Stochastic Simulation

Autumn Semester 2024

Prof. Fabio Nobile *Assistant: Matteo Raviola*

Lab 10 – 21 November 2024

Markov Chains Monte Carlo

Exercise 0

Recall the Metropolis-Hastings algorithm 1.

Algorithm 1 Metropolis-Hastings Algorithm

Require: λ (initial distribution), Q (proposal distribution), π (target distribution)

1: Generate $X_0 \sim \lambda$
2: **for** $n = 0, 1, \dots$ **do**
3: Generate candidate new state $\tilde{X}_{n+1} \sim Q(X_n, \cdot)$
4: Generate $U \sim \mathcal{U}([0,1])$
5: **if** $U \leq \alpha(X_n, \tilde{X}_{n+1})$ **then**
6: Set $X_{n+1} = \tilde{X}_{n+1}$ {Candidate accepted with probability α}
7: **else**
8: Set $X_{n+1} = X_n$ {Candidate rejected with probability $1-\alpha$}
9: **end if**
10: **end for**

---

1. Compute the transition matrix of the Markov chain $\{X_n\}_{n\geq 0}$ generated by Algorithm 1.
2. Show that the transition matrix is in detailed balance with π.

Exercise 1

Let us consider a 2D uniform square-lattice with atoms placed at each vertex, as is sketched in Figure 1. The atoms can have an upward (red arrow) or a downward (blue arrow) pointing magnetic moment (so-called *spin*). Specifically, let the lattice be made out of $m \times m$ atoms. Therefore the system's possible states are the 2^{m^2} possible spin choices for the m^2 atoms. That is, the spin of the atom at position (i, j) in the lattice is denoted with s_{ij}, $1 \leq i, j \leq m$, and can take a value in $\{-1, +1\}$. A specific system configuration is described by the matrix $\boldsymbol{S} = (s_{ij}) \in \{-1, +1\}^{m \times m}$, containing the spin of each of the m^2 atoms.

The energy of a given system state of this Ising model is given by

$$H(\boldsymbol{S}) = -\sum_{i,j=1}^{m} \left(\frac{1}{2} J s_{ij}(s_{i-1,j} + s_{i+1,j} + s_{i,j-1} + s_{i,j+1}) + B s_{ij} \right) , \tag{1}$$

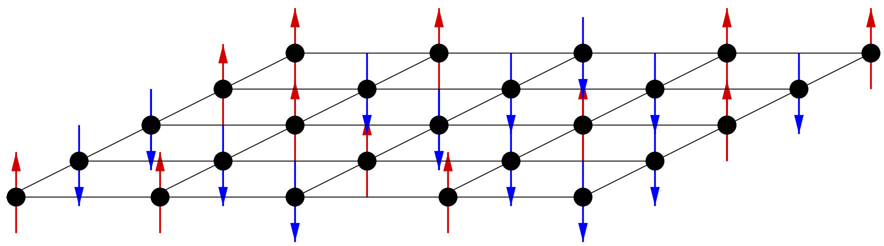

Figure 1: Sketch of 2D square-lattice Ising model.

where J is a magnetic coupling constant and B is a constant describing the external magnetic field. To account for boundary effects, we set $s_{-1,j} = s_{j,-1} = s_{m,j} = s_{j,m} = 0$ in (1). The probability of obtaining a specific system state is then given by the *Boltzmann* distribution with Probability Mass Function (PMF)

$$f(\boldsymbol{S}) \equiv f_\beta(\boldsymbol{S}) = \frac{1}{Z_\beta} e^{-H(\boldsymbol{S})\beta} \,, \tag{2}$$

where $\beta = 1/(k_B T)$ denotes the so-called inverse-temperature (or thermodynamic beta) with k_B being the Boltzmann constant and T the absolute temperature. Here, Z_β denotes the normalization constant that makes the target distribution $f_\beta \colon \{-1, +1\}^{m \times m} \to \mathbb{R}_+$ a proper PMF.

Let's denote by $M(\boldsymbol{S}) = \sum_{i,j=1}^{m} s_{ij}/m^2$ the system's average magnetic moment corresponding to the configuration $\boldsymbol{S}$. Notice that the random realizations of the configuration matrix $\boldsymbol{S}$ depend on the inverse temperature β. The expected value of the average magnetic moment $\overline{M}(\beta)$ as a function of the inverse temperature β thus reads

$$\overline{M}(\beta) = \sum_{\boldsymbol{S} \in \mathcal{K}} M(\boldsymbol{S}) f_\beta(\boldsymbol{S}) = \frac{1}{Z_\beta} \sum_{\boldsymbol{S} \in \mathcal{K}} M(\boldsymbol{S}) e^{-H(\boldsymbol{S})\beta} \,, \tag{3}$$

where $\mathcal{K} = \{-1, 1\}^{m \times m}$ is the set of all possible system configurations. Since the explicit computation of the normalization constant Z_β is computationally expensive (Explain why!), we rely on the Metropolis–Hastings algorithm here. That is, at each step a candidate configuration is proposed by randomly choosing an atom, with uniform probability, and "flipping" its spin.

1. Write a `Python` function that implements the Metropolis–Hastings algorithm for the Ising model. The input parameters for your function are: the number of steps n of the chain that should be simulated, the number of atoms m^2, the inverse temperature β, the constants J and B, and the initial state of the system. The function should return a list of energies and mean magnetic moments computed for each step of the chain, as well as the final configuration of the system.

2. Use your `Python` function with $\beta = 1/3$ and for n, such that both the energy and the average magnetic moment appear to have reached stationarity. Plot also the final system configuration. Furthermore, compute the mean magnetic moment $\overline{M}(\beta)$ for different values of $\beta \in [\frac{1}{3}, 1]$ and $n = 5 \cdot 10^6$. Choose a lattice of 50×50 atoms, $J = 1$, and $B > 0$ for all simulations.

Exercise 2

Recall that the standard Metropolis-Hastings algorithm accepts a new candidate state j drawn from the transition matrix Q, given the current state i, with probability $\alpha(i,j) = \min\left(\frac{\pi_j Q_{ji}}{\pi_i Q_{ij}}, 1\right)$, where π is the target probability measure. Consider now a Metropolis-Hastings algorithm that uses the follwing alternative acceptance probabilities

$$\alpha_1(i,j) = \frac{\pi_j Q_{ji}}{\pi_j Q_{ji} + \pi_i Q_{ij}},$$

and

$$\alpha_2(i,j) = \frac{\delta_{ij}}{\pi_i Q_{ij}},$$

with δ such that $\delta_{ij} \leq \pi_i Q_{ij} \forall i, j$. Show that, in both cases, the produced Markov chain satisfies the detailed balance condition.

Exercise 3

Consider the following AR(k) model defined by

$$\boldsymbol{y}_n = A\boldsymbol{y}_{n-1} + \boldsymbol{\xi}_n, \quad \boldsymbol{\xi}_n \overset{\text{iid}}{\sim} \mathcal{N}(0, \Gamma), \quad \boldsymbol{\xi}_n \in \mathbb{R}^k,$$

with $A \in \mathbb{R}^{k\times k}$, invertible and $\Gamma \in \mathbb{R}^{k\times k}$ full rank.

(a) Show that the previous process is a Markov chain.

(b) Show that if $\boldsymbol{y}_0 \sim \mathcal{N}(0, \Gamma_0)$, then $\boldsymbol{y}_n$ follows a multivariate Gaussian distribution for all n.

(c) Find the invariant distribution of an AR(1) process (i.e, a special case of the previous model).

(d) Simulate the AR(1) process and assess its convergence to the invariant distribution. In addition, verify the ergodic theorem on the quantity

$$\hat{\mu}^N = \frac{1}{N}\sum_{n=1}^{N} y_n.$$

(e) Establish theoretically the convergence of $\hat{\mu}^N$ by using the strong law of large numbers, and a weighted version of the central limit theorem (e.g. Lindberg-Feller)