Chapter 4
Markov Chain Monte Carlo

Qing Zhou*

Contents

*UCLA Department of Statistics (email: zhou@stat.ucla.edu).

1. The Basic Idea

We want to simulate a d-dimensional random vector $X \sim \pi$ (joint distribution) and compute

$$\mu = \mathbb{E}_\pi(h(X)) = \int_{\mathbb{R}^d} h(x)\pi(x)dx.$$

1.1. Markov chain Monte Carlo

Generate a Markov chain $x_1, x_2, \cdots, x_n$ by simulating $x_t \sim p(\cdot|x_{t-1})$, where $x_t = (x_{t1}, \cdots, x_{td})$, such that as $n \to \infty$,

1. $\hat{\mu} = \frac{1}{n}\sum_{t=1}^{n} h(x_t) \approx \mu$,
2. $x_n \sim \pi$.

Note that $x_1, x_2, \cdots, x_n$ are correlated.

1.2. Transition kernel and stationary distribution

Denote the one-step transition kernel of a Markov chain (M.C.) on a general state space ($\mathbb{R}^d$) by $K(x, y) := p_{X_t|X_{t-1}}(y|x)$. This generalizes the one-step transition probabilities $p_{ij} = P(X_t = j \mid X_{t-1} = i)$ for discrete state Markov chains. If a probability density π satisfies

$$\int \pi(x)K(x, y)dx = \pi(y) \quad \text{for all } y, \tag{1}$$

then $\pi(x)$ is a stationary distribution of the Markov chain:

$$X_t \sim \pi \Longrightarrow X_{t+1} \sim \pi.$$

The definition in (1) is a natural generalization of the definition for discrete case:

$$\sum_i \pi_i \cdot p_{ij} = \pi_j \qquad \text{for all } j.$$

1.3. Simulating a Markov chain

Given initial state x_0, transition kernel $K(x, y)$, it is straightforward to simulate an M.C. with the transition kernel for $t = 1, 2, \cdots, n$ by the following algrithm.

For $t = 1, 2, \cdots, n$,

Draw $x_t \sim K(x_{t-1}, \bullet)$.

This is to draw from the conditional distribution $[x_t \mid x_{t-1}]$. Recall for discrete case, we draw x_t from a discrete distribution with probabilities $\mathbb{P}[x_{t-1}, \bullet]$, one row in the transition matrix $\mathbb{P}$.

2. The Metropolis-Hastings Algorithm

Given a target distribution with density $\pi(x)$, the Metropolis-Hastings (MH) algorithm simulates a Markov chain with π as its stationary distribution. Let $\mathcal{S}$ denote the support of $\pi(x)$, i.e.,

$$\mathcal{S} = \{x : \pi(x) > 0\}, \tag{2}$$

which defines the state space for the Markov chain simulated by the MH algorithm.

2.1. Algorithm

Algorithm 1 (The MH algorithm)**.** Pick a random initial state $x^{(0)} \in \mathcal{S}$. Design a proposal distribution $q(x, y)$, which draws a random variable y given the value of x, i.e. it defines a conditional distribution $[y \mid x]$. The proposal must satisfy $q(x, y) = 0$ for any $y \notin \mathcal{S}$, i.e. the proposal only generates y such that $\pi(y) > 0$.

For $t = 1, 2, \cdots, n$,

1. Draw y from the proposed distribution $q(x^{(t-1)}, y)$;
2. Compute the MH ratio $r(x^{(t-1)}, y) = \min\left[1, \dfrac{\pi(y)q(y, x^{(t-1)})}{\pi(x^{(t-1)})q(x^{(t-1)}, y)}\right]$;
3. Draw $u \sim \text{Unif}(0, 1)$ and update

$$x^{(t)} = \begin{cases} y, & \text{if } u \le r(x^{(t-1)}, y); \\ x^{(t-1)}, & \text{otherwise.} \end{cases}$$

First development: Metropolis et al. (1953) with $q(x, y) = q(y, x)$ (symmetric proposal), in which case the MH ratio simplifies:

$$r(x, y) = \min\left[1, \frac{\pi(y)}{\pi(x)}\right] = \begin{cases} 1, & \text{if } \pi(y) \ge \pi(x); \\ \dfrac{\pi(y)}{\pi(x)}, & \text{if } \pi(y) < \pi(x). \end{cases}$$

As an example, consider a simple proposal $q(x, y)$ that draws

$$y \mid x \sim \text{Unif}(x - \delta,\ x + \delta).$$

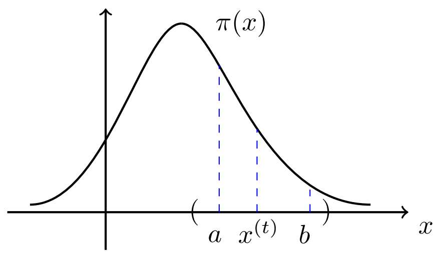
Therefore, the proposal (conditional density)

$$q(x, y) = p(y \mid x) = \frac{1}{2\delta}, \qquad \text{if } |y - x| < \delta.$$

As a bivariate function, $q(x, y)$ is symmetric in x, y, i.e. $q(y, x) = q(x, y)$:

$$q(y, x) = \frac{1}{2\delta}, \qquad \text{if } |x - y| < \delta.$$

Therefore, this is a symmetric proposal. Using this proposal, the main steps of the MH algorithm are illustrated with the following figure. In the figure, both $a, b \in (x^{(t)} - \delta,\ x^{(t)} + \delta)$ and $\pi(a) > \pi(x^{(t)}) > \pi(b)$.



$$y \mid x^{(t)} \sim \text{Unif}(x^{(t)} - \delta,\ x^{(t)} + \delta);$$
$$r(x^{(t)}, y) = \min\left[1, \frac{\pi(y)}{\pi(x^{(t)})}\right].$$

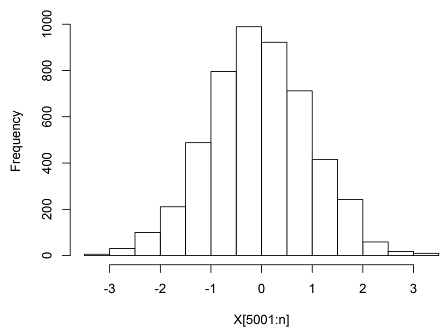
- If $y = a$, then $r(x^{(t)}, y) = 1$ and $x^{(t+1)} = y$.
- If $y = b$, then $r(x^{(t)}, y) = \pi(b)/\pi(x^{(t)}) < 1$: $x^{(t+1)} = y$ with probability $\pi(b)/\pi(x^{(t)})$ and $x^{(t+1)} = x^{(t)}$ with probability $1 - \pi(b)/\pi(x^{(t)})$.

2.2. Examples

Example 1. Draw $\mathcal{N}(0, 1)$ by an MH algorithm using $\text{Unif}(x - \delta, x + \delta)$ with $\delta = 1$ as the proposal.

```
# R code for this example

n=10000;
d=1;
X=numeric(n);
X[1]=0;
a=0;

for(t in 2:n)
{
   Y=runif(1,X[t-1]-d,X[t-1]+d);
   r=min(1,exp(-0.5*Y^2)/exp(-0.5*X[t-1]^2));
   u=runif(1,0,1);
   if(u<r){X[t]=Y;a=a+1}else{X[t]=X[t-1]};
}
```

```
a/n # acceptance rate
[1] 0.805

#use the last 5000 iterations (X[5001:n]) as our samples from N(0,1)
mean(X[5001:n])
[1] -0.04334007
sd(X[5001:n])
[1] 0.9988046

hist(X[5001:n])
```

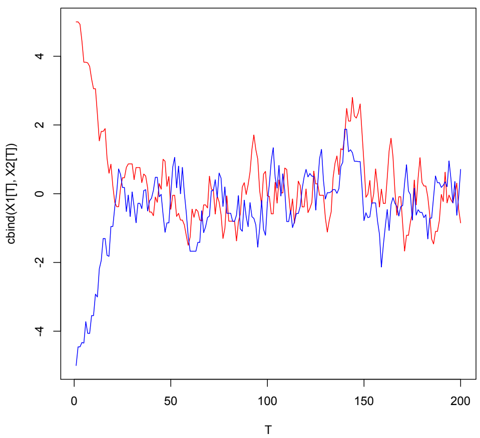


Some remarks:

- If an MH algorithm is irreducible and aperiodic, then the M.C. $\{x^{(t)}\}$ converges to the stationary distribution $\pi(x)$ and sampler averages approximate expectations:

$$\frac{1}{n}\sum_t h(x^{(t)}) \xrightarrow{a.s.} \mathbb{E}_\pi h(x). \tag{3}$$

- Burn-in period. Run this example with different initial values $x^{(0)} = 5$ (red) vs $x^{(0)} = -5$ (blue). The plot shows that the M.C. converges (two curves mix) after about 30 iterations (burn-in period). We usually use the average over $x^{(t)}$ after the burn-in period for estimation in (3).



Next, we demonstrate how to design a proposal $q(x, y)$ such that y always stays in $\mathcal{S}$.

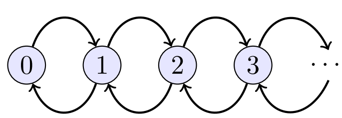
Example 2. Poisson Distribution.

$$\pi(x) = \frac{e^{-\lambda}\lambda^x}{x!} \propto \frac{\lambda^x}{x!}, \qquad x = 0, 1, 2, \cdots .$$

For the example, the state space $\mathcal{S} = \{0, 1, \cdots\}$ (nonnegative integers). Therefore, the proposal $q(x, y)$ should only move in $\mathcal{S}$. One possible design is

$$\begin{array}{ll} \text{If } x \geq 1, \quad \text{then } y = \left\{ \begin{array}{ll} x+1, & \text{with probability } 1/2; \\ x-1, & \text{with probability } 1/2; \end{array} \right. \\ \text{If } x = 0, \quad \text{then } y = 1 \;\; \text{with probability } 1. \end{array}$$

The state transition diagram of $q(x, y)$:



$q(0, 1) = 1$ and $q(x, y) = 1/2$ if $x \geq 1$ and $y \in \{x - 1, x + 1\}$.

The ratio between target densities: $\dfrac{\pi(y)}{\pi(x)} = \dfrac{\lambda^y}{\lambda^x}\dfrac{x!}{y!}$. ($\pi(x)$ can be unnormalized.)

If $x, y \geq 1$, $\dfrac{q(y, x)}{q(x, y)} = 1$: Symmetric.

If $x = 0, y = 1$, $\dfrac{q(y,x)}{q(x,y)} = \dfrac{q(1,0)}{q(0,1)} = \dfrac{\frac{1}{2}}{1} = \dfrac{1}{2}$.

If $x = 1, y = 0$, $\dfrac{q(y,x)}{q(x,y)} = \dfrac{q(0,1)}{q(1,0)} = \dfrac{1}{\frac{1}{2}} = 2$.

2.3. Detail balance

In this section, we verify that π is indeed a stationary distribution of the Markov chain simulated by the MH algorithm. That is, for any $y \in \mathcal{S}$,

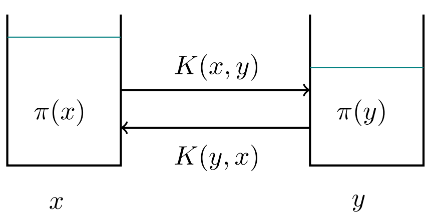
$$\int_{\mathcal{S}} \pi(x)K(x,y)dx = \pi(y), \qquad \text{or} \sum_{x \in \mathcal{S}} \pi(x)K(x,y) = \pi(y)$$

for discrete state space, where $K(x,y)$ is the one-step transition kernel of the MH algorithm.

We consider a sufficient condition that is easy to check, called the detail balance condition:

$$\pi(x)K(x,y) = \pi(y)K(y,x), \qquad \text{for all } x, y \in \mathcal{S}. \tag{4}$$

The key intuition behind the detail balance condition may be understood using water flow between two tanks x and y as an analogy, illustrated in the following figure. The volumes of water in the two tanks are $\pi(x)$ and $\pi(y)$, respectively. Two pipes connect the tanks, one allowing water to flow from x to y and the other from y to x. The flow rates are $K(x,y)$ and $K(y,x)$ per unit volume of water. Thus, the flow rate from tank x to y is $\pi(x)K(x,y)$, and $\pi(y)K(y,x)$ in the other direction. If the detail balance condition holds, then the amount of water flow from x to y will match exactly that from y to x, and as a result, the volumes $\pi(x)$ and $\pi(y)$ will stay constant over time.



Lemma 1. *If the detail balance condition (4) holds, then π is a stationary distribution of the Markov chain with $K(x,y)$ as the one-step transition kernel.*

Proof. Integrating over x on both sides of the detail balance condition, we have, for any y,

$$\int \pi(x)K(x,y)dx = \int \pi(y)K(y,x)dx = \pi(y) \int K(y,x)dx = \pi(y),$$

where the last equality is due to the fact that $K(y, x)$ is a conditional density for $[x \mid y]$. $\square$

Theorem 1. *The MH algorithm simulates a Markov chain for which $\pi(x)$ is a stationary distribution.*

Proof. It suffices to show that the detail balance condition (4) is satisfied, i.e. $\pi(x)K(x, y) = \pi(y)K(y, x)$ for any x and y.

1. It is trivially true for $x = y$;
2. Suppose $y \neq x$. Then the MH algorithm must propose y and accept it. Therefore, the transition kernel

$$K(x, y) = q(x, y) \min\left[1, \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)}\right].$$

Now we have

$$\begin{aligned}
\pi(x)K(x, y) &= \pi(x)q(x, y) \min\left[1, \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)}\right] \\
&= \min[\pi(x)q(x, y), \pi(y)q(y, x)] \\
&= \min\left[\frac{\pi(x)q(x, y)}{\pi(y)q(y, x)}, 1\right] \cdot \pi(y)q(y, x) \\
&= \pi(y)K(y, x).
\end{aligned}$$

$\square$

2.4. Autocorrelation and efficiency

Consider the efficiency of MCMC for estimating

$$\mu_h = \int h(x)\pi(x)dx = \mathbb{E}_\pi[h(X)].$$

Suppose $x^{(1)}, x^{(2)}, \cdots, x^{(m)}$ is a Markov chain with π as its stationary and also limiting distribution. Let $\overline{h}_m = \frac{1}{m}\sum_{i=1}^{m} h(x^{(i)})$.

Assume that $x^{(0)} \sim \pi(x)$. If m is large, then

$$\mathrm{Var}(\overline{h}_m) = \frac{\sigma^2}{m}\left[1 + 2\sum_{j=1}^{m-1}\left(1 - \frac{j}{m}\right)\rho_j\right] \approx \frac{\sigma^2}{m}\left[1 + 2\sum_{j=1}^{\infty}\rho_j\right], \tag{5}$$

where $\sigma^2 = \text{Var}_\pi[h(x)]$ and

$$\rho_j = \text{cor}(h(x^{(1)}), h(x^{(1+j)})) = \text{cor}(h(x^{(t)}), h(x^{(t+j)})), \qquad \text{for any } t = 1, 2, \ldots$$

is the j-step autocorrelation.

Comparing (5) to an independent sample, $x^{(i)} \sim \pi$ independently for $i = 1, \ldots, m$,

$$\text{Var}(\overline{h}_m) = \frac{\sigma^2}{m},$$

we define effective sample size of this Markov chain as

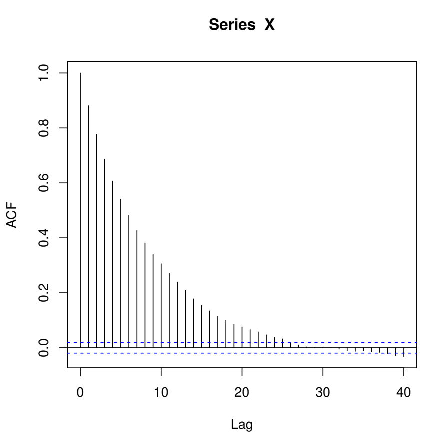
$$\frac{m}{1 + 2\sum_{j=1}^{\infty} \rho_j}.$$

Thus, the faster the autocorrelation ρ_j decays to zero, the more efficient the estimation of μ_h by the MCMC algorithm.

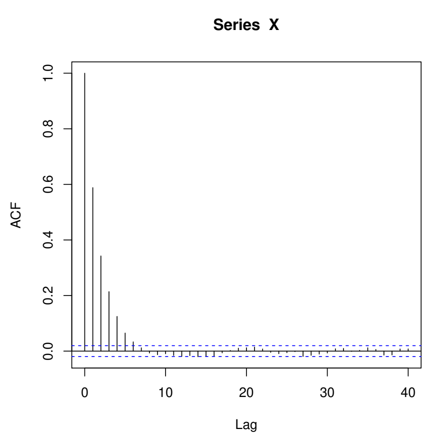
In Example 1, we may change the value of δ in the proposal $\text{Unif}(x - \delta, x + \delta)$ to see the change in autocorrelations, demonstrating different efficiency for different proposals. The figures below show the autocorrelation plot, ρ_j for $j = 0, \ldots, 40$, generated by

```
acf(X)
```

for $\delta = 1$ and $\delta = 5$ and the corresponding acceptance rates P_a. The autocorrelation plots suggest that the choice of $\delta = 5$ gives more efficient estimates. See Section 5.1 for related discussion.



$\delta = 1,\ P_a = 0.805$



$\delta = 5,\ P_a = 0.312$

3. Ising Model

3.1. MH Algorithm for 1-D Ising Model

We use the 1-D Ising model to demonstrate the MH algorithm for simulating from a joint distribution.

Example 3 (1-D Ising Model)**.** Consider a random vector $x = (x_1, \cdots, x_d) \in \{1, -1\}^d$, i.e. every $x_j \in \{1, -1\}$. Define an energy function

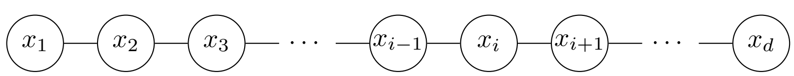
$$U(x) = -\sum_{i=1}^{d-1} x_i x_{i+1}.$$

At a given temperature $T > 0$, the Boltzmann distribution is specified by the probability mass function

$$\pi(x) \propto \exp\left[-\frac{U(x)}{T}\right] = \exp\left[\mu \sum_{i=1}^{d-1} x_i x_{i+1}\right], \tag{6}$$

where $\mu = 1/T > 0$. Note that $\pi(x) = \pi(x_1, \dots, x_d)$ is a joint distribution over d binary random variables $x_i \in \{\pm 1\}, i = 1, \dots, d$. There are a total of 2^d possible combinations among the x_i's. We call each combination a configuration. This is a simple model for a physical system consisting of d particles. The Boltzmann distribution assign a probability $\pi(x)$ for each configuration x.

We can use a graph to represent the joint distribution $\pi(x)$. Each node in the graph corresponds to a random variable and an edge exists if there is a product term $(x_i x_{i+1})$ in $U(x)$:



Given a current configuration $x^{(t)} = (x_1^{(t)}, \cdots, x_d^{(t)})$, one iteration of the MH algorithm consists of:

1. Proposal: Randomly choose j from $\{1, \dots, d\}$ and flip x_j to its opposite:
$$y = (x_1^{(t)}, \cdots, -x_j^{(t)}, \cdots, x_d^{(t)}).$$
This is a symmetric proposal: $q(x^{(t)}, y) = q(y, x^{(t)})$.
2. Thus, the MH ratio
$$r(x^{(t)}, y) = \min\left[1, \frac{\pi(y)}{\pi(x^{(t)})}\right],$$
$$\frac{\pi(y)}{\pi(x^{(t)})} = \exp\left\{-2\mu x_j^{(t)} \left(x_{j-1}^{(t)} + x_{j+1}^{(t)}\right)\right\},$$
where $x_0^{(t)} = x_{d+1}^{(t)} \equiv 0$.

The following R code implements this MH algorithm to simulate from π with $T = 1$ and estimate $\mathbb{E}_\pi g(x) = \mathbb{E}_\pi(\sum_i x_i)$. You may change the value of T (temperature) to see its effect on the distribution and the expectation.

```
n=6000;
d=20;
X=matrix(0,n,d);
X[1,]=sample(c(-1,1),size=d,replace=TRUE);
g=numeric(n);
g[1]=sum(X[1,]);

T=1;

for(t in 2:n)
{
   y=X[t-1,];
   j=sample(1:d,size=1);
   y[j]=-X[t-1,j];
   if(j==1){
      r=exp(-2*X[t-1,1]*X[t-1,2]/T);
   }else if(j==d){
      r=exp(-2*X[t-1,d-1]*X[t-1,d]/T);
   }else{
      r=exp(-2*X[t-1,j]*(X[t-1,j-1]+X[t-1,j+1])/T);
   }
   U=runif(1,0,1);
   if(U<=min(r,1)){X[t,]=y}else{X[t,]=X[t-1,]};

   g[t]=sum(X[t,]);
}
mean(g[1000:n])
```

3.2. Boltzmann Distribution

The Boltzmann distribution

$$P_T(x) = \frac{1}{Z(T)} e^{-h(x)/T}, \tag{7}$$

where $x \in [N] := \{1, \ldots, N\}$ is the configuration (state) of a physical system, $h(x)$ is the energy of state x; $T > 0$ is the temperature, and

$$Z(T) = \sum_{x=1}^{N} e^{-h(x)/T}$$

is normalization constant (partition function). Important physical quantities, such as energy and entropy, are defined via P_T:

Energy $U_T = \mathbb{E}(h(X)) = \sum_x h(x)P_T(x)$.

Entropy $S_T = -\mathbb{E}[\log P_T(X)] = -\sum_x P_T(x)\log P_T(x)$.

However, since the number of states N is typically very large, combinatorial in the number of particles in a system, the above expectations cannot be calculated exactly. For example, if the state $x = (x_1, \dots, x_M)$, each $x_i \in \{\pm 1\}$ representing the state of a particle, then $N = 2^M$. Thus, we usually use Monte Carlo simulation to approximate them: Given $h(x)$ and $T > 0$, draw $x^{(i)} \sim_{iid} P_T$ for $i \in [n]$ to estimate

$$\widehat{U}_T = \frac{1}{n}\sum_{i=1}^n h(x^{(i)}), \qquad \widehat{P}_T(x) = \frac{1}{n}\sum_{i=1}^n I(x^{(i)} = x),$$

and $\widehat{S}_T = -\sum_x \widehat{P}_T(x)\log \widehat{P}_T(x)$.

Derivation of the Boltzmann distribution P_T is based on two physical laws: (i) maximum entropy and (ii) conservation of average energy. Put

$$p = (p_1, \dots, p_N) = (p_x)_{1\le x\le N}$$

and suppose the average energy is u (fixed). Then P_T is the solution to

$$\max_p \left\{-\sum_x p_x \log p_x\right\}$$
$$\text{subject to } \sum_x p_x = 1, \qquad \sum_x p_x h(x) = u.$$

Define the Lagrangian

$$L(p,\beta,\lambda) = -\sum_x p_x \log p_x - \beta\left(\sum_x p_x h(x) - U\right) - \lambda\left(\sum_x p_x - 1\right)$$

and set its derivatives to zero

$$\frac{\partial L}{\partial p_x} = -\{\log p_x + 1 + \beta h(x) + \lambda\} = 0$$

to get

$$p_x = \frac{\exp(-\beta h(x))}{C(\beta)}, \qquad C(\beta) = \sum_x \exp(-\beta h(x)).$$

Moreover, β is determined by the average energy u, since

$$\sum_x \frac{\exp(-\beta h(x))}{C(\beta)} h(x) = u.$$

Now letting $T = 1/\beta$ and $Z(T) = C(1/T)$, we arrive at the Boltzmann distribution P_T in (7).

4. Simulated Annealing

For any $\pi(x)$, let $h(x) = -\log(\pi(x))$. For $T > 0$, define

$$\pi(x;T) \propto \exp\left[-\frac{h(x)}{T}\right],$$

where T is the temperature in the Boltzmann distribution (7) regarding $h(x)$ as the energy function. In particular, $\pi(x) = \pi(x; T = 1)$. Denote the global minimizer of h by

$$x^* = \operatorname{argmin} h(x) = \operatorname{argmax} \pi(x).$$

Varying the temperature $T \in (0, \infty)$, we can change the shape of the distribution $\pi(x;T)$:

- $T \to \infty$: for any x,

$$\frac{\pi(x;T)}{\pi(x^*;T)} = \exp\left[\frac{h(x^*) - h(x)}{T}\right] \to 1.$$

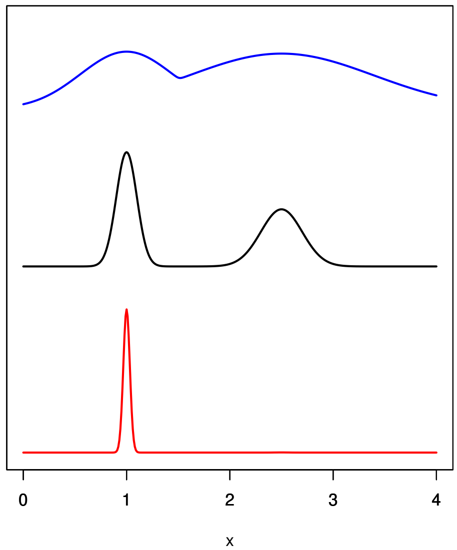
  Thus, $\pi(x;T) \propto 1$, close to uniform distribution.
- $T \to 0$: for any $h(x) > h(x^*)$,

$$\frac{\pi(x;T)}{\pi(x^*;T)} = \exp\left[\frac{h(x^*) - h(x)}{T}\right] \to \exp(-\infty) = 0.$$

  Thus, $\pi(x;T)$ is concentrated at x^*, i.e. a point mass at x^*.

The goal of simulated annealing is to find x^*, the global minimizer of $h(x)$. This method uses the MH algorithm to simulate from $\pi(x;T)$ with a non-increasing sequence of T. It starts with a high temperature (large T) and gradually decreases T to zero. At a high temperature, since $\pi(x;T)$ is pretty flat, the MH algorithm has a decent chance to explore different local modes of the density. Later on, as T decreases to 0, the samples will converge to x^* with a high probability.

The following figure illustrates the idea of simulated annealing, showing $\pi(x;T)$ for $T = 20, 1, 0.1$. The target density $\pi(x)$ (black curve, $T = 1$) has two modes, the global maximizer $x^* = 1$ and another local maximizer at $x = 2.5$.



blue: $T = 20$
black: $T = 1$
red: $T = 0.1$

Algorithm 2 (Simulated annealing)**.** Choose $T_1 \geq T_2 \geq \cdots \geq T_n \to 0$ and pick $x^{(0)}$.

For $t = 1, \ldots, n$:

- Set $T = T_t$.
- Draw $x^{(t)}$ given $x^{(t-1)}$ via one step of an MH algorithm targeting at $\pi(x; T)$. That is, in step 2 of Algorithm 1, we replace $\pi(x)$ and $\pi(y)$ by $\pi(x; T)$ and $\pi(y; T)$, respectively.

5. Some Special Designs

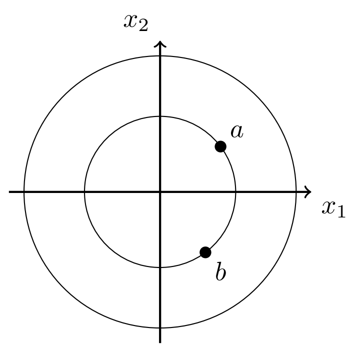
5.1. Random-walk Metropolis

Consider $\pi(x)$ defined on $\mathbb{R}^d$ (d-dimensional Euclidean Space). Use the addition of a random perturbation (an error vector) as the proposal in the MH algorithm.

Given the current sample $x^{(t)}$, the proposal $q(x^{(t)}, y)$ draws

$$y = x^{(t)} + \varepsilon_t, \qquad \varepsilon_t \sim g_\sigma(\varepsilon), \tag{8}$$

where g_σ is a spherically symmetric distribution, i.e., $g_\sigma(a) = g_\sigma(b)$ if $\|a\| = \|b\|$ (Euclidean norm).



Examples of $g_\sigma(\varepsilon)$ include multi-variate Gaussian $\mathcal{N}_d(0, \sigma^2 \mathbf{I}_d)$ and Unif$(B(0,\sigma))$, where $B(0,\sigma)$ is the ball centering at 0 with radius σ, i.e.

$$B(0,\sigma) := \{x \in \mathbb{R}^d : \|x\| \le \sigma\}.$$

The proposal in (8) is symmetric, $q(x^{(t)}, y) = q(y, x^{(t)})$, since $g_\sigma(\varepsilon) = g_\sigma(-\varepsilon)$.

The random-walk Metropolis:

Given $x^{(t)}$,

1. Draw $\varepsilon_t \sim g_\sigma(\varepsilon)$: spherically symmetric (σ can be controlled by the user), set $y = x^{(t)} + \varepsilon_t$, $r(x^{(t)}, y) = \min\left[1, \dfrac{\pi(y)}{\pi(x^{(t)})}\right]$;
2. Draw $u \sim$ Unif(0, 1) and update

$$x^{(t+1)} = \begin{cases} y, & \text{if } u \le r(x^{(t)}, y); \\ x^{(t)}, & \text{otherwise.} \end{cases}$$

How to choose σ: maintain acceptance rate $\in [0.25, 0.35]$. See the autocorrelation plots in Section 2.4.

5.2. Metropolized independence sampler

In some problems, we may have ways to approximate the target distribution π by a trial distribution g that we can simulate from. In these cases, we may choose $q(x, y) = g(y)$, which defines a proposal that is independent of x. An MH algorithm with such an independent proposal is called a Metropolized independence sampler:

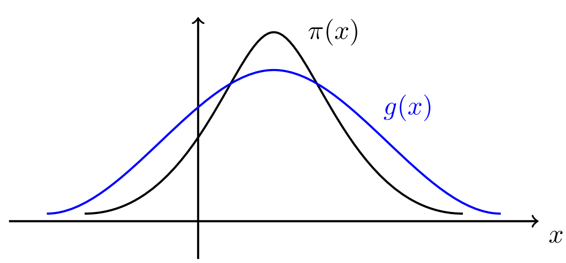
Given $x^{(t)}$,

1. Draw $y \sim g(y)$,
$$r(x^{(t)}, y) = \min\left[1, \frac{\pi(y)}{\pi(x^{(t)})}\frac{g(x^{(t)})}{g(y)}\right] = \min\left[1, \frac{w(y)}{w(x^{(t)})}\right],$$
where $w(x) = \pi(x)/g(x)$ is the importance weight;
2. Draw $u \sim$ Unif $(0, 1)$,
$$x^{(t+1)} = \begin{cases} y, & \text{if } u \le r(x^{(t)}, y); \\ x^{(t)}, & \text{otherwise.} \end{cases}$$

Some remarks:

(a) This method is closely related to importance sampling and it uses importance weights $w(y)/w(x^{(t)})$ to calculate the MH ratio. Similar to importance sampling, the efficiency of this MH algorithm depends on how close $g(y)$ is to $\pi(y)$. One way to measure the closeness is by the variance of the importance weights: $\mathrm{Var}_g[w(x)] := V_w$. Small V_w suggests that g is close to π and usually leads to a higher acceptance rate. If $\mathrm{Var}_g(w(x)) = 0$, then $g = \pi$ and $r(x, y) = 1$ for all x, y. Therefore, for a Metropolized independence sampler, the higher the acceptance rate, the more efficient of the algorithm.

(b) To get robust performance and reduce the variance V_w, the trial distribution g should have a heavier tail than π. For example, if π is a normal distribution then g could be a t-distribution.



Example 4 (Gamma distribution)**.** Design a Metropolized independent sampler to draw from Gamma(α, β), $\alpha > 1, \beta > 0$,

$$\pi(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}, \quad x > 0,$$

using Exp(λ) as the trial distribution. Let

$$w(x) = \frac{x^{\alpha-1} e^{-\beta x}}{\lambda e^{-\lambda x}}.$$

Choose λ to minimize $\text{Var}_g(w(x))$.

Since $\mathbb{E}_g(w(x)) = \int x^{\alpha-1} e^{-\beta x} dx = \Gamma(\alpha)/\beta^\alpha$ is a constant independent of λ, it is equivalent to minimizing $\mathbb{E}_g[w(x)^2] = \int w(x)^2 g(x) dx$. Some calculation shows that

$$\mathbb{E}_g[w(x)^2] = \frac{1}{\lambda} \int_0^\infty x^{2\alpha-2} e^{-(2\beta-\lambda)x} dx.$$

Note that $\mathbb{E}_g[w(x)^2] < \infty$ if and only if $2\beta - \lambda > 0$. So we must choose

$$\lambda < 2\beta. \tag{9}$$

Under this condition, the integrand is an unnormalized Gamma$(2\alpha - 1, 2\beta - \lambda)$ and thus

$$\mathbb{E}_g[w(x)^2] = \frac{1}{\lambda} \cdot \frac{\Gamma(2\alpha - 1)}{(2\beta - \lambda)^{2\alpha-1}}.$$

Therefore, to minimize $\mathbb{E}_g[w(x)^2]$ we just need to maximize

$$f(\lambda) = \lambda(2\beta - \lambda)^{2\alpha-1}$$

over λ. Since the objective $f(\lambda) > 0$, we can equivalently

$$\max_\lambda \Big[\log f(\lambda) = \log \lambda + (2\alpha - 1) \log(2\beta - \lambda) \Big]$$

of which the only maximizer is

$$\lambda^* = \beta/\alpha$$

by setting derivative to zero. Since $\alpha > 1$, we have $\lambda^* < \beta$ satisfying the constraint (9). This also shows that the tail of g is heavier than that of π (Remark b):

$$\lim_{x\to\infty} \frac{\pi(x)}{g(x)} = C \lim_{x\to\infty} \frac{x^{\alpha-1}}{e^{(\beta-\lambda^*)x}} = 0,$$

where $C > 0$ is a constant.

In fact, with $\lambda^* = \beta/\alpha$, g and π have the same mean $(1/\lambda^* = \alpha/\beta)$. That is, we have matched the expectations of the two distributions with this optimal choice.

5.3. Single-coordinate updating

This design is for multivariate distributions. For

$$\mathbf{x} = (x_1, \cdots, x_{i-1}, x_i, x_{i+1} \cdots, x_d) \in \mathbb{R}^d,$$

define

$$\begin{aligned} \mathbf{x}_i(y) &:= (x_1, \cdots, x_{i-1}, y, x_{i+1}, \cdots, x_d): \ \mathbf{x} \text{ with } y \text{ replacing } x_i; \\ \mathbf{x}_{[-i]} &:= (x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_d): \ \mathbf{x} \text{ with } x_i \text{ omitted.} \end{aligned}$$

Our target distribution is $\pi(\mathbf{x})$.

To do single-coordinate update in the MH algorithm, the proposal $q(\mathbf{x}, \mathbf{y})$ has two steps:

(a) Select a coordinate i, either cycling through 1 to d deterministically, or randomly from $\{1, \dots, d\}$.
(b) Given i, draw $y \sim q_i(x_i, y)$, which proposes a scaler y from some univariate distribution $[y \mid x_i]$, e.g. $y \sim \mathcal{N}(x_i, 1)$. Then put $\mathbf{y} = \mathbf{x}_i(y)$. That is, the proposal only changes the ith coordinate of $\mathbf{x}$.

The MH ratio is determined by

$$\frac{\pi(\mathbf{y})}{\pi(\mathbf{x})} \frac{q(\mathbf{y}, \mathbf{x})}{q(\mathbf{x}, \mathbf{y})} = \frac{\pi(\mathbf{x}_i(y))}{\pi(\mathbf{x})} \frac{q_i(y, x_i)}{q_i(x_i, y)}. \tag{10}$$

Let $\pi(\cdot \mid \mathbf{x}_{[-i]})$ be the conditional density of $[x_i \mid \mathbf{x}_{[-i]}]$. Then we have

$$\begin{aligned} \pi(\mathbf{x}) &= \pi(x_i|\mathbf{x}_{[-i]}) \cdot \pi(\mathbf{x}_{[-i]}), \\ \pi(\mathbf{x}_i(y)) &= \pi(y|\mathbf{x}_{[-i]}) \cdot \pi(\mathbf{x}_{[-i]}), \end{aligned}$$

and consequently, the ratio in (10) simplifies to

$$\frac{\pi(\mathbf{y})}{\pi(\mathbf{x})} \frac{q(\mathbf{y}, \mathbf{x})}{q(\mathbf{x}, \mathbf{y})} = \frac{\pi(y|\mathbf{x}_{[-i]})}{\pi(x_i|\mathbf{x}_{[-i]})} \frac{q_i(y, x_i)}{q_i(x_i, y)}. \tag{11}$$

This is the same as an MH algorithm with the conditional distribution $\pi(\cdot \mid \mathbf{x}_{[-i]})$ as the target and $q_i(x_i, y)$ as the proposal.

An *important* special case is to choose $q_i(x_i, y) = \pi(y|\mathbf{x}_{[-i]})$, i.e., we propose y by sampling from the conditional distribution $y \sim \pi(\cdot|\mathbf{x}_{[-i]})$. Accordingly, $q_i(y, x_i) = \pi(x_i|\mathbf{x}_{[-i]})$. Then by (11) the MH ratio

$$r(\mathbf{x}, \mathbf{y}) = \min\left[1, \frac{\pi(y|\mathbf{x}_{[-i]})}{\pi(x_i|\mathbf{x}_{[-i]})} \cdot \frac{\pi(x_i|\mathbf{x}_{[-i]})}{\pi(y|\mathbf{x}_{[-i]})}\right] \equiv 1,$$

so $\mathbf{y} = \mathbf{x}_i(y)$ is always accepted. In other words, we just iteratively sample from the conditional distribution $\pi(\cdot|\mathbf{x}_{[-i]})$ for a chosen coordinate $i \in \{1, \dots, d\}$. This is the Gibbs sampler.