3

Markov Chain Monte Carlo

3.1 Monte Carlo as Numerical Integration

Although our driving applications of Monte Carlo integration will mostly revolve around Bayesian inference, we would like to point out that all Monte Carlo methods can (should?) be viewed as a numerical integration problem. Such problems usually start with either discrete ($\mathbf{x}$) or continuous ($\boldsymbol{\theta}$) vector of random variables. Despite the fact that distributions of these vectors are known only up to a proportionality constant, we are interested in taking expectations with respect to these distributions. Compare the following integration problems faced by physicists and Bayesian statisticians.

Statistical mechanics	*Bayesian statistics*
$\Pr(\mathbf{x}) = \frac{1}{Z} e^{-\mathscr{E}(\mathbf{x})}$	$\Pr(\boldsymbol{\theta} \mid \mathbf{y}) = \frac{1}{C} \Pr(\mathbf{y} \mid \boldsymbol{\theta}) \Pr(\boldsymbol{\theta})$
Objective: $\mathrm{E}[f(\mathbf{x})] = \sum_{\mathbf{x}} f(\mathbf{x}) \Pr(\mathbf{x})$	Objective: $\mathrm{E}[f(\boldsymbol{\theta}) \mid \mathbf{y}] = \int f(\boldsymbol{\theta}) \Pr(\boldsymbol{\theta} \mid \mathbf{y}) \, \mathrm{d}\boldsymbol{\theta}$

Note 3.1. Many applications involve both intractable summation and integration:

$$\mathrm{E}[f(\mathbf{x}, \boldsymbol{\theta})] = \sum_{\mathbf{x}} \int f(\mathbf{x}, \boldsymbol{\theta}) \Pr(\mathbf{x}, \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta}.$$

The above integration problems are difficult to solve even numerically, especially in high dimensions, e.g. when the length of $\mathbf{x}$ and/or $\boldsymbol{\theta}$ is on the order of $10^3 - 10^6$. All Monte Carlo techniques attempt to solve such high dimensional integration problems by stochastic simulation.

Stan Ulam
1909 – 1984

During the Manhattan project at the Los Alamos National Laboratory, Polish-American mathematician Stanislaw (Stan) Ulam invented a Monte Carlo method for computing probabilities of various outcomes of a complex stochastic experiment. The key insight came to Ulam while playing solitaire and imagining "simulating" multiple solitaire games. Ulam's legacy is not limited to the Monte Carlo method — he made significant contributions to many branches of mathematics, including ergodic theory, set theory, and combinatorics, to name a few.

3.1.1 Classical Monte Carlo

In general, Monte Carlo integration (Metropolis and Ulam, 1949) aims at approximating expectations of the form

$$\mathrm{E}[h(X)] = \int h(x) f(x) \mathrm{d}x. \tag{3.1}$$

If $X_1, \ldots, X_n \overset{\text{iid}}{\sim} f(x)$ and $\mathrm{E}[h(X_1)] < \infty$, then we know from SLLN that

$$\frac{1}{n} \sum_{i=1}^{n} h(X_i) \overset{\text{a.s.}}{\to} \mathrm{E}_f[h(X_1)].$$

Therefore, we can approximate the desired expectation with

$$\bar{h}_n = \frac{1}{n}\sum_{i=1}^{n} h(X_i) \approx \mathrm{E}_f[h(X_1)]$$

for some large, yet finite n. Conveniently, the variance of this Monte Carlo estimator can be approximated as

$$\mathrm{Var}(\bar{h}_n) = \frac{1}{n^2} \times n \times \mathrm{Var}[h(X_1)] = \frac{1}{n}\int \left\{h(x) - \mathrm{E}_f[h(x)]\right\}^2 f(x)\mathrm{d}x \approx \frac{1}{n^2}\sum_{i=1}^{n}\left[h(X_i) - \bar{h}_n\right]^2 = v_n$$

Moreover, the central limit theorem says that

$$\frac{\bar{h}_n - \mathrm{E}_f[h(X_1)]}{\sqrt{v_n}} \xrightarrow{\mathrm{D}} \mathcal{N}(0,1),$$

allowing us to estimate the Monte Carlo error, e.g. $\bar{h}_n \pm 1.96\sqrt{v_n}$.

3.2 Markov chain Monte Carlo

Before we dive into MCMC, let us ask ourselves why we are not happy with classical Monte Carlo and if there is any need to invent something more complicated? The main motivation for developing MCMC is the fact that classical Monte Carlo is very hard to implement in high dimensional spaces. MCMC also often experiences difficulties in high dimensions. However, for almost any high dimensional integration, it is fairly straightforward to formulate an MCMC algorithm, while the same is not true for classical Monte Carlo.

Recall that our objective in MCMC is the same as in classical Monte Carlo: to estimate expectations of the form

$$\mathrm{E}_{\boldsymbol{\pi}}[h(\mathbf{x})] = \sum_{\boldsymbol{x}\in\Omega} \pi_{\boldsymbol{x}} h(\mathbf{x}). \tag{3.2}$$

Notice that here we assume that our state space is fini te so the above expectation is a finite sum. However we assume that the size of Ω is so large that carrying out this summation s impractical even on fastest computers. We also assume that we do not know how to produce iid samples from $\boldsymbol{\pi}$. The general MCMC strategy then is to construct an ergodic Markov chain $\{X_n\}$ with stationary distribution $\boldsymbol{\pi}$. Then from the ergodic theorem and N realizations from the Markov chain, we get

$$\mathrm{E}_{\boldsymbol{\pi}}[h(\mathbf{x})] \approx \frac{1}{N}\sum_{i=1}^{N} h(X_i). \tag{3.3}$$

The question is how to construct such a Markov chain, $\{X_n\}$.

3.2.1 Metropolis-Hastings Algorithm

In designing MCMC algorithms, we start with a target distribution $\boldsymbol{\pi}$. Given some initial value $X_0 = x_0$, we construct a Markov chain according to the following set of rules.

Algorithm 2 Metropolis-Hastings Algorithm: approximate $\mathrm{E}_{\boldsymbol{\pi}}[h(\mathbf{x})]$

1: Start with some initial value $X_0 = x_0$.
2: **for** $n = 0$ to N **do**
3: Simulate a candidate value $Y \sim q(j \,|\, X_n = i)$. Suppose $Y = j$.
4: Compute the Metropolis-Hastings **acceptance probability**

$$a_{ij} = \min\left\{ \frac{\pi_j q(i \,|\, j)}{\pi_i q(j \,|\, i)}, 1 \right\} \tag{3.4}$$

5: Generate $U \sim \mathrm{Unif}[0,1]$.
6: Accept the candidate $Y = j$ if $U \leq a_{ij}$, otherwise set $X_{n+1} = X_n$. More specifically, set

$$X_{n+1} = \begin{cases} Y & \text{if } U \leq a_{ij} \\ X_n & \text{if } U > a_{ij} \end{cases} \tag{3.5}$$

7: **return** $\frac{1}{N}\sum_{i=1}^{N} h(X_i)$.

Nicholas Metropolis
1915-1999

W. Keith Hastings
1930-2016

Nicholas Metropolis was the first author on a paper, written shortly after World War 2 at Los Alamos, developing an algorithm to estimate integrals using Markov chains. The algorithm that now has his name was likely invented by his co-author Marshall Rosenbluth, but Metropolis was team leader and therefore could get to be first author. The algorithm was generalized to asymmetric proposal distributions, which greatly enhanced applicability of the method, by Keith Hastings at University of Toronto. He got the idea from a consulting appointment with a chemistry professor. Hastings later moved to the University of Victoria where he was liked for his jokes in mathematics lectures. He did not continue working on MCMC methods.

Proposition 3.1. *The Metropolis-Hastings algorithm generates a Markov chain with stationary distribution* $\boldsymbol{\pi}$.

Proof: Let $\mathbf{P} = \{p_{ij}\}$ be the transition matrix for X_n. Then for $i \neq j$,

$$p_{ij} = \Pr(X_{n+1} = j \,|\, X_n = i) = \Pr(X_1 = j \,|\, X_0 = i) = a_{ij} q(j \,|\, i).$$

Again, for $i \neq j$,

$$\pi_i p_{ij} = \pi a_{ij} q(j \,|\, i) = \begin{cases} \pi_i q(j \,|\, i) \dfrac{\pi_j q(i \,|\, j)}{\pi_i q(j \,|\, i)} & \text{if } \dfrac{\pi_j q(i \,|\, j)}{\pi_i q(j \,|\, i)} \leq 1 \\ \pi_i q(j \,|\, i) \cdot 1 & \text{otherwise} \end{cases} = \begin{cases} \pi_j q(i \,|\, j) & \text{if } \dfrac{\pi_j q(i \,|\, j)}{\pi_i q(j \,|\, i)} \leq 1 \\ \pi_i q(j \,|\, i) & \text{otherwise} \end{cases}$$

and

$$\pi_j p_{ji} = \pi_j a_{ji} q(i \,|\, j) = \begin{cases} \pi_j q(i \,|\, j) \cdot 1 & \text{if } \dfrac{\pi_j q(i \,|\, j)}{\pi_i q(j \,|\, i)} \leq 1 \\ \pi_j q(i \,|\, j) \dfrac{\pi_i q(j \,|\, i)}{\pi_j q(i \,|\, j)} & \text{otherwise} \end{cases} = \begin{cases} \pi_j q(i \,|\, j) & \text{if } \dfrac{\pi_j q(i \,|\, j)}{\pi_i q(j \,|\, i)} \leq 1 \\ \pi_i q(j \,|\, i) & \text{otherwise} \end{cases}$$

So we have shown $\pi_i p_{ij} = \pi_j p_{ji}$. We require $\pi_i > 0$ for all i and $q(i|j) > 0 \Leftrightarrow q(j|i) > 0$. Since we have detailed balance, we conclude that $\boldsymbol{\pi}$ is a stationary distribution. $\square$

Note 3.2. If we choose $\{q(i,j)\}$ so that $\{X_n\}$ is irreducible, then $\{X_n\}$ is positive recurrent by the stationary distribution criterion even on an infinite state-space. Therefore, we can use the ergodic theorem.

Note 3.3. We do not need the normalizing constant of $\boldsymbol{\pi}$ in order to execute the Metropolis-Hastings algorithm.

Example: *Ising model on a circle* The magnetization of a permanent magnet diminishes in strength as the magnet is heated. Above a certain critical temperature the magnet stops being a magnet. At the other extreme, for very low temperatures non-magnets may exhibit spontaneous magnetization. At the atomic level, each atom is by itself a small magnet. A material is magnetized when all (or most) of the atoms align magnetically. The interaction between these tiny magnets explains spontaneous magnetization.

Consider n sites (or atoms), equispaced on a circle, and associate with each site a magnetic dipole (spin) which can be either positive or negative. The state space is $\Omega = \{-1,1\}^n$. In order to describe magnetic behavior, Ising (1925) in his PhD dissertation introduced a probability measure in the following fashion. For each configuration $\mathbf{x} = (x_1,...,x_n)$ of n signs, associate an energy

$$U(\mathbf{x}) = -J\sum_{i\sim j} x_i x_j - mH\sum_i x_i, \tag{3.6}$$

where $i \sim j$ means that sites i and j are neighbors. This assumes that only neighboring sites affect any given site. The constant J describes the material. If $J > 0$ we have the **attractive** case: configurations with a lot of sign changes have larger energy. Since nature strives towards the lowest possible energy, an attractive material would tend to be magnetized quite easily. The opposite case is called **repulsive**, and magnetization of such materials is hard. The second term of (3.6) represents the influence of an external magnetic field of (signed) intensity H. The factor m is a property of the material. In the attractive case, energy is minimized when all sites line up (first term) in the direction of the external field (second term). We can write $U(\mathbf{x}) = \sum U_i(\mathbf{x})$, where

$$U_i(\mathbf{x}) = -J\sum_{|j-i|=1} x_i x_j - mHx_i$$

describes the energy contribution from site i.

To go from energy to probability we use thermodynamic considerations. The American physicist J. W. Gibbs showed that for a system in thermal equilibrium, the probability of finding the system in a state (or configuration) with energy E is proportional to $\exp(-E/kT)$ where T is the absolute temperature, and k is Boltzmanns constant. The Gibbs distribution of configuration $\mathbf{x}$, in the absence of an external magnetic field, is then

$$\pi(\mathbf{x}) = \frac{1}{Z} e^{\beta\sum_{i=1}^n x_i x_{i+1}},$$

where the normalizing constant

$$Z = \sum_{\mathbf{x}\in\{1,-1\}^n} e^{\beta\sum_{i=1}^n x_i x_{i+1}}$$

is called a partition function, and $\beta = J/kT$. In this particular example, Z can be computed using a transfer matrix method, but we will pretend that Z is not available to us.

To set up a Metropolis-Hastings algorithm, we need a proposal mechanism to move from one configuration to another. At each step, Let us choose a site uniformly at random and change the direction of the spin. This translates to the proposal probabilities

$$q(\mathbf{y}\,|\,\mathbf{x}) = q(\mathbf{x}\,|\,\mathbf{y}) = \begin{cases} \frac{1}{n} & \text{if } \mathbf{x} \text{ and } \mathbf{y} \text{ differ at exactly one location,} \\ 0 & \text{otherwise.} \end{cases}$$

If $\mathbf{x}^{(t)}$ is the current state of the Markov chain and $\mathbf{x}'$ is a proposed state with the jth site changed to the opposite direction, then

$$a_{\mathbf{x}^{(t)},\mathbf{x}'} = \frac{\pi(\mathbf{x}')\frac{1}{n}}{\pi(\mathbf{x}^{(t)})\frac{1}{n}} = \frac{e^{\beta\sum_{i\notin\{j,j-1\}} x_i^{(t)}x_{i+1}^{(t)}} e^{\beta(-x_{j-1}^{(t)}x_j^{(t)} - x_j^{(t)}x_{j+1}^{(t)})}}{e^{\beta\sum_{i\notin\{j,j-1\}} x_i^{(t)}x_{i+1}^{(t)}} e^{\beta(x_{j-1}^{(t)}x_j^{(t)} + x_j^{(t)}x_{j+1}^{(t)})}} = e^{-2\beta x_j^{(t)}(x_{j-1}^{(t)} + x_{j+1}^{(t)})}.$$

Clearly, this proposal mechanism makes it possible to get from any state to any other state of spin configurations, so the Metropolis-Hastings chain is irreducible.

Variants of Metropolis-Hastings:

1. $q(i\,|\,j) = q(j\,|\,i)$ - symmetric proposal. This is the original Metropolis algorithm. Here, the acceptance probability simplifies to $a_{ij} = \min\left\{\frac{\pi_j}{\pi_i}, 1\right\}$. So we move to a more probable state with probability 1, and move to less probable states sometimes (more rarely if the candidate is much less probable).
2. Independence sampler: $q(j\,|\,i) = q(j)$. Note this is *not* the same as iid sampling. Independence sampler is still a Markov chain, since the sampler can stay in the same place with some probability at each step of the algorithm.

Metropolis-Hastings algorithm can be executed without any difficulties on continuous state spaces. This requires defining Markov chains on continuous state spaces.

Definition. A sequence of r.v.s $X_0, X_1, \ldots$ is called a Markov chain on a state space Ω if $\forall t$ and $\forall A \subset \Omega$

$$\Pr(X_{n+1} \in A\,|\,X_n, X_{n-1}, X_0) = \Pr(X_{n+1} \in A\,|\,X_n) = [\text{in homogeneous case}] = \Pr(X_1 \in A\,|\,X_0).$$

A family of functions $\Pr(X_1 \in A\,|\,x) = K(x,A)$ is called **transition kernel**.

If there exists $f(x,y)$ such that

$$\Pr(X_1 \in A\,|\,x) = \int_A f(x,y)dy,$$

then $f(x,y)$ is called transition kernel density. This is a direct analog of a transition probability matrix in discrete state spaces.

A lot of notions transfer from discrete to continuous state spaces: irreducibility, periodicity, etc. Chapman-Kolmogorov, for example takes the following form:

$$K^{m+n}(x,A) = \int_\Omega K^n(y,A)K^m(x,dy),$$

where $K^n(x,A) = \Pr(X_n \in A\,|\,x)$.

Definition. A probability distribution Π on Ω is called a stationary distribution of a Markov process with transition kernel $K(x,A)$ if for any Borel set B in Ω

$$\Pi(B) = \int_\Omega K(x,B)\Pi(dx).$$

If transition kernel density is available and Π has density π, then global balance equation can be re-written

$$\pi(y) = \int_\Omega \pi(x)f(x,y)dx.$$

Using the introduced terminology, we define a Metropolis-Hastings algorithm for continuous state spaces. Let $f(\mathbf{x})$ be a target density, where $\mathbf{x}$ is a vector in $\mathbb{R}^n$ now. Then we simply can replace the proposal probabilities $q(j\,|\,i)$ with proposal densities $q(\mathbf{y}\,|\,\mathbf{x})$ so that the Metropolis-Hastings acceptance ratio becomes

$$a(\mathbf{x},\mathbf{y}) = \min\left\{\frac{f(\mathbf{y})q(\mathbf{x}\,|\,\mathbf{y})}{f(\mathbf{x})q(\mathbf{y}\,|\,\mathbf{x})}, 1\right\}. \tag{3.7}$$

The rest of the algorithm remains intact. As before, we need to ensure that the resulting Markov chain is irreducible. One way to do this is to require that $q(\mathbf{y}\,|\,\mathbf{x}) > 0$ for all $\mathbf{x},\mathbf{y} \in \Omega$. Alternately, a less restrictive assumption is that there exists some fixed δ and ε so that $q(\mathbf{y}\,|\,\mathbf{x}) > \varepsilon$ if $|\mathbf{x}-\mathbf{y}| < \delta$.

A common example of a proposal scheme is a **random walk**. The proposal is given by

$$Y = X_n + \varepsilon_n, \tag{3.8}$$

where ε_n is some random perturbation independent of X_n with $\mathrm{E}(\varepsilon_n) = 0$. By convention, random walk proposals are always taken to be symmetric and have the following form

$$q(y\,|\,x) = q(|y-x|). \tag{3.9}$$

Example: *Approximating standard normal distribution* Suppose our target is a univariate standard normal distribution with density $f(x) = 1/(\sqrt{2\pi})e^{-x^2/2}$. Given current state $x^{(t)}$, we generate two uniform r.v.s $U_1 \sim U[-\delta,\delta]$ and $U_2 \sim U[0,1]$. Then set

$$x^{(t+1)} = \begin{cases} x^{(t)} + U_1 & \text{if } U_2 \le \min\left\{e^{\left[\left(x^{(t)}\right)^2 - (x^{(t)}+U_1)^2\right]/2}, 1\right\} \\ x^{(t)} & \text{otherwise.} \end{cases}$$

δ is a tuning parameter. Large δ leads to small acceptance rate, small δ leads to slow exploration of the state space. The rule of thumb for random walk proposals is to keep acceptance probabilities around 30-40%. If your proposal is close to the target, then higher acceptance rates are favorable.

3.2.2 Combining Markov Kernels and Gibbs Algorithm

Suppose we have constructed m transition kernels with stationary distribution $\boldsymbol{\pi}$. In discrete state spaces, this means that we have m transition matrices, $\mathbf{P}_1,\ldots,\mathbf{P}_m$, where $\boldsymbol{\pi}^T\mathbf{P}_i = \boldsymbol{\pi}$ for all $i = 1,\ldots,m$. There are two simple ways to combine these transition kernels. First, we can construct a Markov chain, where at each step we sequentially generate new states from all kernels in a predetermined order. The transition probability matrix of this new Markov chain is

$$\mathbf{S} = \mathbf{P}_1 \times \cdots \times \mathbf{P}_m.$$

It is easy to show that $\boldsymbol{\pi}^T\mathbf{S} = \boldsymbol{\pi}$. So as long the new Markov chain is irreducible, we can use the ergodic theorem applied to the new Markov chain.

In the second method of combining Markov kernels, we first create a probability vector $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_m)$. Next, we first randomly select kernel i with probability α_i and then use this kernel to advance the Markov chain. The corresponding transition kernel is

$$\mathbf{R} = \sum_{i=1}^{m} \alpha_i \mathbf{P}_i.$$

Again, $\boldsymbol{\pi}^T\mathbf{R} = \boldsymbol{\pi}$, so this MCMC sampling strategy is valid as long as we can guarantee irreducibility.

Gibbs Algorithms

Suppose now that our state space is a Cartesian product of smaller subspaces, $\Omega = E_1 \times \cdots \times E_m$. The target distribution or density is $f(\mathbf{x})$ and we still want to calculate $\mathrm{E}_f[h(\mathbf{x})]$. We assume that we can sample from full conditional distributions $x_i \,|\, \mathbf{x}_{-i}$, where the notation $\mathbf{x}_{-i}$ means all elements of $\mathbf{x}$ except the ith component. It turns out that if keep iteratively sampling from these full conditionals, we will form a Markov chain with the required target distribution or density $f(\mathbf{x})$. More formally, consider the **sequential scan** Gibbs sampling algorithm below.

Algorithm 3 *Sequential Scan* Gibbs Sampling Algorithm: approximate $\mathrm{E}_{\mathbf{f}}[h(\mathbf{x})]$

1: Start with some initial value $\mathbf{x}^{(0)}$.
2: **for** $t = 0$ to N **do**
3: Sample $x_1^{(t+1)} \sim f_1\left(x_1 \,|\, \mathbf{x}_{-1}^{(t)}\right)$
4: Sample $x_2^{(t+1)} \sim f_2\left(x_2 \,|\, x_1^{(t+1)}, x_3^{(t)}, \ldots, x_p^{(t)}\right)$
$\vdots$
5: Sample $x_p^{(t+1)} \sim f_p\left(x_p \,|\, \mathbf{x}_{-p}^{(t+1)}\right)$
6: **return** $\frac{1}{N}\sum_{t=1}^{N} h(\mathbf{x}^{(t)})$.

The question remains why the Gibbs sampling algorithm actually works. Consider one possible move in the Gibbs sampling procedure from $\mathbf{x}^{\text{cur}} \to \mathbf{x}^{\text{new}}$, where $\mathbf{x}^{\text{new}}$ is obtained by replacing the ith component in $\mathbf{x}^{\text{cur}}$ with a draw from the full conditional $f_i\left(x_i \,|\, \mathbf{x}_{-i}^{\text{cur}}\right)$. Now, Let us view this "move" in light of the Metropolis-Hastings algorithm. Our proposal density will be the full conditional itself. Then the Metropolis-Hastings acceptance ratio becomes

$$a(\mathbf{x}^{\text{cur}}, \mathbf{x}^{\text{new}}) = \min\left\{\frac{f\left(x_i^{\text{new}}, \mathbf{x}_{-i}^{\text{cur}}\right) f_i\left(x_i^{\text{cur}} \,|\, \mathbf{x}_{-i}^{\text{cur}}\right)}{f\left(x_i^{\text{cur}}, \mathbf{x}_{-i}^{\text{cur}}\right) f\left(x_i^{\text{new}} \,|\, \mathbf{x}_{-i}^{\text{cur}}\right)}, 1\right\} = \min\left\{\frac{f\left(\mathbf{x}_{-i}^{\text{cur}}\right)}{f\left(\mathbf{x}_{-i}^{\text{cur}}\right)}, 1\right\} = 1. \tag{3.10}$$

So when we use full conditionals as our proposals in the Metropolis-Hastings step, we always accept. This means that drawing from a full conditional distribution produces a Markov chain with stationary distribution $f(\mathbf{x})$. Clearly, we can not keep updating just the ith component, because we will not be able to explore the whole state space this way. Therefore, we update each component in turn. This is not the only way to execute Gibbs sampling. We can also randomly select an component to update. This is called a **random scan** Gibbs sampling.

Algorithm 4 *Random Scan* Gibbs Sampling Algorithm: approximate $\mathrm{E}_{\mathbf{f}}[h(\mathbf{x})]$

1: Start with some initial value $\mathbf{x}_0$.
2: **for** $t = 0$ to N **do**
3: Sample index i by drawing a random variable with probability mass function $\{\alpha_1, \ldots, \alpha_m\}$.
4: Sample $x_i^{(t+1)} \sim f_i\left(x_i \mid \mathbf{x}_{-i}^{(t)}\right)$
5: **return** $\frac{1}{N}\sum_{t=1}^{N} h(\mathbf{x}^t)$.

Note 3.4. Although it is not obvious, in many cases sampling from full conditional distribution does not require knowing the normalizing constant of the target distribution.

Example: *Ising model (continued)* Recall that in the Ising model

$$\pi(\mathbf{x}) = \frac{1}{Z} e^{\beta \sum_{i=1}^{k} x_i x_{i+1}},$$

where $\mathbf{x} = (x_1, \ldots, x_k)$. The full conditional is

$$\pi(x_j \mid \mathbf{x}_{-j}) = \frac{\pi(\mathbf{x})}{\pi(\mathbf{x}_{-j})} = \frac{\pi(\mathbf{x})}{\sum_{y\in\{-1,1\}} \pi(y, \mathbf{x}_{-j})} = \frac{\frac{1}{Z} e^{\beta \sum_{i=1}^{k} x_i x_{i+1}}}{\frac{1}{Z} e^{\beta \sum_{i\notin\{j,j-1\}} x_i x_{i+1}} \left[e^{\beta(x_{j-1}+x_{j+1})} + e^{-\beta(x_{j-1}+x_{j+1})} \right]}$$

$$= \frac{e^{\beta(x_{j-1}x_j + x_j x_{j+1})}}{e^{\beta(x_{j-1}+x_{j+1})} + e^{-\beta(x_{j-1}+x_{j+1})}}.$$

Definition. Let $f(\mathbf{x}) = f(x_1, \ldots, x_m)$ be a density and $f_i(x)$, $i = 1, \ldots, m$ be the corresponding marginal densities. f satisfies the **positivity condition** if $f_i(x_i) > 0$ for all $i = 1, \ldots, m$ implies that $f(x_1, \ldots, x_m) > 0$.

Example: *Actually, a counterexample* Let (X_1, X_2) be r.v.s with joint probability mass function

X_1	X_2	$\Pr(X_1, X_2)$
0	0	0
0	1	$\alpha > 0$
1	0	$\beta > 0$
1	1	$1 - \alpha - \beta$

$$\Pr(X_1 = 0) = \Pr(X_1 = 0, X_2 = 0) + \Pr(X_1 = 0, X_2 = 1) = \alpha > 0,$$
$$\Pr(X_2 = 0) = \Pr(X_1 = 0, X_2 = 0) + \Pr(X_1 = 1, X_2 = 0) = \beta > 0.$$

$\Pr(X_1 = 0) > 0$, $\Pr(X_2 = 0) > 0 \not\Rightarrow \Pr(X_1 = 0, X_2 = 0) > 0$. Therefore, this probability mass function does not satisfy the positivity condition.

Proposition 3.2. *If the density f satisfies the positivity condition, then the full conditional distribution of the ith component does not reduce the range of possible values of this component. In other words, if the marginal density $f_i(x_i) > 0$, then $f(x_i \mid \mathbf{x}_{-i}) > 0$ for any $\mathbf{x}_{-i}$ such that $f(\mathbf{x}_{-i}) > 0$.*

Proof. $f(\mathbf{x}_{-i}) > 0 \Rightarrow f_j(x_j) > 0$ for all $j \neq i$. Since $f(x_i) > 0$, the positivity condition implies that $f(x_i, \mathbf{x}_{-i}) > 0$. Therefore,

$$f_i(x_i \mid \mathbf{x}_{-i}) = \frac{f(x_i, \mathbf{x}_{-i})}{f(\mathbf{x}_{-i})} > 0.$$

□

This proposition shows that if the target density satisfies the positivity condition, then the Gibbs sampler for this density is irreducible.

Example: *Nonconnected support* Let

$$f(x_1, x_2) = \frac{1}{2\pi^2} \left(1_{\{(x_1,x_2)\in\mathscr{C}_1\}} + 1_{\{(x_1,x_2)\in\mathscr{C}_2\}} \right),$$

where $\mathscr{C}_1$ is circle centered at $(1,1)$ with radius 1 and $\mathscr{C}_1$ is circle centered at $(-1,-1)$ with radius 1. If we start with $(x_1^{(0)}, x_2^{(0)}) = (0.5, 0.5)$, then if we execute a Gibbs sampler from this point, we will always stay in the positive quadrant of the plane. Change of variables that rotates the axes by 90 degrees in any direction solves the problem in this case.

The positivity condition is not easy to check in practice. Tierney (1994) provides alternative, more manageable, criteria.

Combining Gibbs and Metropolis-Hastings samplers

Our discussion of combining Markov kernels suggests that it it is possible to combine Gibbs and Metropolis-Hastings steps in MCMC sampler. Consider the following example.

Example: *Beta-binomial example continued* Let $\mathbf{x} = (x_1, \ldots, x_n)$, where $x_i \,|\, \theta_i \sim \text{Bin}(n_i, \theta_i)$ and x_is are independent given θ_is. We further assume that $\theta_i \overset{\text{iid}}{\sim} \text{Beta}(\alpha, \beta)$. We group all success probabilities into a vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$ and put a prior distribution on hyper-parameters α and β, $\Pr(\alpha, \beta)$. Under our assumptions, the posterior distribution becomes

$$\Pr(\boldsymbol{\theta}, \alpha, \beta \,|\, \mathbf{x}) \propto \Pr(\alpha, \beta) \prod_{i=1}^{n} \frac{\Gamma(\alpha, \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta_i^{x_i+\alpha-1} (1-\theta_i)^{n_i-x_i+\beta-1}.$$

The full conditional distribution of θ_i is

$$\Pr(\theta_i \,|\, \mathbf{x}, \alpha, \beta, \boldsymbol{\theta}_{-i}) \propto \theta^{x_i+\alpha-1} (1-\theta_i)^{n_i-x_i+\beta-1}.$$

Therefore,

$$\theta_i \mid \mathbf{x}, \alpha, \beta, \boldsymbol{\theta}_{-i} \sim \text{Beta}(x_i + \alpha, n_i - x_i + \beta).$$

Sampling from $\Pr(\alpha, \beta \,|\, \mathbf{x}, \boldsymbol{\theta})$ directly is difficult, so we will use two Metropolis-Hastings steps to update α and β.

To propose new values of α and β, we will multiply their current values by $e^{\lambda(U-0.5)}$, where $U \sim U[0,1]$ and λ is a tuning constant. The proposal density is

$$q(y_{\text{new}} \,|\, y_{\text{cur}}) = \frac{1}{\lambda y_{\text{new}}}.$$

This proposal is not symmetric, so we will have to include it into the M-H acceptance ratio.

Algorithm 5 MCMC for the beta-binomial hierarchical model

1: Start with some initial values $(\boldsymbol{\theta}^{(0)}, \alpha^{(0)}, \beta^{(0)})$.
2: **for** $t = 0$ to N **do**
3: **for** $i = 0$ to n **do**
4: Sample $\theta_i^{(t+1)} \sim \text{Beta}(x_i + \alpha^{(t)}, n_i - x_i + \beta^{(t)})$
5: Generate $U_1 \sim U[0,1]$ and set $\alpha^* = \alpha^{(t)} e^{\lambda_\alpha (U_1 - 0.5)}$. Generate $U_2 \sim U[0,1]$ and set

$$\alpha^{(t+1)} = \begin{cases} \alpha^* & \text{if } U_2 \leq \min\left\{ \dfrac{\Pr\left(\boldsymbol{\theta}^{(t+1)}, \alpha^*, \beta^{(t)} \,|\, \mathbf{x}\right) q(\alpha^{(t)} \,|\, \alpha^*)}{\Pr\left(\boldsymbol{\theta}^{(t+1)}, \alpha^{(t)}, \beta^{(t)} \,|\, \mathbf{x}\right) q(\alpha^* \,|\, \alpha^{(t)})}, 1 \right\}, \\ \alpha^{(t)} & \text{otherwise.} \end{cases}$$

6: Generate $U_3 \sim U[0,1]$ and set $\beta^* = \beta^{(t)} e^{\lambda_\beta (U_3 - 0.5)}$. Generate $U_4 \sim U[0,1]$ and set

$$\beta^{(t+1)} = \begin{cases} \beta^* & \text{if } U_4 \leq \min\left\{ \dfrac{\Pr\left(\boldsymbol{\theta}^{(t+1)}, \alpha^{(t+1)}, \beta^* \,|\, \mathbf{x}\right) q(\beta^{(t)} \,|\, \beta^*)}{\Pr\left(\boldsymbol{\theta}^{(t+1)}, \alpha^{(t+1)}, \beta^{(t)} \,|\, \mathbf{x}\right) q(\beta^* \,|\, \beta^{(t)})}, 1 \right\}, \\ \beta^{(t)} & \text{otherwise.} \end{cases}$$

7: **return** $(\boldsymbol{\theta}^{(t)}, \alpha^{(t)}, \beta^{(t)})$, for $t = 1, \ldots, N$.

3.2.3 Variance of MCMC Estimators and Convergence Diagnostics

Let $X_1, X_2, \ldots$ be an ergodic Markov chain and

$$\hat{h} = \frac{1}{N}\sum_{i=1}^{N} h(X_i)$$

be the corresponding estimate of $\mathrm{E}_f[h(X)]$, where f is the stationary distribution of the chain. Estimating the variance of this estimator is complicated by the dependence among $X_1, X_2, \ldots, X_N$. One simple way to get around it is to subsample the Markov chain output so that the resulting sample is approximately iid. Then, the variance can be approximated as before with

$$\hat{v} = \frac{1}{N^2}\sum_{i=1}^{N} [h(X_i) - \hat{h}]^2.$$

Subsampling can be wasteful and impractical for slow mixing chains. One way to quantify the loss of efficiency due to dependence among samples is to compute the effective sample size,

$$\hat{N}_{eff} = \frac{N}{\kappa_h},$$

where

$$\kappa_h = 1 + 2\sum_{i=1}^{\infty} \mathrm{corr}[h(X_0), h(X_i)]$$

is the autocorrelation time that can be estimated using spectral analysis for time series. After $\hat{N}_{eff}$ is obtained, the variance of $\hat{h}$ is computed as

$$\tilde{v} = \frac{1}{N}\frac{1}{N_{eff}}\sum_{i=1}^{N} [h(X_i) - \hat{h}]^2.$$

Calculating κ_h can be tricky, but the R package CODA (Plummer et al., 2006) has a reliable implementation.

Convergence diagnostics

Although there is no definitive way to tell whether one ran a Markov chain long enough to converge, several useful diagnostic tools can illuminate problems with the sampler, bugs in the code, and suggest ways to improve the design of the MCMC sampler. We organize these tools into the following categories:

1. Visualizing MCMC output. Trace plots provide a useful method for detecting problems with MCMC convergence and mixing. Ideally, trace plots of unnormalized log posterior and model parameters should look like stationary time series. Slowly mixing Markov chains produce trace plots with high autocorrelation, which can be further visualized by autocorrelation plots at different lags. Slow mixing does not imply lack of convergence.
2. Comparing batches. We take two vectors from MCMC output: $(\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(T/)2})$ and $(\boldsymbol{\theta}^{(T/2+1)}, \ldots, \boldsymbol{\theta}^{(T)})$. If MCMC achieved stationarity at the time of collecting these batches, then both vectors follow the same stationary distribution. To test this hypothesis, we can apply the Kolmogorov-Smirnov test, for example.
3. Renewal theory methods. Monitor return times of the Markov chain to a particular state and check whether these return times are iid. Care is needed on continuous state-spaces. See Mykland et al. (1995) for details.

4. Comparing multiple chains, started from random initial conditions. There are many ways of performing such a comparison. One popular method is called Potential Scale Reduction Factor (PSRF) due to Gelman and Rubin (1992).

Many useful diagnostic tools are implemented in R package CODA (Plummer et al., 2006). Cowles and Carlin (1995) and Brook and Roberts (1998) review many of the methods in depth.

3.2.4 Special topics

1. Perfect sampling. Strictly speaking perfect sampling is a Monte Carlo, not Markov chain Monte Carlo method. However, the algorithm relies on running Markov chains. Coupling these Markov chains in a certain way (coupling from the past), allows one to generate a sample from the stationary distribution exactly (Propp and Wilson, 1996).
2. Green (1995) formally introduced a Metropolis-Hastings algorithm for sampling parameter spaces with variable dimensions. This class of MCMC is called reversible jump MCMC (rjMCMC). Newton et al. (1992) and Arjas and Gasbarra (1994) have developed reversible jump procedure before Peter Green popularized these algorithms with his now classical 1995 paper.
3. Simulated tempering. Simulated tempering, proposed by Geyer and Thompson (1995), constructs a multivariate Markov chain $(X^{(1)},\ldots,X^{(n)}$ to sample from the vector-valued function $(f(\mathbf{x}),f^{1/\tau_1}(\mathbf{x}),\ldots,f^{1/\tau_n}(\mathbf{x}))^T$. The auxiliary "heated" chains allow for better exploration of multimodal targets. The idea is similar in spirit to simulated annealing (section 3.2.5).
4. Sequential importance sampling and particle filters. These methods are useful for sequential building of instrumental densities in high dimensions. The main idea is to use the following representation:

$$f(x_1,\ldots,x_n) = f(x_1 \mid x_2,\ldots,x_n)f(x_2 \mid x_3,\ldots,x_n)\cdots f(x_n).$$

Using specific structure of the problem at hand, conditioning often simplifies due to conditional independences (Liu and Chen, 1998; Chen et al., 2005).