Projet de simulations aléatoires : Modèles d'Ising

Aurélien Enfroy, Shmuel Rakotonirina--Ricquebourg

6 janvier 2018

1 Implémentation du modèle d'Ising

On rappelle la définition du modèle d'Ising sur un réseau carré :

Définition 1.1. On fixe C le réseau carré de dimension 2 de taille N^2. Le modèle d'Ising est la distribution sur l'espace d'état $\{\pm 1\}^C$ dont la loi est donnée par

$$\forall x \in \{\pm 1\}^C, \pi(x) = \frac{1}{Z_T} \exp\left(\left(\sum_{u\sim v} J_{u,v} x_u x_v + \sum_u h_u x_u\right)/T\right)$$

où $T > 0$ est appelée la température, Z_T est une constante de normalisation, $J_{u,v}$ est la force d'interaction entre u et v et h_u est le champ magnétique extérieur en u.

Pour l'implémentation, on remarque qu'il n'y a pas besoin du paramètre T, qu'on peut compter dans J et h. On représente alors ces paramètres en prenant $x \in \mathcal{M}_{N,N}(\{\pm 1\})$, $h \in \mathcal{M}_{N,N}(\mathbb{R})$ et J comme une matrice à trois entrées $\widetilde{J} \in \mathcal{M}_{N,N,2}(\mathbb{R})$ où

$$\widetilde{J}_{i,j,1} = J_{(i,j),(i+1,j)} \text{ et } \widetilde{J}_{i,j,2} = J_{(i,j),(i,j+1)}.$$

2 Simulation naïve en petite taille

En petite taille, on peut faire une simulation naïve. Pour cela, on numérote les 2^{N^2} états (dans l'ordre lexicographique en lisant les matrices colonne par colonne), on calcule exactement la loi et on choisit un numéro en utilisant cette loi et une loi uniforme sur $[0, 1]$.

3 Simulation par Metropolis-Hastings

Nous avons implémenté l'algorithme de Metropolis-Hastings en utilisant la fonction de rejet de Metropolis-Hastings. Pour le choix du noyau instrumental Q, nous avons choisi de donner à une coordonnée aléatoire (uniforme)

une valeur aléatoire (uniforme), ce qui permet d'avoir un noyau symétrique irréductible apériodique (probabilité strictement positive de rester sur place).

Avec ce noyau instrumental, on peut réduire les calculs effectués : si x est un état et y l'état proposé à partir de x, alors

— soit $y = x$, auquel cas on peut considérer qu'on accepte le mouvement,

— soit $y \neq x$, auquel cas ils ne diffèrent que d'au plus une coordonnée (celle tirée uniformément). Si on note u cette coordonnée et s la nouvelle valeur, alors $s = y_u = -x_u$ et, en considérant que T a été pris en compte dans J et h,

$$\begin{aligned}\frac{\pi(y)}{\pi(x)} &= \frac{\exp\left(\sum_{v\sim u} J_{u,v} s x_v + h_u s\right)}{\exp\left(\sum_{v\sim u} J_{u,v}(-s) x_v + h_u(-s)\right)} \\ &= \frac{e^{sV_u(x)}}{e^{-sV_u(x)}} \\ &= e^{2sV_u(x)}\end{aligned}$$

où on a noté $V_u(x) \doteq \sum_{v\sim u} J_{u,v} x_v + h_u$.

4 Simulation par l'échantilloneur de Gibbs

En reprenant les notations du cours, on a pour $u \in C$ et $x \in \{\pm 1\}^C$, en considérant que J et h dépendent de T,

$$\begin{aligned}\pi_u(x_u \mid x^u) &= \frac{\pi(x_u, x^u)}{\pi(1, x^u) + \pi(-1, x^u)} \\ &= \frac{e^{x_u V_u(x)}}{e^{V_u(x)} + e^{-V_u(x)}}\end{aligned}$$

où on note encore $V_u(x) \doteq \sum_{v\sim u} J_{u,v} x_v + h_u$. Donc $\pi_u(\cdot \mid x^u) = \mathcal{B}(\frac{e^{V_u(x)}}{e^{V_u(x)}+e^{-V_u(x)}}) = B(\frac{1}{1+e^{-2V_u(x)}})$

5 Couplage par le passé

Pour le couplage par le passé, deux fonctions d'actualisation sont possibles. Celle vue en cours est celle issue de l'échantillonneur de Gibbs (par balayage séquentiel). L'autre, présente dans le livre de Propp-Wilson, est issue de l'algorithme de Metropolis-Hastings (i.e. même transition qu'en section 3).

L'algorithme consiste à répéter les algorithmes vers le futur jusqu'à un temps donné et, si il n'y a pas eu coalition, ajouter des nouvelles actualisations au début du processus, ce qui demande de recommencer du début. Pour gagner du temps, plutôt que d'ajouter une actualisation à chaque fois, on double le nombre d'actualisations.

Pour les deux méthodes (Metropolis-Hastings et échantillonneur de Gibbs), les choses se passent bien au-delà de la température critique. Cependant, en dessous de la température critique, les états $\pm(1,\dots,1)$ ont tendance à rester inchangés par les fonctions d'actualisation précédentes : dans les deux cas, la probabilité de réussir à changer un point x_u égaux à ses quatre voisins est très faible : pour $J/T = \alpha$ constant, $h = 0$, elle vaut $\exp(-8\alpha)$.

En conséquence, l'algorithme met extrêmement longtemps à terminer (nous n'avons pas vu terminer). En observant l'évolution des chaînes de Markov (parties de $(1,\dots,1)$ et de $(-1,\dots,-1)$), on voit que souvent les deux chaînes reviennent à leur état initial, ce qui signifie que tout ce qui a été simulé jusqu'ici a été une perte de temps.