

Introduction aux méthodes de Monte Carlo par dynamique Hamiltonienne

Shmuel RAKOTONIRINA-RICQUEBOURG, Amaury DURAND

13 novembre 2017

Table des matières

1	Introduction	1
1.1	Méthodes de Monte Carlo	1
1.2	Monte Carlo Markov Chains (MCMC)	1
1.2.1	Loi invariante et réversibilité	2
1.2.2	Ergodicité	2
1.2.3	Algorithme de Metropolis-Hastings	2
2	Hamiltonian Monte Carlo	3
2.1	Dynamique Hamiltonienne	3
2.2	Principe général	3
2.3	Outils nécessaires	4
2.4	Propriétés de l'algorithme	4
3	Dynamique hamiltonienne	4
4	Algorithme et simulations	4

1 Introduction

Ce rapport présente le travail effectué lors d'un projet du cours d'approfondissements en chaînes de Markov par Eric Moulines dans le cadre du master Mathématiques de l'aléatoires à l'université Paris-Sud. Le contenu présenté ci-dessous repose essentiellement sur [1] et [2].

Les méthodes de Monte Carlo par dynamique Hamiltonienne, plus communément appelées Hamiltonian Monte Carlo (HMC), font partie d'une grande famille de méthodes de simulation : les méthodes de Monte Carlo, et plus précisément dans la famille des méthodes de Monte Carlo par chaînes de Markov (ou Monte Carlo Markov Chains). Ces méthodes se placent dans le cadre suivant :

Définition 1.1 (Carde général). Soit $(\mathbb{X}, \mathcal{X})$ un espace mesurable. Soit π une loi de probabilité sur cet espace. On suppose que π n'est connue qu'à un facteur de proportionnalité près i.e on connaît $\lambda\pi$ où $\lambda \in \mathbb{R}$ une constante. Le but est de pouvoir approcher $\pi f = \mathbb{E}[f(X)]$ avec $X \sim \pi$ et $f \in \mathbb{F}_+(\mathbb{X}, \mathcal{X}) \cup \mathbb{F}_b(\mathbb{X}, \mathcal{X})$.

1.1 Méthodes de Monte Carlo

La méthode de Monte Carlo la plus simple (appelée Monte Carlo naïf) est la suivante : supposons que l'on sait simuler des variables aléatoires de loi π alors en tirant n échantillons $X_1, \dots, X_n \stackrel{\text{iid}}{\sim} \pi$, la loi des grands nombre nous indique qu'une bonne approximation de πf est $\frac{1}{n} \sum_{i=1}^n f(X_i)$. D'autres méthodes permettent d'atteindre le même but en ne sachant pas simuler de variables aléatoires de loi π . C'est le cas par exemple de l'échantillonnage d'importance qui consiste à simuler des variables i.i.d sous une loi différente de π .

Dans ces deux cas, l'approximation de πf repose sur une simulation de variables aléatoires i.i.d. Cette particularité permet alors de montrer des résultats de convergence notamment grâce à la loi des grands nombre. Les méthodes de Monte Carlo par chaînes de Markov ne reposent pas sur le caractère i.i.d des variables mais sur les propriétés des Chaînes de Markov.

1.2 Monte Carlo Markov Chains (MCMC)

Les méthode MCMC se basent sur les notions de loi invariante, de réversibilité et d'ergodicité.

1.2.1 Loi invariante et réversibilité

On considère P un noyau de Markov sur $\mathbb{X} \times \mathcal{X}$.

Définition 1.2. Une loi de probabilité π sur $(\mathbb{X}, \mathcal{X})$ est dite

- P -invariante si $\pi P = \pi$
- P -réversible si $\forall A, B \in \mathcal{X}, \pi \otimes P(A \times B) = \pi \otimes P(B \times A)$

Proposition 1.1. Toute probabilité P -réversible est P -invariante.

1.2.2 Ergodicité

Théorème 1.1. Soit $(X_k)_{k \in \mathbb{N}}$ une chaîne de Markov de noyau P admettant une loi invariante π . Alors pour tout $f \in \mathbb{F}_+(\mathbb{X}, \mathcal{X}) \cup \mathbb{F}_b(\mathbb{X}, \mathcal{X})$ et pour π -presque tout $x \in \mathbb{X}$,

$$\frac{1}{n} \sum_{k=0}^{n-1} f(X_k) \xrightarrow[n \rightarrow +\infty]{\mathbb{P}_x - p.s.} \pi f$$

Les algorithmes MCMC visent à construire, à partir de π , une chaîne de Markov vérifiant les hypothèses de ce théorème afin d'approcher πf par $\frac{1}{n} \sum_{k=0}^{n-1} f(X_k)$.

1.2.3 Algorithme de Metropolis-Hastings

L'algorithme HMC est une version améliorée de l'algorithme de Metropolis-Hastings. Nous allons donc d'abord définir ce dernier et nous l'utiliserons plus tard comme base de comparaison. On se place dans le cas où $\mathbb{X} = \mathbb{R}^d$ et $\mathcal{X} = \mathcal{B}(\mathbb{R}^d)$.

On suppose que π a une densité h_π par rapport à une mesure μ . On considère de plus un noyau markovien Q sur $(\mathbb{X}, \mathcal{X})$ de densité q par rapport à μ . Q est appelé noyau instrumental. La construction de la chaîne de Markov se fait en proposant un mouvement via Q , puis en acceptant ou en rejetant ce mouvement (algorithme 1.1).

Algorithme 1.1 : Metropolis-Hastings

Données : h_π proportionnel à la densité cible, Q noyau markovien simulable

$X_0 \leftarrow x \in \mathbb{X}$ arbitraire;

répéter

$Y_{k+1} \sim Q(X_k, \cdot)$; // Proposer un mouvement

$\alpha_{k+1} \leftarrow \alpha(X_k, Y_{k+1})$ où $\alpha(x, y) = 1 \wedge \frac{h_\pi(y)q(y, x)}{h_\pi(x)q(x, y)}$;

$X_{k+1} \leftarrow \begin{cases} Y_{k+1} & \text{avec probabilité } \alpha_{k+1} \\ X_k & \text{avec probabilité } 1 - \alpha_{k+1} \end{cases}$; // Accepter ou rejeter le mouvement

jusqu'à une condition d'arrêt;

retourner $(X_k)_k$

Dans cet algorithme, on cherche à explorer l'espace \mathbb{X} en visitant moins souvent les régions où h_π est faible (peu chargées par π) car ce sont les régions de forte probabilité qui donnent des informations sur la loi π . Ainsi, si le mouvement proposé vérifie $h_\pi(Y_{k+1})q(Y_{k+1}, X_k) \geq h_\pi(X_k)q(X_k, Y_{k+1})$, on se dirige vers une zone plus chargée par π , on accepte donc le mouvement. Dans le cas contraire, il est moins intéressant de bouger. On autorise quand même le mouvement avec une probabilité $\frac{h_\pi(Y_{k+1})q(Y_{k+1}, X_k)}{h_\pi(X_k)q(X_k, Y_{k+1})}$ d'autant plus faible que la position proposée est dans une région de probabilité faible.

Remarque 1.1. Il y a d'autres fonctions de rejet α qui permettent à l'algorithme de fonctionner. De même, le choix du noyau Q est un degré de liberté de l'algorithme.

Le choix de Q est en fait une difficulté de l'algorithme, puisque le mouvement proposé doit permettre l'exploration de tout l'espace (afin de passer régulièrement par toutes les régions fortement chargées par π). Nous verrons que l'algorithme HMC résout ce problème en proposant le mouvement selon une *dynamique hamiltonienne*.

Définition 1.3. On parle de *Random-walk Metropolis* quand Q est le noyau d'une marche aléatoire

$$Y_{k+1} \leftarrow X_k + U_{k+1}$$

avec $(U_k)_{k \in \mathbb{N}}$ iid. On supposera que Q admet une densité q par rapport à μ avec $q(x) = q(-x)$.

Algorithme 1.2 : Random Walk Metropolis

Données : h_π proportionnel à la densité cible, Q loi simulable

$X_0 \leftarrow x \in \mathbb{X}$ arbitraire;

$(U_k)_{k \in \mathbb{N}} \stackrel{\text{iid}}{\sim} Q$;

répéter

$Y_{k+1} \sim X_k + U_{k+1}$; // Proposer un mouvement

$\alpha_{k+1} \leftarrow \alpha(X_k, Y_{k+1})$ où $\alpha(x, y) = 1 \wedge \frac{h_\pi(y)}{h_\pi(x)}$;

$X_{k+1} \leftarrow \begin{cases} Y_{k+1} & \text{avec probabilité } \alpha_{k+1} \\ X_k & \text{avec probabilité } 1 - \alpha_{k+1} \end{cases}$; // Accepter ou rejeter le mouvement

jusqu'à une condition d'arrêt;

retourner $(X_k)_k$

2 Hamiltonian Monte Carlo

On se place dans le cas où $\mathbb{X} = \mathbb{R}^d$, $\mathcal{X} = \mathcal{B}(\mathbb{R}^d)$ et μ est la mesure de Lebesgue.

L'algorithme HMC fait partie des algorithmes MCMC. Il diffère de l'algorithme de Metropolis-Hastings en ce que le mouvement proposé ne sera pas tiré selon une chaîne de Markov de noyau Q mais selon une *dynamique hamiltonienne* associée à la loi à simuler.

2.1 Dynamique Hamiltonienne

Un système physique de position $x : \mathbb{R}_+ \rightarrow \mathbb{R}^d$ et de quantité de mouvement $p : \mathbb{R}_+ \rightarrow \mathbb{R}^d$ (toutes deux fonctions du temps t) est caractérisé par des équations de mouvement de type $\forall t \geq 0, \forall i \in \llbracket 1, d \rrbracket$,

$$\begin{aligned} \frac{dx_i}{dt}(t) &= \frac{\partial H}{\partial p_i}(q(t), p(t)) \\ \frac{dp_i}{dt}(t) &= -\frac{\partial H}{\partial q_i}(q(t), p(t)) \end{aligned}$$

où $H : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ s'appelle le *Hamiltonien* du système.

Le Hamiltonien s'interprète comme une énergie sur x et p . En général $H(x, p) = U(x) + K(p)$ où U est l'énergie potentielle et K l'énergie cinétique.

2.2 Principe général

La dynamique hamiltonienne décrit le mouvement d'un objet qui glisse sans frottement le long d'une surface ou d'une courbe. L'objet est décrit par sa position $x \in \mathbb{R}^d$ et sa quantité de mouvement $p \in \mathbb{R}^d$ et on lui associe des énergies potentielle $U(x)$ et cinétique $K(p)$.

Définition 2.1 (Distribution canonique). En physique, une énergie $E : \mathbb{R}^n \rightarrow \mathbb{R}$ et une température $T \in \mathbb{R}$ sont associées à une loi de probabilité par la loi de Boltzmann (ou distribution canonique) de densité par rapport à la mesure de Lebesgue sur \mathbb{R}^n ,

$$\forall x \in \mathbb{R}^n, p(x) \propto \exp\left(-\frac{E(x)}{T}\right)$$

Dans le cas des HMC, on étend l'espace d'états $\mathbb{X} = \mathbb{R}^d$ à $\mathbb{R}^d \times \mathbb{R}^d$ et on va construire une chaîne de Markov $(X_k, P_k)_{k \in \mathbb{N}}$ à valeurs dans ce nouvel espace avec $\forall k \in \mathbb{N}, X_k \perp P_k$ et de loi invariante $\pi \otimes \nu$ où π est une distribution canonique associée à l'énergie potentielle et ν est une distribution canonique associée à l'énergie cinétique. Ainsi leurs densités h_π et h_ν par rapport à la mesure de Lebesgue sur \mathbb{R}^d vérifient $\forall x \in \mathbb{R}^d, \forall p \in \mathbb{R}^d$,

$$\begin{aligned} h_\pi(x) &\propto \exp\left(-\frac{U(x)}{T}\right) \\ h_\nu(p) &\propto \exp\left(-\frac{K(p)}{T}\right) \end{aligned}$$

Ce qui donne une loi jointe de densité p telle que $\forall (x, p) \in \mathbb{R}^d \times \mathbb{R}^d$,

$$p(x, p) \propto \exp\left(-\frac{H(x, p)}{T}\right) \quad (1)$$

où le Hamiltonien $H(x, p) = U(x) + K(p)$ est choisi.

Remarque 2.1. En dimension 1, on décrit le glissement sans frottement d'un objet sur une rampe. Comme l'énergie potentielle est proportionnelle à la hauteur de la rampe, (1) donne que les creux de la rampe représentent les régions de forte probabilité. L'objet aura tendance à glisser vers les creux de la rampe, mais peut remonter une pente si sa quantité de mouvement p (et donc son énergie cinétique $K(p)$) est assez grande.

L'algorithme HMC propose ses mouvements selon le principe suivant

1. Considérer un objet placé en X_k de quantité de mouvement P_k
2. Simuler la dynamique hamiltonienne (déterministe) de cet objet pendant une durée fixée, pour l'énergie potentielle U donnée par (1)
3. Considérer Y_{k+1} la position de cet objet à la fin de la simulation, et accepter ou rejeter le mouvement selon une fonction de rejet α dépendant de U

2.3 Outils nécessaires

2.4 Propriétés de l'algorithme

3 Dynamique hamiltonienne

Algorithme 3.1 : Discrétisation de l'évolution par saute-mouton (*leapfrog*)

Données : pas ϵ , nombre de pas L , état initial (x_0, p_0)

$(x, p) \leftarrow (x_0, p_0);$

$p \leftarrow p - \frac{\epsilon}{2} \nabla U(x);$ // Demi-pas en p

pour $k \in \llbracket 1, L-1 \rrbracket$ **faire** // Saute-mouton

$x \leftarrow x + \epsilon p;$

$p \leftarrow p - \epsilon \nabla U(x);$

$x \leftarrow x + \epsilon p;$ // Dernier pas en x

$p \leftarrow p - \frac{\epsilon}{2} \nabla U(x);$ // Demi-pas en p

retourner (x, p)

4 Algorithme et simulations

Algorithme 4.1 : Hamiltonian Monte-Carlo

Données : h_π proportionnel à la densité cible, ϵ pas du saute-mouton, L nombre de pas du saute-mouton

$X_0 \leftarrow x \in \mathbb{X}$ arbitraire;

répéter

$P_k \sim \mathcal{N}(0, 1);$ // Tirer la quantité de mouvement

$(X_{prop}, P_{prop}) \leftarrow \text{leapfrog}(X_k, P_k);$ // Proposer un mouvement

$U_k \leftarrow U(X_k);$

$K_k \leftarrow \|P_k\|^2/2;$

$U_{prop} \leftarrow U(X_{prop});$

$K_{prop} \leftarrow \|P_{prop}\|^2/2;$

si $\mathcal{U}([0, 1]) < \exp(U_k - U_{prop} + K_k - K_{prop})$ **alors**

$X_{k+1} \leftarrow X_{prop};$ // Accepter

sinon

$X_{k+1} \leftarrow X_k;$ // Rejeter

jusqu'à une condition d'arrêt;

retourner $(X_k)_k$

Références

- [1] Neal Radford M. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 54 :113–162, 2010.
- [2] Douc Randal ; Moulines Eric ; Priouret Pierre ; Soulier Philippe. *Markov Chains*. 2017.