

# Inférence bayésienne et méthodes MCMC

Travaux dirigés

*Pierre Gloaguen*

## 1 Inférence bayésienne pour le modèle linéaire

Soit  $Y$  un vecteur d'observations de  $\mathbb{R}^n$ ,  $\beta$  un vecteur de paramètres inconnus  $\mathbb{R}^{p+1}$  (tel que  $n > p + 1$ ) et  $X$  une matrice  $n \times (p + 1)$  telle que la matrice  $X^T X$  soit inversible. On considère le modèle linéaire Gaussien:

$$Y = X\beta + E$$

où  $E$  est un vecteur Gaussien de loi  $\mathcal{N}(0, \sigma^2 I_n)$ .

### 1.1 Cas où $\sigma^2$ est connu

1. Dans le cas où  $\sigma^2$  est connu, écrire la vraisemblance associée au modèle précédent. Montrer que cette vraisemblance est proportionnelle, en tant que densité de probabilité pour le vecteur  $\beta$ , à la densité d'un loi  $\mathcal{N}((X^T X)^{-1} X^T Y, \sigma^2 (X^T X)^{-1})$ . En déduire la densité a posteriori sur  $\beta$  pour une inférence bayésienne effectuée avec un prior impropre.

### 1.2 Cas où $\sigma^2$ est inconnu

Dans ce cas, on pose comme loi a priori que le couple  $(\beta, \sigma^2)$  suit une loi normale inverse Gamma de paramètres  $\mu \in \mathbb{R}^{p+1}$ ,  $V$  (une matrice de variance-covariance de taille  $(p + 1) \times (p + 1)$ ),  $a$  et  $b$  (deux réels positifs).

Formellement:

$$\pi(\beta, \sigma^2 | \mu, \mathbf{V}, a, b) \propto \left(\frac{1}{\sigma^2}\right)^{\frac{p+1}{2}} \left(\frac{1}{\sigma^2}\right)^{a+1} \exp\left(-\frac{b}{\sigma^2}\right) \exp\left(-\frac{(\beta - \mu)^T \mathbf{V}^{-1} (\beta - \mu)}{2\sigma^2}\right).$$

**Remarque** Cette modélisation est en fait assez naturelle, elle correspond au cas où  $\sigma^2$  suit une loi inverse  $\text{Gamma}(a, b)$  (usuelle pour les variances) et  $\beta | \sigma^2 \sim \mathcal{N}(\mu, \sigma^2 V)$ .

1. Montrer que la loi de  $(\beta, \sigma^2) | Y, X$  suit également une loi Normale inverse Gamma dont vous préciserez les paramètres.
2. Interprétez les paramètres en terme “d'apprentissage bayésien”, c'est à dire en distinguant le poids du prior et des données.

## 2 Modèle probit avec covariables

On reprend l'exemple vu en cours et dans l'exercice 5 du TD3 sur l'estimation de covariables corrélées à la présence d'oiseaux.

## 2.1 Notations et modèle

On note  $y_1, \dots, y_n$  les observations de présence (1 si on observe un oiseau, 0 sinon) sur les sites 1 à  $n$ .

On note  $x_{ij}$  la valeur de la  $j$ -ème ( $1 \leq j \leq 3$ ) covariable sur le  $i$ -ème site.

On suppose que les  $y_1, \dots, y_n$  sont les réalisations de variables aléatoires  $Y_1, \dots, Y_n$  telles que

$Y_i \sim \text{Bern}(p_i)$  où

$$p_i = \phi(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3}) = \phi(\mathbf{x}_i^T \theta)$$

où  $\theta = (\beta_0, \dots, \beta_3)^T$  et  $\phi$  la fonction de répartition d'une  $\mathcal{N}(0, 1)$ . L'objectif est d'estimer le vecteur  $\theta$  dans un cadre bayésien.

### 2.1.1 Vraisemblance et posterior

1. Rappelez l'expression de la vraisemblance d'un paramètre  $\theta$  pour vecteur d'observations  $\mathbf{y}$  ainsi que l'expression du posterior associée à un prior  $\mathcal{N}(0, 4I_4)$ .

## 2.2 Algorithme de Metropolis Hastings

2. On se propose d'approcher la loi *a posteriori* en utilisant un algorithme MCMC. Plus précisément, on se propose de générer une chaîne de Markov  $(\theta_n)_{n \geq 0}$  dont l'unique loi stationnaire est le posterior défini plus haut. Pour cela, on utilisera un algorithme de Metropolis Hastings dont le noyau de transition est une marche aléatoire de loi normale  $\mathcal{N}(0, \sigma^2 I_4)$  où  $I_4$  est la matrice identité  $4 \times 4$ . Définir l'algorithme de Metropolis Hastings pour un jeu de données  $\mathbf{y}$ .
3. Le fichier `donnees_presence_complet.txt` contient les observations de 300 sites sur lesquels la présence d'oiseaux a été constatée, ainsi que différentes variables environnementales mesurées. Ecrire un programme R codant l'algorithme de Metropolis Hastings précédent pour ce jeu de données. Vous testerez plusieurs valeurs de  $\sigma^2$  pour la variance de la marche aléatoire, et choisirez celle qui vous semble la meilleure.
4. Pour le  $\sigma^2$  choisi quelle est la probabilité d'acceptation empirique?
5. Quelle est la valeur réalisée de l'estimateur Bayésien  $\mathbb{E}[\theta | \mathbf{Y}]$ ?
6. Donner un intervalle de crédibilité pour chacun des paramètres.

## 3 Metropolis Hastings et loi de mélange

On s'intéresse à simuler à la loi d'un mélange de deux Gaussiennes:

$$f(x) = \frac{1}{2}f_Z(x-4) + \frac{1}{2}f_Z(x+4)$$

où  $f_Z$  est la densité d'une loi  $\mathcal{N}(0, 1)$ .

1. À l'aide du logiciel R, tracez la densité de cette loi.
2. On se propose de construire un algorithme de Metropolis Hastings pour simuler une chaîne de Markov de loi stationnaire  $f$  à partir d'une marche aléatoire d'étendue uniforme. Plus précisément, pour  $\alpha \in \mathbb{R}_+$ , à partir d'une position  $X = x$ , on définit la prochaine position  $Y | X = x$  comme une variable aléatoire de densité

$$q_\alpha(x, y) = \mathbf{1}_{x-\alpha \leq y \leq x+\alpha}$$

- a. Ecrire un code R permettant de simuler selon cette loi.

- b. Justifier que pour toute régions  $A$  et  $B$  de  $\mathbb{R}$ , on peut accéder de  $A$  à  $B$  en un nombre fini de pas.
3. Définir l'algorithme de Métropolis Hastings pour simuler une chaîne de Markov  $(X_n)_{n \geq 0}$ , de loi stationnaire  $f$  partir du noyau de Markov  $q_\alpha$  et d'un point de départ  $x_0$ .
4. Implémentez cet algorithme en R pour  $\alpha = 0.01$  à partir de  $x_0 = 0$ , pour  $n = 5000$ . Faites tourner cet algorithme plusieurs fois, que constatez vous?
5. Implémentez cet algorithme en R pour  $\alpha = 5$  et à partir de  $x_0 = -10$ . Faites tourner cet algorithme plusieurs fois, que constatez vous?
6. Dans les deux cas, donnez la probabilité d'acceptation empirique de l'algorithme (ou le nombre d'essai moyen avant de faire un pas dans la chaîne).

## 4 Echantillonneur de Gibbs

*Cet exercice est un exemple trivial d'implémentation (inutile ici!) de l'échantillonneur de Gibbs*

On veut simuler par échantillonneur de Gibbs un échantillon de vecteur aléatoire  $(X, Y)$  distribué selon la loi  $\mathcal{N}\left(0, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right)$ .

1. Donnez la loi conditionnelle de  $Y|X$  et  $X|Y$ .
2. Implémentez un échantillonneur de Gibbs partant du point  $(10, 10)$  pour simuler selon la loi jointe de  $(X, Y)$ .
3. *Burn-in*: Vérifiez empiriquement que l'algorithme converge vers la loi voulue. Quelle partie initiale de la chaîne simulée conseillez vous d'omettre?
4. *Thinning*: Regardez la fonction d'autocorrélation de l'échantillon simulé. Quelle proportion de l'échantillon préconisez vous de garder en pratique pour avoir un échantillon qu'on pourra supposer indépendant?

## 5 Décryptage bayésien

[1] "CECI SERA LE DERNIER DEVOIR DU COURS."

### 5.1 Présentation du problème

On se place dans le cadre où on dispose d'un alphabet de taille finie, disons  $K$ . Chaque élément de l'alphabet est codé comme un nombre l'ensemble  $\{1, \dots, K\}$ . On suppose qu'on dispose d'un message crypté de ce type:

[1] "NJNEFYJ OFXJFCJ 'EJ FCJAHE FCDNFHD YL"

L'objectif est de décrypter ce message, et retrouver le message original, à savoir:

[1] "CECI SERA LE DERNIER DEVOIR DU COURS."

en utilisant l'inférence bayésienne. Pour cela,

- On suppose que le message est issue d'une langue connue, disons le Français, dont on connaît certaines caractéristiques (décrites plus bas).
- On suppose que ce message est la transformation du vrai message par une permutation  $f_*^{-1}$  des éléments de l'alphabet. Ainsi,  $f_*^{-1}$  a envoyé chaque élément de la phrase initiale sur un autre élément de l'alphabet (deux éléments identiques dans la phrase de départ le sont encore dans la phrase d'arrivée).

On recherche donc la permutation  $f_*$  afin de décrypter le message. Cette inconnue vit donc dans un espace discret à  $K!$  éléments.

- Pour un message  $X$  et permutation  $f$  donnée, la vraisemblance associée à  $f$  est donnée par

$$L(X|f) = \prod_{i,j=1}^K M(i,j)^{f_X(i,j)},$$

où

- $M(i,j)$  est le nombre de transitions  $i \rightarrow j$  (pour chaque élément  $i$  et  $j$  de l'alphabet) observées *par ailleurs* dans la langue Française.
- $f_X(i,j)$  est le nombre de transitions de  $i \rightarrow j$  dans la décryption du message  $X$  par  $f$ . Par exemple, pour si  $f$  est l'identité, notre message précédent présente 2 transitions " $C \rightarrow J$ ", 0 transition " $A \rightarrow B$ ", 3 transitions " $J \rightarrow \text{' '}$ ", 0 transition " $J \rightarrow C$ ", etc. . .

On voit que cette fonction de vraisemblance est grande si la décryption de  $X$  par  $f$  présente une fréquence de transition constante avec celle de la matrice  $M$ , connue dans la langue française.

## 5.2 Objectif

On veut obtenir une vision des décryptions possibles par inférence bayésienne. On suppose que la loi a priori de  $f$  est une loi uniforme sur l'ensemble des permutations possibles. Au vu d'un message  $X$ , on cherche à obtenir des échantillons tirés selon la loi a posteriori de  $f$ . Pour cela, vous implémenterez un algorithme de Metropolis Hastings dont la loi stationnaire sera donnée par cette loi a posteriori.

- C'est vous qui choisirez le(s) point(s) de départ de cet algorithme en justifiant votre démarche.
- De même, c'est vous qui choisirez le noyau de transition de l'algorithme de Metropolis Hastings utilisé.
- L'objectif est d'obtenir des tirages dans la loi a posteriori. Vous présenterez plusieurs de ces tirages et vous en servirez pour essayer de décrypter au mieux votre texte.
- La démarche devra être décrite clairement et reproductible (donc vous fournirez vos codes).

Afin de vous aider dans la démarche, vous pourrez la très complète référence: *Decrypting Classical Cipher Text Using Markov Chain Monte Carlo* de Chen et Rosenthal.

## 5.3 Détails techniques

L'alphabet considéré sera celui composé de toutes les lettres majuscules, sans accent, de la langue française, auxquelles s'ajouteront l'apostrophe "'", la virgule ",", le point "." et l'espace " ". On indexera ces éléments de 1 (la lettre "A") à 30 (l'espace " ").

```
# Un alphabet à 30 éléments
(my_alphabet <- c(LETTERS, "'", ",", ".", " "))
```

```
[1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q"
[18] "R" "S" "T" "U" "V" "W" "X" "Y" "Z" " '" ", " . " " "
```

La matrice  $M$  transmise sera donc une matrice 30 par 30, indexée de la même manière. Ainsi,  $M(5, 29)$  comptera le nombre de transitions observées dans mon texte modèle (écrit en Français) entre le "E" et le ".".

Fatalement, l'exercice voudra que vous manipulier des chaînes de caractères avec R. Vous pourrez vous aider du package `stringr` pour lequel il existe de nombreux tutoriels (dont celui ci).