RPM4 2024 - TP sur les algorithmes MCMC

Clément Gauchy - clement.gauchy@cea.fr

INSTN - Novembre 2024

Algorithm 1: Échantilloneur de Metropolis-Hastings

Data: Attribuer une valeur initiale aux paramètres $\theta^{(0)} := (\theta_1^{(0)}, \ldots, \theta_p^{(0)})$. Initialiser la variable compteur $t := 0$ et le nombre max d'itérations T.

while $t < T$ **do**
- Proposer une valeur candidate $\theta^{(c)} = \mathcal{Q}_t(\theta|\theta^{(t-1)})$
- Calculer le ratio de Metropolis-Hastings: $r = \frac{[\theta^{(c)}|\mathbf{y}]\mathcal{Q}_t(\theta^{(t-1)}|\theta^{(c)})}{[\theta^{(t-1)}|\mathbf{y}]\mathcal{Q}_t(\theta^{(c)}|\theta^{(t-1)})}$
- Tirer $U \sim \mathcal{U}([0,1])$
- **if** $U < \min(1, r)$ **then**
  - $\theta^{(t)} \leftarrow \theta^{(c)}$
- **else**
  - $\theta^{(t)} \leftarrow \theta^{(t-1)}$
- $t \leftarrow t + 1$

Introduction

L'objectif de ce TP est d'apprendre à utiliser en pratique les algorithmes de Monte Carlo par chaîne de Markov (MCMC pour *Markov Chain Monte Carlo*). Ces algorithmes sont particulièrement adaptées à l'échantillonage d'une densité de probabilité connuesà une constante près. C'est un cas que l'on retrouve fréquemment en physique statistique et en inférence Bayésienne.

Exercice 1: Algorithme de Metropolis-Hastings

L'objectif de l'exercice est d'effectuer un échantillonage d'une loi normale avec comme seul outil une loi uniforme

1. Expliquer en quoi la fonction `loi_normale` dans le notebook Jupyter ci-joint nous suffira pour simuler la loi normale centrée réduite.

2. On considère le noyau de transition $\mathcal{Q}(y|x) = \mathbf{1}_{x-\alpha<y<x+\alpha}$. Montrez que ce noyau est symmétrique.

4. Exécuter l'algorithme `MetropolisHastings` avec plusieurs valeurs de talle de l'échantillon `n`

5. L'amplitude d'exploration des valeurs candidates α joue un rôle crucial dans l'autocorrelation de la série générée. Quelles sont les conséquences de son augmentation? De sa diminution?

6. Modifier la fonction `MetropolisHastings` afin qu'elle permette de calculer le taux d'acceptation des valeurs candidates.

7. Tracer en fonction de quelques valeurs de α le taux d'acceptation de l'algorithme

8. Une pratique standard suggère de régler le taux d'acceptation à 40% pour une mise à jour univariée et 20% pour une mise à jour multivariée des paramètres. Dans ce contexte, quelle(s) valeur(s) de α vous semble(nt) "optimale(s)" ?

Exercice 2: Estimation Bayésienne en imagerie TEP

On rappelle ici le modèle statistique de la Tomographie à Émission de Position (TEP)

- **Sources de photons** $1 \leq j \leq m$, loi de Poisson de paramètre $\lambda_j : X_j \sim \mathcal{P}(\lambda_j)$. X_j est le nombre total de photons émis par la source j.

- **Capteurs des photons**, $1 \leq i \leq n$ et p_{ij} = probabilité que le capteur i détecte le photon j avec $\sum_{j=1}^{m} p_{ij} = 1$

- N_{ij} : Nombre de photons émis par la source j puis détectés par le capteur i :

$$(N_{ij}|X_j) \sim \mathcal{B}(X_j, p_{ij})$$

  $\hookrightarrow$ Loi conditionelle $N_{ij}|X_j$ binomiale

- Nombre de photons émis par toutes les sources puis détectés par le capteur i

$$Y_i = \sum_{j=1}^{m} N_{ij}$$

Le but en imagerie TEP est d'estimer les intensités $(\lambda_j)_{1 \leq j \leq m}$ avec comme unique données les $(Y_i)_{1 \leq i \leq n}$. Dans cet exercice, on utilisera un formalisme bayésien pour l'estimation de $\Lambda = (\lambda_j)_{1 \leq j \leq m}$.

1. On se place dans le cas simplifié où $m = 2$. Écrivez la vraisemblance $[Y|\Lambda]$ et codez en Python une fonction `likelihood` qui retourne la valeur de la vraisemblance en Λ.

2. On suppose que $\lambda_1 \sim \Gamma(a_1, b)$ et $\lambda_2 \sim \Gamma(a_2, b)$ de façon indépendante. Calculez la loi a posteriori $[\Lambda|Y]$ à une constante multiplicative près. Codez en Python la fonction `postfunc` qui renvoit cette valeur.

3. On va échantilloner Λ selon la loi a posteriori en utilisant un algorithme de Metropolis Hastings. On utilise comme loi de transition $\Lambda^{(t+1)} \sim \mathcal{LN}(\Lambda^{(t)}, \sigma^2 I_2)$. Codez la fonction Python `MetropolisHastings_TEP` correspondante

4. Lancez 2 ou 3 chaines de Markov avec $\sigma = 1$ et différentes valeurs initiales, faites un examen visuel des résultats.

5. Testez plusieurs valeurs de σ et choisissez celle qui vous semble la meilleur. Quelle est la probabilité d'acceptation associée ?

6. Complétez le fichier `tep_exo.stan` pour implémenter en Stan le modèle statistique de l'exercice. L'installation de Stan est décrite ici