}) \\ \boldsymbol{\eta} &\sim \boldsymbol{p(\eta)} \\ \boldsymbol{\sigma^2} &\sim \text{Inverse-Gamma}(\boldsymbol{\alpha}, \boldsymbol{\xi}) \end{aligned} \tag{3}$$

où $\boldsymbol{Y}$ et $\boldsymbol{X}$ sont définis dans (6), $\boldsymbol{\sigma^2}$ correspond à la variance résiduelle eu modèle, et $\boldsymbol{\eta}$ correspond au paramètre de *shrinkage* dans notre cas. Les paramètres $\boldsymbol{\alpha}, \boldsymbol{\xi}$ concernent quant à eux la distribution à priori que l'on pose pour le paramètre $\boldsymbol{\sigma}$. Dans le cas particulier de la **régression groupée**, R. Jin et A. Tan proposent un cadre bayésien s'inscrivant dans le modèle proposé dans (3) :

$$\begin{aligned} \boldsymbol{Y}|\beta, \boldsymbol{\sigma^2} &\sim \boldsymbol{\mathcal{N}_n}(\boldsymbol{X}\beta, \boldsymbol{\sigma^2 I_n}) \\ \beta_{G_k}|\boldsymbol{\tau_k^2}, \boldsymbol{\sigma^2} &\overset{ind}{\sim} \boldsymbol{\mathcal{N}_p}(\boldsymbol{0}_{m_k}, \boldsymbol{\sigma^2 \tau_k^2 I_{m_k}}) \\ \boldsymbol{\tau_k^2} &\overset{ind}{\sim} \text{Gamma}\left(\frac{\boldsymbol{m_k + 1}}{\boldsymbol{2}}, \frac{\boldsymbol{\lambda^2}}{\boldsymbol{2}}\right) \\ \boldsymbol{\sigma^2} &\sim \text{Inverse-Gamma}(\boldsymbol{\alpha}, \boldsymbol{\xi}) \end{aligned} \tag{4}$$

où l'indice $\boldsymbol{k}$ correspond au groupe concerné, $\boldsymbol{m_k}$ est la taille du $\boldsymbol{k^e}$ groupe, et $\boldsymbol{\tau}$ correspond au paramètre de *shrinkage* (l'équivalent de $\boldsymbol{\eta}$ dans le modèle 3). Notons que le rôle de la matrice $\boldsymbol{\sigma_n u}$ est joué par $\boldsymbol{\tau_k^2 I_{m_k}}$. Le paramètre $\boldsymbol{\tau_k}$ est effectivement propre à chaque groupe ;

il peut représenter la variance intra-groupe par exemple. R. Jin et A. Tan montrent que sous le modèle (4), on peut dériver les lois à posteriori suivantes :

$$\begin{aligned}
&\frac{\mathbf{1}}{\boldsymbol{\tau}_k^2}\Big|\boldsymbol{\beta}, \boldsymbol{\sigma}^2, \boldsymbol{Y} \overset{ind}{\sim} \text{Inverse-Gaussian}\left(\sqrt{\frac{\boldsymbol{\lambda}^2\boldsymbol{\sigma}^2}{\|\boldsymbol{\beta}_{G_k}\|^2}}, \boldsymbol{\lambda}^2\right),\ \boldsymbol{k = 1...K} \\
&\boldsymbol{\sigma}^2|\boldsymbol{\beta}, \boldsymbol{\tau}^2, \boldsymbol{Y} \sim \text{Inverse-Gamma}\left(\frac{\boldsymbol{n+p+2\alpha}}{\mathbf{2}}, \frac{\|\boldsymbol{Y - X\beta}\|_{\mathbf{2}}^{\mathbf{2}} + \boldsymbol{\beta}^T\boldsymbol{D}_{\boldsymbol{\tau}}^{-1}\boldsymbol{\beta} + \mathbf{2\xi}}{\mathbf{2}}\right) \qquad (5) \\
&\boldsymbol{\beta}|\boldsymbol{\sigma}^2, \boldsymbol{\tau}^2, \boldsymbol{Y} \sim \boldsymbol{\mathcal{N}_p}\left(\left(\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{D}_{\boldsymbol{\tau}}^{-1}\right)^{-1}\boldsymbol{X}^T\boldsymbol{Y}, \boldsymbol{\sigma}^2\left(\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{D}_{\boldsymbol{\tau}}^{-1}\right)^{-1}\right)
\end{aligned}$$

où $\boldsymbol{D_\tau} = \text{Diag}(\underbrace{\boldsymbol{\tau_1^2, ..., \tau_2^2}}_{\boldsymbol{m_1}}, \underbrace{\boldsymbol{\tau_2^2, ..., \tau_2^2}}_{\boldsymbol{m_1}}, \boldsymbol{...}, \underbrace{\boldsymbol{\tau_K^2, ..., \tau_K^2}}_{\boldsymbol{m_K}}) = \boldsymbol{\tau_k^2 I_{m_k}}$.

Maintenant que l'on possède un modèle bayésien précis, avec des distributions à posteriori connues, l'idée est d'appliquer les bonnes propriétés de l'inférence bayésienne dans le but d'approcher le vecteur cible à l'aide de ces lois.

1.2 Objectifs

De la même manière que l'approche fréquentiste, on souhaite approcher un vecteur cible $\boldsymbol{\beta^*}$. Dans notre approche, nous ne tentons pas de construire des estimateurs basés sur les observations empiriques de $\boldsymbol{X}$ et $\boldsymbol{Y}$ (des réalisations $((X_i, Y_i)_{i=1...n})$), mais de proposer des différentes lois à priori sur les paramètres, un modèle statistique (cette fois-ci grâce aux observations empiriques), d'en déduire les lois à posteriori et enfin de simuler ces dernières de nombreuses fois dans le but d'approcher la vraie distribution de $\boldsymbol{\beta}$. L'algorithme classique utilisé pour ce genre de simulation est le Gibbs sampling, qui consiste en des tirages successifs selon les distributions à posteriori dans l'espoir que les paramètres convergent. On rappelle l'algorithme générique de Gibbs, pour deux paramètres $\boldsymbol{\beta}$ et $\boldsymbol{\sigma}$ que l'on souhaite estimer, suivant respectivement les distributions à posteriori $\boldsymbol{\pi(\beta|X, Y, \sigma)}$ et $\boldsymbol{\pi(\sigma|X, Y, \beta)}$:

Algorithm 1: Gibbs sampling

input : $\boldsymbol{\beta^{[0]} \sim \mathcal{N}_p(0, 1)}$,$\boldsymbol{n}$
for $\boldsymbol{i = 1...n}$ **do**
  draw $\boldsymbol{\sigma^{[k]} \sim \pi(\sigma|X, Y, \beta^{[k-1]})}$;
  draw $\boldsymbol{\beta^{[k]} \sim \pi(\beta|X, Y, \sigma^{[k]})}$;
end
return $\boldsymbol{(\beta^{[n]}, \sigma^{2[n]})}$

Sous des conditions concernant les noyaux de transitions des deux distributions, il peut être montré que l'algorithme 1 converge (voir [G.O.Roberts, A.F.M.Smith]). Cependant, la convergence de l'algorithme ne peut être connue en amont ; le nombre d'itérations nécéssaires avant d'atteindre un voisinage autour de la solution peut être grand. L'objectif de R. Jin et A. Tan est de proposer une méthode **accélérée** de l'algorithme 1 dans le but d'approcher le vecteur cible dans un contexte de grande dimension.

2 Algorithmes

Le premier algorithme correspond à une application du Gibbs sampling adapté au cadre (5).

Algorithm 2: 3BG

input : $\boldsymbol{\beta^{[0]} \sim \mathcal{N}_p(0,1)}$, $\boldsymbol{\sigma^{[0]} \sim \mathcal{N}(0,1)}$, $\boldsymbol{n}$

for $\boldsymbol{i = 1...n}$ **do**

draw $\boldsymbol{\frac{1}{\tau_k^{2[i]}} \Big| \beta, \sigma^2, Y \overset{ind}{\sim}}$ Inverse-Gaussian $\boldsymbol{\left(\sqrt{\frac{\lambda^2 \sigma^{2[i-1]}}{\|\beta_{G_k}{}^{[i-1]}\|^2}}, \lambda^2 \right)}$, $\boldsymbol{k = 1...K}$;

draw $\boldsymbol{\sigma^{2[i]} | \beta, \tau^2, Y \sim}$ Inverse-Gamma $\boldsymbol{\left(\frac{n+p+2\alpha}{2}, \frac{\|Y - X\beta^{[i]}\|_2^2 + \beta^{T[i]} D_\tau^{-1[i]} \beta^{[i]} + 2\xi}{2} \right)}$;

draw

$$\boldsymbol{\beta^{[i]} | \sigma^2, \tau^2, Y \sim \mathcal{N}_p \left(\left(X^T X + D_\tau^{-1[i]} \right)^{-1} X^T Y, \sigma^{2[i]} \left(X^T X + D_\tau^{-1[i]} \right)^{-1} \right)}$$

end

return $\boldsymbol{(\beta^{[n]}, \sigma^{2[n]})}$

Le second algorithme est une version modifiée de l'algorithme (2), en modifiant la loi à posteriori pour $\boldsymbol{\sigma^2}$. Les preuves concernant les calculs de cette distribution à posteriori peuvent être trouvées dans le supplementary material fourni par les auteurs de l'article.

Algorithm 3: 2BG

input : $\beta^{[0]} \sim \mathcal{N}_p(0,1)$, $\sigma^{[0]} \sim \mathcal{N}(0,1)$, n

for $i = 1...n$ **do**

draw $\frac{1}{\tau_k^{2[i]}} \Big| \beta, \sigma^2, Y \overset{ind}{\sim}$ Inverse-Gaussian $\left(\sqrt{\frac{\lambda^2 \sigma^{2[i-1]}}{\|\beta_{G_K}{}^{[i-1]}\|^2}}, \lambda^2 \right)$, $k = 1...K$;

draw $\sigma^{2[i]} | \beta, \tau^2, Y \sim$ Inverse-Gamma $\left(\frac{n}{2} + \alpha, \frac{Y^T (I_n - X(X^T X + D_\tau^{-1})^{-1} X^T) Y}{2} + \xi \right)$

$\beta^{[i]} | \sigma^2, \tau^2, Y \sim \mathcal{N}_p \left(\left(X^T X + D_\tau^{-1[i]} \right)^{-1} X^T Y, \sigma^{2[i]} \left(X^T X + D_\tau^{-1[i]} \right)^{-1} \right)$

end

return $(\beta^{[n]}, \sigma^{2[n]})$

Il est important de comprendre que pour 2BG, les tirages de σ^2 et β ne sont pas les mêmes que pour 3BG. En effet, R. Jin et A. Tan proposent une alternative pour la distribution à posteriori de σ^2, qui ne repose plus sur le vecteur β entièrement ; en ce sens, la densité de transition de la chaîne de Markov liée à (σ^2, β) ne nécessite plus un calcul prenant en compte β pour la distribution conditionnelle, ce qui allège le temps de calcul. De plus, le paramètre de l'inverse-Gaussienne est pris en se concentrant sur le K^e groupe de variables.

3 Résultats

Pour des raisons de puissance de machine, nous avons fixé le nombre d'itérations des samplers à 500. Les auteurs ont pu faire tourner les algorithmes 2BG et 3BG pour $\mathbf{100,000}$ itérations, car disposant d'une machine de 64 GB RAM, 16 coeurs et processeur de 2.6 GHz. Notre machine contient 8 GB RAM, 4 coeurs et processeur 1,4 GHz, donc une capacité d'exécution drastiquement plus faible.

3.1 Données simulées

Nous avons décidé de tester nos algorithmes sur des données simulées selon un schéma proposé par les auteurs, afin de prendre en compte l'idée de "groupes" dans la matrice de features. On rappelle le modèle général donné dans (6) :

$$Y = X\beta^* + \varepsilon \tag{6}$$

Notre objectif est de travailler en grande dimension et de s'intéresser à l'évolution des performances de nos algorithmes lorsque $\frac{p}{n}$ devient de plus en plus grand. D'autre part, on souhaite dans notre cas prendre en compte la présence de corrélation intra-groupes. Notons n la

taille de l'échantillon d'entrée, et $\boldsymbol{K}$ un élément de l'ensemble $\boldsymbol{\mathcal{K} = \{5, 6, 7, 8, 9, 10, 20, 30, 40, 50\}}$ (on le fera varier dans nos expériences). Pour la simulation de la matrice de features, R. Jin et A. Tan proposent la démarche suivante :

- Simuler une matrice intermédiaire $\boldsymbol{A}$ de taille $\boldsymbol{n \times K}$ formée de $\boldsymbol{K}$ colonnes de lois $\boldsymbol{\mathcal{N}_K(0_K, I_K)}$.
- Faire apparaître le phénomène de groupe en stackant à la suite de chaque colonne, les colonnes contenant ses valeurs au carré, au cube, puissance **4** et puissance **5**.
- On pose alors $\boldsymbol{X}$ comme étant cette matrice-là, donc de la forme :

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{X_{11}} & \boldsymbol{X_{11}^2} & \cdots & \cdots & \boldsymbol{X_{11}^5} & \cdots & \cdots \boldsymbol{X_{1K}} & \boldsymbol{X_{1K}^2} & \cdots & \boldsymbol{X_{1K}^5} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{X_{n1}} & \boldsymbol{X_{n1}^2} & \cdots & \cdots & \boldsymbol{X_{n1}^5} & \cdots & \cdots \boldsymbol{X_{nK}} & \boldsymbol{X_{nK}^2} & \cdots & \boldsymbol{X_{nK}^5} \end{pmatrix}$$

Une fois la matrice $\boldsymbol{X}$ simulée, nous générons $\boldsymbol{\beta^*}$ selon un vecteur aléatoire suivant une loi de Student pour les $\boldsymbol{K}$ premières coordonnées et des $\boldsymbol{0}$ sur les $\boldsymbol{4K}$ coordonnées suivant. Puis on génère un réel positif $\boldsymbol{\sigma^*}$ (généralement pas plus grand que 2). Finalement, on peut générer $\boldsymbol{Y}$ tel que $\boldsymbol{Y = X\beta^* + \varepsilon}$ avec $\boldsymbol{\varepsilon \sim \mathcal{N}_n(0_n, \sigma^* I_n)}$.

À noter que l' algorithme $\boldsymbol{2BG}$ prend en entrée les éléments suivant :

- $\boldsymbol{X}$: de taille $\boldsymbol{n \times p}$, $\boldsymbol{p = 5K}$.
- $\boldsymbol{Y}$: de taille $\boldsymbol{n \times 1}$.
- $\boldsymbol{group\_size}$: nombre d'individus par groupe (dans notre cas $\boldsymbol{5}$).
- $\boldsymbol{\beta}$: initiatlisation du vecteur $\boldsymbol{\beta}$.
- $\boldsymbol{\sigma^2}$: initiatlisation du vecteur $\boldsymbol{\sigma^2}$.
- $\boldsymbol{\lambda}$: hyperparamètre.
- $\boldsymbol{\alpha}$: hyperparamètre.
- $\boldsymbol{\xi}$: hyperparamètre.
- $\boldsymbol{n_{iter}}$: nombre d'itérations.

3.2 Convergence en cas sur-déterminé

Cas de $\boldsymbol{p < n}$: convergence de $\boldsymbol{\sigma^2}$ et $\boldsymbol{\beta}$ Dans notre exemple présent, nous avons

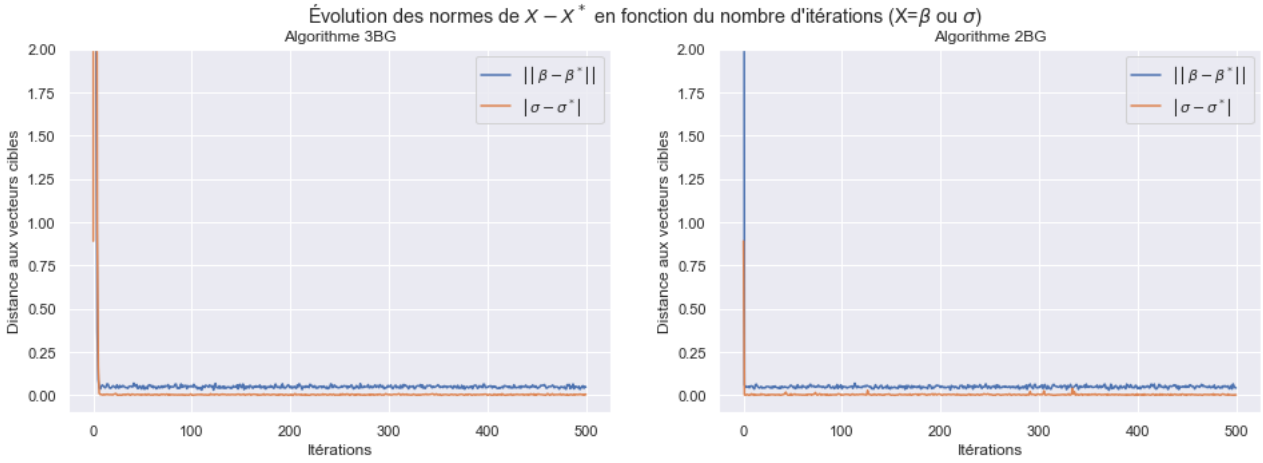

FIGURE 1 – Évolution de $|\boldsymbol{\sigma} - \boldsymbol{\sigma}^*|$ et $\|\boldsymbol{\beta} - \boldsymbol{\beta}^*\|$ en fonction du nombre d'itérations

exécuté les algorithmes $\boldsymbol{2BG}$ et $\boldsymbol{3BG}$ avec $\boldsymbol{n = 1000}$ et $\boldsymbol{p = 50}$. On note que les deux estimateurs ($\boldsymbol{\sigma}$ et $\boldsymbol{\beta}$) issu des deux algorithmes convergent bien et en peu d'itérations vers $\boldsymbol{\beta}^*$ et $\boldsymbol{\sigma}^*$. Cependant, rien ne nous permet de savoir quel algorithme semble le plus adapté à ce problème.

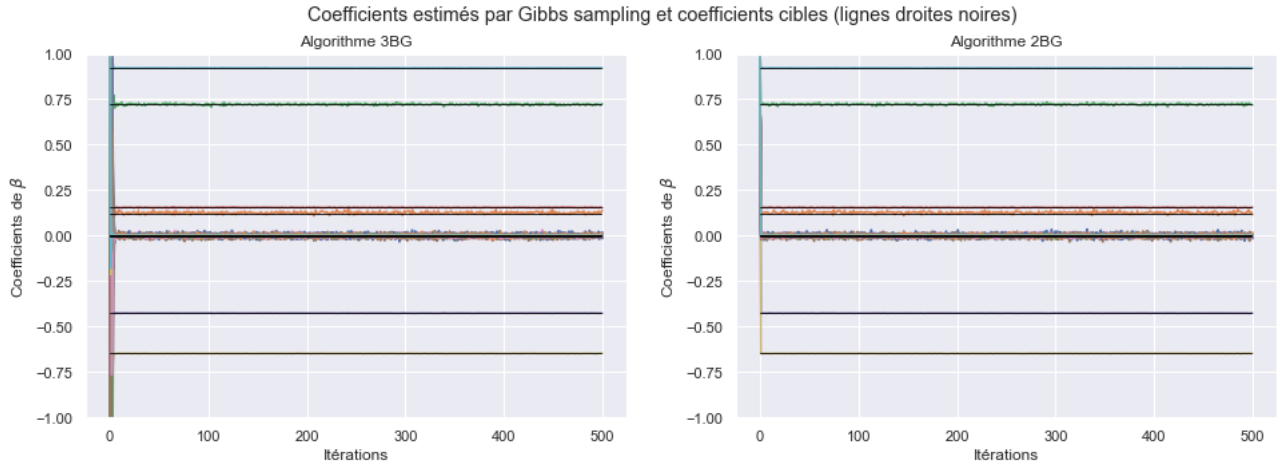

FIGURE 2 – Coefficients estimés par Gibbs sampling et coefficients cibles (lignes droites noires)

3.3 Convergence en grande dimension

L'objectif principal de ce projet est d'étudier les simulations et le comportement des différents algorithmes. On ne regardera pas vraiment quels modèles ou quelles valeurs d'hyperparamètres correspondent le mieux à chaque scénario. Ainsi, nous présentons les résultats empiriques des modèles les plus raisonnables. Nous avons aussi décidé de fixer nous même les hyperparamètres en prenant $\boldsymbol{\alpha = \xi = 0}$ et $\boldsymbol{\lambda = 1}$.

3.3.1 Évolution de $\boldsymbol{\sigma}$

Cas de $\boldsymbol{p > n}$: convergence de $\boldsymbol{\sigma^2}$

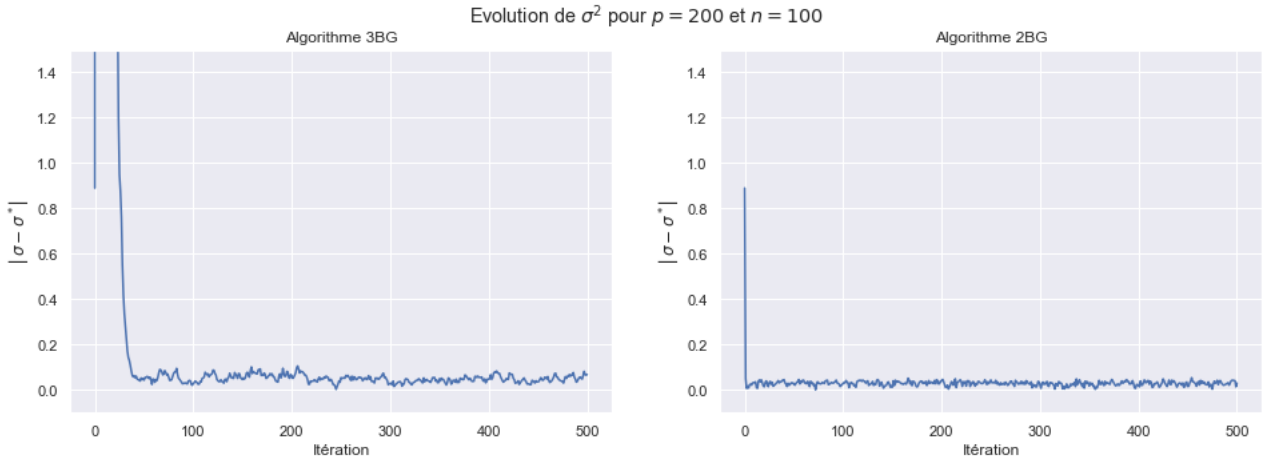

FIGURE 3 – Evolution de $\boldsymbol{\sigma^2}$ pour $\boldsymbol{p = 200}$ et $\boldsymbol{n = 100}$

Dans ce cas présent, nous nous sommes concentrés sur $\boldsymbol{\sigma}$. Les deux algorithmes nous donnent une approximation de la variance. Au bout de $\mathbf{100}^e$ itération, l'algorithme $\boldsymbol{2BG}$ proposé par les auteurs atteint une approximation plus satisfaisante. Cependant, notons que en changeant un hyperparamètre, un Sampler peut performer mieux que l'autre. Les résultats sont globalement comparables. En regardant l'évolution de la chaîne, l'algorithme Gibbs classique semble prendre plus de temps avant de se rapprocher de la valeur cible. 2BG semble plus direct.

3.3.2 Évaluation de la vitesse de l'algorithme

On s'intéresse ici à une autre façon d'interpréter les résultats. On peut aussi se demander si la convergence est sensible au paramètre que nous posons sur l'a priori du paramètre de shrinkage $\boldsymbol{\tau}$. L'idée de R. Tin et A.Tan est alors d'évaluer, en termes de vitesse de convergence, la chaîne de Markov construite par le Gibbs sampler. Nous définissions l'effective sampling size, donnée par :

$$N_{eff} = \frac{N}{1 + 2\sum_{k=1}^{\infty} \rho_k}$$

où ρ_k est l'autocorrélation de lag k de la chaîne de σ^2 (sous hypothèse de stationnarité).

L'effective sampling size est une estimation du nombre de tirages requis lors de l'algorithme afin d'atteindre le même niveau de précision que si le tirage était aléatoire. Il s'agit d'une mesure importante de l'efficacité d'une méthode de Monte-Carlo comme celles présentées par les algorithmes que les auteurs ont proposé. Cette métrique évalue la qualité du tirage effectué par les algorithmes 2BG et 3BG à chaque itération. Numériquement, plus N_{eff} est élevé, plus ceci indique une meilleure mixité de la chaîne puisqu'il s'agit d'une fonction décroissante des autocorrélations.

Nous allons également prendre le temps de calcul en considération et nous allons examiner l'effective sample size moyen par seconde $\frac{N_{eff}}{T}$. Autrement dit, le temps de calcul de l'effective sample size est également pris en compte afin de normaliser les valeurs de N_{eff} d'un algorithme à l'autre, mais également de donner un sens à la caractérisation dite "rapide" de l'algorithme 2BG proposé par les auteurs.

On va chercher à évaluer, pour chaque valeur de p, l'effective sample size de la chaîne de σ^2 induite par les algorithmes $2BG$ et $3BG$. Les autocorrélations seront également évaluées afin de calculer la mixité des chaînes. R. Jin et A. Tan indiquent qu'une faible auto-corrélation implique une bonne mixitié de la chaîne induite et donc un meilleur mélange réalisé par l'algorithme.

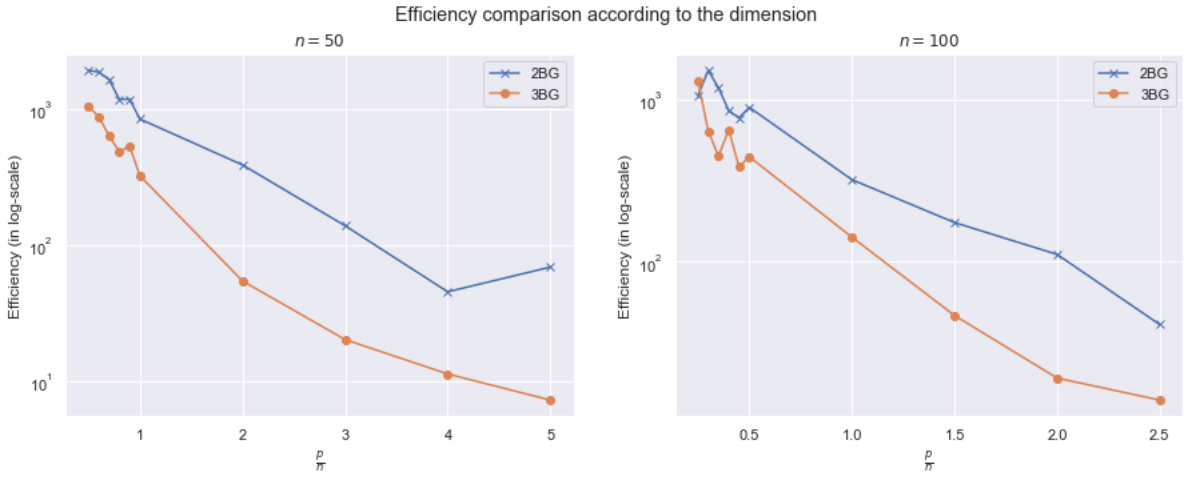

FIGURE 4 – Comparaison de l'efficacité en fonction de la dimension

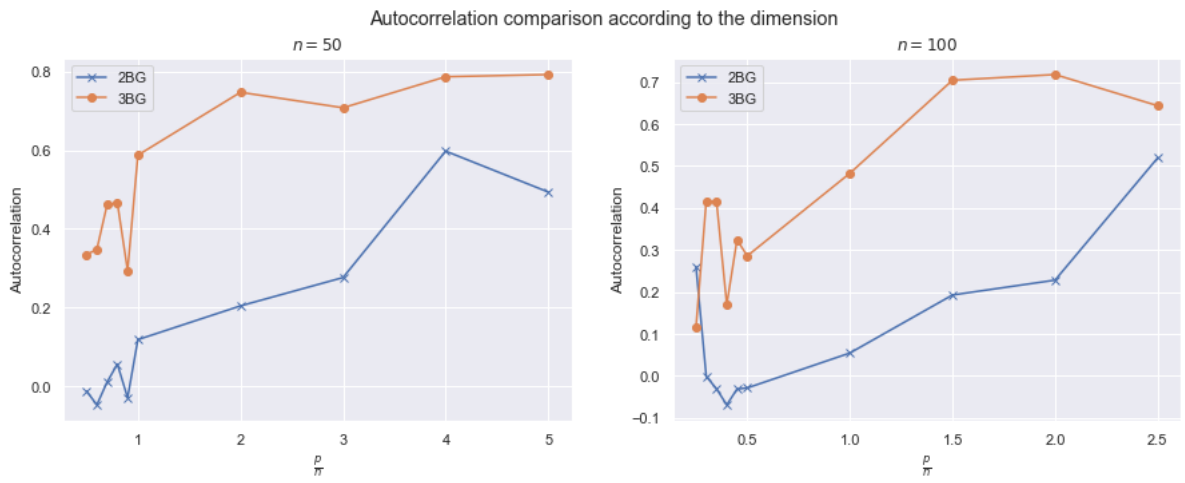

FIGURE 5 – Comparaison d'autocorrélation en fonction de la dimension

La figure 5 montre l'autocorrélation moyenne du décalage de la chaîne $\boldsymbol{\sigma^2}$. Nous pouvons voir que le $\boldsymbol{2BG}$ proposé présente des autocorrélations nettement plus faibles que le $\boldsymbol{3BG}$ original dans toutes les configurations de $\boldsymbol{(n, p)}$, ce qui indique un meilleur mélange du $\boldsymbol{2BG}$. En outre, la figure 4 prend en considération le temps de calcul en examinant la taille moyenne effective de l'échantillon par seconde, sur l'échelle logarithmique de chaque échantillonneur. Il est clair que la $\boldsymbol{2BG}$ est plus efficace. Par exemple, à $\boldsymbol{n = p = 50}$, la $\boldsymbol{2BG}$ produit environ deux fois plus d'échantillons effectifs par seconde que la $\boldsymbol{3BG}$.

4 Conclusion

Dans le modèle bayésien de type Group Lasso, la $\boldsymbol{2BG}$ nous semble toujours plus efficace que la $\boldsymbol{3BG}$. L'avantage majeur de la $\boldsymbol{2BG}$ est lorsque le nombre de prédicteurs $\boldsymbol{p}$ est supérieur à la taille de l'échantillon de données $\boldsymbol{n}$. Dans ces problèmes de régression à haute dimension, la $\boldsymbol{2BG}$ produit environ $\boldsymbol{10}$ à $\boldsymbol{100}$ fois plus d'échantillons efficaces par seconde que son concurrent. Le taux de mélange rapide de la $\boldsymbol{2BG}$ et le faible coût de calcul par itération de la $\boldsymbol{2BG}$, qui n'est pas supérieur à celui de la $\boldsymbol{3BG}$, contribuent à cet avantage.

Pour aller plus loin, une comparaison avec la méthode fréquentiste du GroupLasso classique, dont l'estimateur est donné par la formule $\boldsymbol{2}$, est disponible dans notre Notebook Python joint à ce rapport. Nous verrons notamment que l'efficacité de notre méthode $\boldsymbol{2BG}$ (et même $\boldsymbol{3BG}$) dans le cas en grande dimension est largement comparable à l'estimateur fréquentiste, pour le cas de l'estimation de $\boldsymbol{\beta}$ (pas seulement $\boldsymbol{\sigma}$, paramètre sur lequel nous nous sommes intéressés pour le cas en grande dimension dans ce papier).