Université Paris Descartes Master MA

Algorithmes Stochastiques

Méthodes de Monte Carlo avec Chaînes de Markov (MCMC)

Corrigés mis en ligne au fur et à mesure à l'adresse http://www.cmapx.polytechnique.fr/~benaych/M2AlgosStos/

1 Metropolis-Hasting versus rejet

Soit f la densité (sur un espace $\mathcal{X}$ quelconque, par rapport à une mesure de référence μ) sous laquelle on cherche à simuler, appelée *densité cible*. On considère une autre densité g (par rapport à μ), appelé *densité instrumentale* ou *densité de proposition*, telle que :

- il est aisé de simuler suivant g,
- il existe une constante M telle que $f(x) \leq Mg(x)$ pour tout x (ce qui implique que $M \geq 1$ et $\mathrm{supp}(f) \subset \mathrm{supp}(g)$).

On peut alors générer un échantillon suivant l'algorithme suivant :

Algorithme du rejet :

- Générer X suivant la densité g.
- Générer U suivant une loi $\mathcal{U}([0, M])$
- Accepter la valeur X si $U \leq \frac{f(X)}{g(X)}$, sinon, recommencer.

On a par ailleurs vu en cours l'algorithme de Metropolis-Hasting. Soit f une densité de probabilité sur $\mathbb{R}^d$. Posons $\mathrm{supp}_>(f) := \{x\,;\, f(x) > 0\}$.

Algorithme de Metropolis-Hastings de simulation de X de densité $\approx f$:

- Choisir un noyau de transition Q irréductible sur $\mathrm{supp}_>(f)$.
- Choisir $x \in \mathrm{supp}_>(f)$.
- Répéter un grand nombre de fois :
  - ⋆ tirer, indépendamment, $Y \sim Q(x, \cdot)$ et $U \sim \mathcal{U}([0,1])$,
  - ⋆ remplacer x par Y si $U < \rho(x, Y)$ et le laisser invariant sinon, où

$$\rho(x,y) := \frac{f(y)Q(x,y)}{f(x)Q(y,x)}.$$

- Rendre $X := x$.

Exercice 1. Le but ici est de générer des échantillons selon la loi sur $\mathbb{R}$ de densité :

$$\pi_\alpha(x) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-x^2/2}(1 + \sin(\alpha x)),$$

où $\alpha > 0$ (on pourra prendre $\alpha = 2$).

1. Tracer la densité π_α pour plusieurs valeurs de α.
2. Simuler un échantillon de taille $n = 1000$ de vaiid de loi de densité π_α via la méthode MH avec une loi de proposition $Q(x, \cdot) = \mathcal{N}(x, \sigma^2)$ (pour $\sigma = 1$ par exemple). Simuler ensuite, en dédiant à la tâche autant de temps, un échantillon de taille par méthode de rejet (on notera que $\pi_\alpha(x) \le 2g(x)$ pour g la densité de $\mathcal{N}(0,1)$). Comparer les histogrammes et afficher le rapport entre les tailles des deux échantillons.
3. Calculer ensuite la valeur théorique de l'espérance ainsi que sa valeur empirique calculée grâce aux deux méthodes.

2 Modèle d'Ising via MH

Le *modèle d'Ising* offre un cadre théorique très simple pour décrire les transitions de phase de l'aimantation d'un métal ferromagnétique. À chaque site v du réseau $\Gamma = \{1, \ldots, N\}^2$, on associe un spin s_v prenant les valeurs ± 1 et on note $S = (s_v)_{v \in \Gamma}$ une configuration de spins. Les spins interagissent avec leurs plus proches voisins et une énergie est attribuée à chaque configuration S est

$$H(S) = -\sum_{u \sim v} s_u s_v$$

où $u \sim v$ signifie que les sites u et v sont à distance 1 sur le réseau Γ. Un système physique a tendance à minimiser son énergie ce qui permet de distinguer deux configurations privilégiées (les états fondamentaux) : les spins sont tous égaux à 1 ou tous égaux à -1. Pour tenir compte des fluctuations thermiques, on définit la mesure de Gibbs qui attribue à la configuration S la probabilité

$$\mu_\beta(S) = \frac{1}{Z_\beta} \mathrm{e}^{-\beta H(S)},$$

où la fonction de partition Z_β sert à normaliser la mesure de Gibbs. Le paramètre

$$T := \frac{1}{\beta}$$

s'interprète comme une température : quand T est grand les fluctuations thermiques dominent et le système est désordonné, par contre pour T proche de 0 les configurations de basse énergie sont privilégiées et les spins ont tendance à s'aligner. Ce modèle très simple de spins en interaction permet de mettre en évidence l'existence d'une transition de phase quand la taille du domaine N tend vers l'infini. Les transitions de phase constituent une source de questions fascinantes. Nous allons implémenter l'algorithme de Metropolis-Hasting afin de simuler le modèle d'Ising. Retraduit dans le formalisme des chaînes de Markov, une configuration S correspond à un état et l'espace d'états est $E = \{-1, 1\}^\Gamma$. Pour un domaine bi-dimensionnel de taille $N = 40$, le cardinal de E est $2^{40\times 40} \approx 10^{481}$. Il est donc impossible d'énumérer toutes les configurations pour calculer la distribution μ_β. Pour tout $v \in \Gamma$, on note $S(v)$ la configuration déduite de S en changeant simplement le signe du spin en v. La matrice de référence Q décrit une évolution sur l'espace des configurations

$$\forall v \in \Gamma, \qquad Q(S, S(v)) = \frac{1}{\mathrm{Card}(\Gamma)}.$$

Elle correspond au mécanisme suivant : un site v est choisi au hasard dans Γ et son spin dans S est retourné. Ce sont les seules transitions autorisées. Ces transitions modifient les configurations seulement localement, par conséquent la variation de l'énergie correspondant au changement du spin en v ne dépend que de la moyenne des spins autour de v :

$$\begin{aligned}
\Delta H(S, v) &:= H(S(v)) - H(S) \\
&= -\sum_{w\sim v} s_w(-s_v) - \left(-\sum_{w\sim v} s_w s_v\right) \\
&= 2s_v \sum_{w\sim v} s_w.
\end{aligned}$$

La fonction ρ est alors donnée par

$$\rho(S, S(v)) = \mathrm{e}^{-\beta \Delta H(S,v)}.$$

On en déduit :

> **Algorithme de Metropolis-Hastings de simulation de $S \approx \mu_\beta$:**
>
> - Initialiser avec une configuration S quelconque.
> - Répéter un grand nombre de fois :
>   - ⋆ tirer, indépendamment, v de loi unif. sur Γ et $U \sim \mathcal{U}([0, 1])$,
>   - ⋆ remplacer S par $S(v)$ si $U < \mathrm{e}^{-\beta\Delta H(S,v)}$ et le laisser invariant sinon.
> - Rendre S.

Exercice 2. Implémenter l'algorithme précédent et le faire tourner pour diverses configurations initiales. Afficher les résultats au moyen de la fonction **plt.imshow**.

3 Voyageur de commerce

Considérons à présent le problème du voyageur de commerce pour n villes positionnées dans l'espace en $V_1, \ldots, V_n$, qui consiste à trouver une tournée de longueur minimale, c'est-à-dire une permutation $x \in S_n$ dans le groupe symétrique S_n qui minimise la fonction

$$x \in S_n \longmapsto H(x) := \sum_{i=1}^{n} \mathrm{dist}(V_{x(i)}, V_{x(i+1)}) \qquad \text{où } x(n+1) := x(1).$$

La longueur d'une tournée joue le rôle d'énergie ici.

Une méthode simple consisterait à calculer $H(x)$ pour toutes les permutations x. Mais lorsque n est élevé, cette méthode est irréalisable. C'est le cas dès que le nombre de villes dépasse la dizaine : le cardinal de S_n est de $n!$ (par exemple, pour $n = 30$, $n! \simeq 2.10^{32}$).

Dans ce cas, l'idée est alors d'appliquer l'algo de recuit simulé en se déplaçant sur S_n aléatoirement de proche en proche, ce qui implique d'avoir, sur S_n, une notion d'*éléments voisins* (de *permutations voisines*). Par exemple, on peut considèrer comme voisines deux permutations qui se déduisent l'une de l'autre par la permutation de deux éléments seulement. Cette "notion de voisinage" fonctionne, mais dans notre algorithme, nous la remplaçons en fait par une autre, qui permet une convergence plus rapide : nous considérons deux permutations $(x_1, \ldots, x_n)$, $(y_1, \ldots, y_n)$ comme voisines si il existe $1 \le i < k \le n$ tels que

$$(y_1, \ldots, y_n) = (x_1, \ldots, x_{i-1}, x_k, x_{k-1}, \ldots, x_{i+1}, x_i, x_{k+1}, \ldots, x_n).$$

Par exemple, pour $n = 8$, les permutations

$$(1, 2, 3, 4, 8, 7, 6, 5) \quad \text{et} \quad (1, 2, 7, 8, 4, 3, 6, 5)$$

sont voisines (avec $i = 3, k = 6$).

On rappelle l'algorithme du recuit simulé de détermination de $x_{\min}$ tel que $H(x_{\min}) \approx \min_E H$.

Algorithme du recuit simulé :

- Choisir un noyau de transition Q irréductible apériodique sur E vérifiant $Q(x, y) = Q(y, x)$ et une fonction $t \mapsto \beta_t$ tendant vers $+\infty$ lentement comme, par exemple $c \log t$.
- Choisir $x_0 \in E$.
- Répéter un grand nombre de fois :
  - ⋆ tirer, indépendamment, $Y \sim Q(x_t, \cdot)$ et $U \sim \mathcal{U}([0, 1])$,
  - ⋆ poser $x_{t+1} = y$ si $U \le \mathrm{e}^{-\beta_t(H(y) - H(x))}$ et $x_{t+1} = x_t$ sinon.
- Rendre $x_{\min} := x_t$.

Exercice 3. Implémenter cet algo pour trouver une solution au problème du voyageur de commerce, pour diverses configurations de villes.