MCMC Methods: Gibbs Sampling and the Metropolis-Hastings Algorithm

Patrick Lam

Outline

Introduction to Markov Chain Monte Carlo

Gibbs Sampling

The Metropolis-Hastings Algorithm

Outline

Introduction to Markov Chain Monte Carlo

Gibbs Sampling

The Metropolis-Hastings Algorithm

What is Markov Chain Monte Carlo (MCMC)?

What is Markov Chain Monte Carlo (MCMC)?

Markov Chain: a *stochastic process* in which future states are independent of past states given the present state

What is Markov Chain Monte Carlo (MCMC)?

Markov Chain: a *stochastic process* in which future states are independent of past states given the present state

Monte Carlo: simulation

What is Markov Chain Monte Carlo (MCMC)?

Markov Chain: a *stochastic process* in which future states are independent of past states given the present state

Monte Carlo: simulation

Up until now, we've done a lot of Monte Carlo simulation to find integrals rather than doing it analytically, a process called *Monte Carlo Integration*.

What is Markov Chain Monte Carlo (MCMC)?

Markov Chain: a *stochastic process* in which future states are independent of past states given the present state

Monte Carlo: simulation

Up until now, we've done a lot of Monte Carlo simulation to find integrals rather than doing it analytically, a process called *Monte Carlo Integration*.

Basically a fancy way of saying we can take quantities of interest of a distribution from simulated draws from the distribution.

Monte Carlo Integration

Monte Carlo Integration

Suppose we have a distribution $p(\theta)$ (perhaps a posterior) that we want to take quantities of interest from.

Monte Carlo Integration

Suppose we have a distribution $p(\theta)$ (perhaps a posterior) that we want to take quantities of interest from.

To derive it analytically, we need to take integrals:

$$I = \int_{\Theta} g(\theta) p(\theta) d\theta$$

where $g(\theta)$ is some function of θ ($g(\theta) = \theta$ for the mean and $g(\theta) = (\theta - E(\theta))^2$ for the variance).

Monte Carlo Integration

Suppose we have a distribution $p(\theta)$ (perhaps a posterior) that we want to take quantities of interest from.

To derive it analytically, we need to take integrals:

$$I = \int_{\Theta} g(\theta) p(\theta) d\theta$$

where $g(\theta)$ is some function of θ ($g(\theta) = \theta$ for the mean and $g(\theta) = (\theta - E(\theta))^2$ for the variance).

We can approximate the integrals via Monte Carlo Integration by simulating M values from $p(\theta)$ and calculating

$$\hat{I}_M = \frac{1}{M} \sum_{i=1}^{M} g(\theta^{(i)})$$

For example, we can compute the expected value of the Beta(3,3) distribution analytically:

For example, we can compute the expected value of the Beta(3,3) distribution analytically:

$$E(\theta) = \int_{\Theta} \theta p(\theta) d\theta = \int_{\Theta} \theta \frac{\Gamma(6)}{\Gamma(3)\Gamma(3)} \theta^2 (1-\theta)^2 d\theta = \frac{1}{2}$$

For example, we can compute the expected value of the Beta(3,3) distribution analytically:

$$E(\theta) = \int_{\Theta} \theta p(\theta) d\theta = \int_{\Theta} \theta \frac{\Gamma(6)}{\Gamma(3)\Gamma(3)} \theta^2 (1-\theta)^2 d\theta = \frac{1}{2}$$

or via Monte Carlo methods:

For example, we can compute the expected value of the Beta(3,3) distribution analytically:

$$E(\theta) = \int_{\Theta} \theta p(\theta) d\theta = \int_{\Theta} \theta \frac{\Gamma(6)}{\Gamma(3)\Gamma(3)} \theta^2 (1-\theta)^2 d\theta = \frac{1}{2}$$

or via Monte Carlo methods:

```
> M <- 10000
> beta.sims <- rbeta(M, 3, 3)
> sum(beta.sims)/M

[1] 0.5013
```

For example, we can compute the expected value of the Beta(3,3) distribution analytically:

$$E(\theta) = \int_{\Theta} \theta p(\theta) d\theta = \int_{\Theta} \theta \frac{\Gamma(6)}{\Gamma(3)\Gamma(3)} \theta^2 (1-\theta)^2 d\theta = \frac{1}{2}$$

or via Monte Carlo methods:

```
> M <- 10000
> beta.sims <- rbeta(M, 3, 3)
> sum(beta.sims)/M

[1] 0.5013
```

Our Monte Carlo approximation $\hat{I}_M$ is a simulation consistent estimator of the true value I:

For example, we can compute the expected value of the Beta(3,3) distribution analytically:

$$E(\theta) = \int_{\Theta} \theta p(\theta) d\theta = \int_{\Theta} \theta \frac{\Gamma(6)}{\Gamma(3)\Gamma(3)} \theta^2 (1-\theta)^2 d\theta = \frac{1}{2}$$

or via Monte Carlo methods:

```
> M <- 10000
> beta.sims <- rbeta(M, 3, 3)
> sum(beta.sims)/M

[1] 0.5013
```

Our Monte Carlo approximation $\hat{I}_M$ is a simulation consistent estimator of the true value I: $\hat{I}_M \rightarrow I$ as $M \rightarrow \infty$.

For example, we can compute the expected value of the Beta(3,3) distribution analytically:

$$E(\theta) = \int_{\Theta} \theta p(\theta) d\theta = \int_{\Theta} \theta \frac{\Gamma(6)}{\Gamma(3)\Gamma(3)} \theta^2 (1-\theta)^2 d\theta = \frac{1}{2}$$

or via Monte Carlo methods:

```
> M <- 10000
> beta.sims <- rbeta(M, 3, 3)
> sum(beta.sims)/M

[1] 0.5013
```

Our Monte Carlo approximation $\hat{I}_M$ is a simulation consistent estimator of the true value I: $\hat{I}_M \to I$ as $M \to \infty$.

We know this to be true from the Strong Law of Large Numbers.

Strong Law of Large Numbers (SLLN)

Strong Law of Large Numbers (SLLN)

Let $X_1, X_2, \ldots$ be a sequence of **independent** and identically distributed random variables, each having a finite mean $\mu = E(X_i)$.

Then with probability 1,

$$\frac{X_1 + X_2 + \cdots + X_M}{M} \to \mu \text{ as } M \to \infty$$

Strong Law of Large Numbers (SLLN)

Let $X_1, X_2, \ldots$ be a sequence of **independent** and identically distributed random variables, each having a finite mean $\mu = E(X_i)$.

Then with probability 1,

$$\frac{X_1 + X_2 + \cdots + X_M}{M} \rightarrow \mu \text{ as } M \rightarrow \infty$$

In our previous example, each simulation draw was **independent** and distributed from the same Beta(3,3) distribution.

Strong Law of Large Numbers (SLLN)

Let $X_1, X_2, \dots$ be a sequence of **independent** and identically distributed random variables, each having a finite mean $\mu = E(X_i)$.

Then with probability 1,

$$\frac{X_1 + X_2 + \cdots + X_M}{M} \rightarrow \mu \text{ as } M \rightarrow \infty$$

In our previous example, each simulation draw was **independent** and distributed from the same Beta(3,3) distribution.

This also works with variances and other quantities of interest, since a function of i.i.d. random variables are also i.i.d. random variables.

Strong Law of Large Numbers (SLLN)

Let $X_1, X_2, \ldots$ be a sequence of **independent** and identically distributed random variables, each having a finite mean $\mu = E(X_i)$.

Then with probability 1,

$$\frac{X_1 + X_2 + \cdots + X_M}{M} \to \mu \text{ as } M \to \infty$$

In our previous example, each simulation draw was **independent** and distributed from the same Beta(3,3) distribution.

This also works with variances and other quantities of interest, since a function of i.i.d. random variables are also i.i.d. random variables.

But what if we can't generate draws that are **independent**?

Suppose we want to draw from our posterior distribution $p(\theta|y)$, but we cannot sample independent draws from it.

Suppose we want to draw from our posterior distribution $p(\theta|y)$, but we cannot sample independent draws from it.

For example, we often do not know the normalizing constant.

Suppose we want to draw from our posterior distribution $p(\theta|y)$, but we cannot sample independent draws from it.

For example, we often do not know the normalizing constant.

However, we may be able to sample draws from $p(\theta|y)$ that are slightly dependent.

Suppose we want to draw from our posterior distribution $p(\theta|y)$, but we cannot sample independent draws from it.

For example, we often do not know the normalizing constant.

However, we may be able to sample draws from $p(\theta|y)$ that are slightly dependent.

If we can sample slightly dependent draws using a **Markov chain**, then we can still find quantities of interests from those draws.

What is a Markov Chain?

What is a Markov Chain?

Definition: a *stochastic process* in which future states are independent of past states given the present state

What is a Markov Chain?

Definition: a *stochastic process* in which future states are independent of past states given the present state

Stochastic process: a *consecutive* set of *random* (not deterministic) quantities defined on some known state space Θ.

What is a Markov Chain?

Definition: a *stochastic process* in which future states are independent of past states given the present state

Stochastic process: a *consecutive* set of *random* (not deterministic) quantities defined on some known state space Θ.

- think of Θ as our parameter space.

What is a Markov Chain?

Definition: a *stochastic process* in which future states are independent of past states given the present state

Stochastic process: a *consecutive* set of *random* (not deterministic) quantities defined on some known state space Θ.

- think of Θ as our parameter space.
- *consecutive* implies a time component, indexed by t.

What is a Markov Chain?

Definition: a *stochastic process* in which future states are independent of past states given the present state

Stochastic process: a *consecutive* set of *random* (not deterministic) quantities defined on some known state space Θ.

- think of Θ as our parameter space.
- *consecutive* implies a time component, indexed by t.

Consider a draw of $\boldsymbol{\theta}^{(t)}$ to be a state at iteration t. The next draw $\boldsymbol{\theta}^{(t+1)}$ is dependent only on the current draw $\boldsymbol{\theta}^{(t)}$, and not on any past draws.

What is a Markov Chain?

Definition: a *stochastic process* in which future states are independent of past states given the present state

Stochastic process: a *consecutive* set of *random* (not deterministic) quantities defined on some known state space Θ.

- think of Θ as our parameter space.
- *consecutive* implies a time component, indexed by t.

Consider a draw of $\boldsymbol{\theta}^{(t)}$ to be a state at iteration t. The next draw $\boldsymbol{\theta}^{(t+1)}$ is dependent only on the current draw $\boldsymbol{\theta}^{(t)}$, and not on any past draws.

This satisfies the **Markov property**:

What is a Markov Chain?

Definition: a *stochastic process* in which future states are independent of past states given the present state

Stochastic process: a *consecutive* set of *random* (not deterministic) quantities defined on some known state space Θ.

- think of Θ as our parameter space.
- *consecutive* implies a time component, indexed by t.

Consider a draw of $\boldsymbol{\theta}^{(t)}$ to be a state at iteration t. The next draw $\boldsymbol{\theta}^{(t+1)}$ is dependent only on the current draw $\boldsymbol{\theta}^{(t)}$, and not on any past draws.

This satisfies the **Markov property**:

$$p(\boldsymbol{\theta}^{(t+1)}|\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(t)}) = p(\boldsymbol{\theta}^{(t+1)}|\boldsymbol{\theta}^{(t)})$$

So our Markov chain is a bunch of draws of $\boldsymbol{\theta}$ that are each slightly dependent on the previous one.

So our Markov chain is a bunch of draws of $\boldsymbol{\theta}$ that are each slightly dependent on the previous one. The chain wanders around the parameter space, remembering only where it has been in the last period.

So our Markov chain is a bunch of draws of $\boldsymbol{\theta}$ that are each slightly dependent on the previous one. The chain wanders around the parameter space, remembering only where it has been in the last period.

What are the rules governing how the chain jumps from one state to another at each period?

So our Markov chain is a bunch of draws of $\boldsymbol{\theta}$ that are each slightly dependent on the previous one. The chain wanders around the parameter space, remembering only where it has been in the last period.

What are the rules governing how the chain jumps from one state to another at each period?

The jumping rules are governed by a **transition kernel**, which is a mechanism that describes the probability of moving to some other state based on the current state.

Transition Kernel

Transition Kernel

For discrete state space (k possible states):

Transition Kernel

For discrete state space (k possible states): a $k \times k$ matrix of transition probabilities.

Transition Kernel

For discrete state space (k possible states): a $k \times k$ matrix of transition probabilities.

Example: Suppose $k = 3$.

Transition Kernel

For discrete state space (k possible states): a $k \times k$ matrix of transition probabilities.

Example: Suppose $k = 3$. The 3×3 transition matrix $\mathbf{P}$ would be

$p(\boldsymbol{\theta}_{\mathbf{A}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{A}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{B}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{A}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{C}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{A}}^{(t)})$
$p(\boldsymbol{\theta}_{\mathbf{A}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{B}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{B}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{B}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{C}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{B}}^{(t)})$
$p(\boldsymbol{\theta}_{\mathbf{A}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{C}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{B}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{C}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{C}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{C}}^{(t)})$

where the subscripts index the 3 possible values that $\boldsymbol{\theta}$ can take.

Transition Kernel

For discrete state space (k possible states): a $k \times k$ matrix of transition probabilities.

Example: Suppose $k = 3$. The 3×3 transition matrix $\mathbf{P}$ would be

$p(\boldsymbol{\theta}_{\mathbf{A}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{A}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{B}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{A}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{C}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{A}}^{(t)})$
$p(\boldsymbol{\theta}_{\mathbf{A}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{B}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{B}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{B}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{C}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{B}}^{(t)})$
$p(\boldsymbol{\theta}_{\mathbf{A}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{C}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{B}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{C}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{C}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{C}}^{(t)})$

where the subscripts index the 3 possible values that $\boldsymbol{\theta}$ can take.

The rows sum to one and define a conditional PMF, conditional on the current state.

Transition Kernel

For discrete state space (k possible states): a $k \times k$ matrix of transition probabilities.

Example: Suppose $k = 3$. The 3×3 transition matrix $\mathbf{P}$ would be

$p(\boldsymbol{\theta}_{\mathbf{A}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{A}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{B}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{A}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{C}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{A}}^{(t)})$
$p(\boldsymbol{\theta}_{\mathbf{A}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{B}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{B}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{B}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{C}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{B}}^{(t)})$
$p(\boldsymbol{\theta}_{\mathbf{A}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{C}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{B}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{C}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{C}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{C}}^{(t)})$

where the subscripts index the 3 possible values that $\boldsymbol{\theta}$ can take.

The rows sum to one and define a conditional PMF, conditional on the current state. The columns are the marginal probabilities of being in a certain state in the next period.

Transition Kernel

For discrete state space (k possible states): a $k \times k$ matrix of transition probabilities.

Example: Suppose $k = 3$. The 3×3 transition matrix $\mathbf{P}$ would be

$p(\boldsymbol{\theta}_{\mathbf{A}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{A}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{B}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{A}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{C}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{A}}^{(t)})$
$p(\boldsymbol{\theta}_{\mathbf{A}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{B}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{B}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{B}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{C}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{B}}^{(t)})$
$p(\boldsymbol{\theta}_{\mathbf{A}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{C}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{B}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{C}}^{(t)})$	$p(\boldsymbol{\theta}_{\mathbf{C}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{C}}^{(t)})$

where the subscripts index the 3 possible values that $\boldsymbol{\theta}$ can take.

The rows sum to one and define a conditional PMF, conditional on the current state. The columns are the marginal probabilities of being in a certain state in the next period.

For continuous state space (infinite possible states), the transition kernel is a bunch of conditional PDFs: $f(\boldsymbol{\theta}_{\mathbf{j}}^{(t+1)} \mid \boldsymbol{\theta}_{\mathbf{i}}^{(t)})$

How Does a Markov Chain Work? (Discrete Example)

How Does a Markov Chain Work? (Discrete Example)

1. Define a starting distribution $\prod^{(0)}$ (a $1 \times k$ vector of probabilities that sum to one).

How Does a Markov Chain Work? (Discrete Example)

1. Define a starting distribution $\prod^{(0)}$ (a $1 \times k$ vector of probabilities that sum to one).

2. At iteration 1, our distribution $\prod^{(1)}$ (from which $\theta^{(1)}$ is drawn) is

$$\begin{array}{ccccc} \prod^{(1)} & = & \prod^{(0)} & \times & \mathbf{P} \\ (1 \times k) & & (1 \times k) & \times & (k \times k) \end{array}$$

How Does a Markov Chain Work? (Discrete Example)

1. Define a starting distribution $\prod^{(0)}$ (a $1 \times k$ vector of probabilities that sum to one).

2. At iteration 1, our distribution $\prod^{(1)}$ (from which $\boldsymbol{\theta}^{(1)}$ is drawn) is

$$\begin{array}{ccccc} \prod^{(1)} & = & \prod^{(0)} & \times & \mathbf{P} \\ (1 \times k) & & (1 \times k) & \times & (k \times k) \end{array}$$

3. At iteration 2, our distribution $\prod^{(2)}$ (from which $\boldsymbol{\theta}^{(2)}$ is drawn) is

$$\begin{array}{ccccc} \prod^{(2)} & = & \prod^{(1)} & \times & \mathbf{P} \\ (1 \times k) & & (1 \times k) & \times & (k \times k) \end{array}$$

How Does a Markov Chain Work? (Discrete Example)

1. Define a starting distribution $\prod^{(0)}$ (a $1 \times k$ vector of probabilities that sum to one).

2. At iteration 1, our distribution $\prod^{(1)}$ (from which $\boldsymbol{\theta}^{(1)}$ is drawn) is

$$\begin{array}{ccccc} \prod^{(1)} & = & \prod^{(0)} & \times & \mathbf{P} \\ (1 \times k) & & (1 \times k) & \times & (k \times k) \end{array}$$

3. At iteration 2, our distribution $\prod^{(2)}$ (from which $\boldsymbol{\theta}^{(2)}$ is drawn) is

$$\begin{array}{ccccc} \prod^{(2)} & = & \prod^{(1)} & \times & \mathbf{P} \\ (1 \times k) & & (1 \times k) & \times & (k \times k) \end{array}$$

4. At iteration t, our distribution $\prod^{(t)}$ (from which $\boldsymbol{\theta}^{(t)}$ is drawn) is $\prod^{(t)} = \prod^{(t-1)} \times \mathbf{P} = \prod^{(0)} \times \mathbf{P}^t$

Stationary (Limiting) Distribution

Stationary (Limiting) Distribution

Define a stationary distribution π to be some distribution $\prod$ such that $\pi = \pi\mathbf{P}$.

Stationary (Limiting) Distribution

Define a stationary distribution π to be some distribution $\prod$ such that $\pi = \pi\mathbf{P}$.

For all the MCMC algorithms we use in Bayesian statistics, the Markov chain will typically **converge** to π regardless of our starting points.

Stationary (Limiting) Distribution

Define a stationary distribution π to be some distribution $\prod$ such that $\pi = \pi \mathbf{P}$.

For all the MCMC algorithms we use in Bayesian statistics, the Markov chain will typically **converge** to π regardless of our starting points.

So if we can devise a Markov chain whose stationary distribution π is our desired posterior distribution $p(\boldsymbol{\theta}|y)$, then we can run this chain to get draws that are approximately from $p(\boldsymbol{\theta}|y)$ once the chain has converged.

Burn-in

Burn-in

Since convergence usually occurs regardless of our starting point, we can usually pick any feasible (for example, picking starting draws that are in the parameter space) starting point.

Burn-in

Since convergence usually occurs regardless of our starting point, we can usually pick any feasible (for example, picking starting draws that are in the parameter space) starting point.

However, the time it takes for the chain to converge varies depending on the starting point.

Burn-in

Since convergence usually occurs regardless of our starting point, we can usually pick any feasible (for example, picking starting draws that are in the parameter space) starting point.

However, the time it takes for the chain to converge varies depending on the starting point.

As a matter of practice, most people throw out a certain number of the first draws, known as the **burn-in**. This is to make our draws closer to the stationary distribution and less dependent on the starting point.

Burn-in

Since convergence usually occurs regardless of our starting point, we can usually pick any feasible (for example, picking starting draws that are in the parameter space) starting point.

However, the time it takes for the chain to converge varies depending on the starting point.

As a matter of practice, most people throw out a certain number of the first draws, known as the **burn-in**. This is to make our draws closer to the stationary distribution and less dependent on the starting point.

However, it is unclear how much we should **burn-in** since our draws are all slightly dependent and we don't know exactly when convergence occurs.

Monte Carlo Integration on the Markov Chain

Monte Carlo Integration on the Markov Chain

Once we have a Markov chain that has converged to the stationary distribution, then the draws in our chain appear to be like draws from $p(\boldsymbol{\theta}|y)$,

Monte Carlo Integration on the Markov Chain

Once we have a Markov chain that has converged to the stationary distribution, then the draws in our chain appear to be like draws from $p(\boldsymbol{\theta}|y)$, so it seems like we should be able to use Monte Carlo Integration methods to find quantities of interest.

Monte Carlo Integration on the Markov Chain

Once we have a Markov chain that has converged to the stationary distribution, then the draws in our chain appear to be like draws from $p(\boldsymbol{\theta}|y)$, so it seems like we should be able to use Monte Carlo Integration methods to find quantities of interest.

One problem: our draws are not independent, which we required for Monte Carlo Integration to work (remember SLLN).

Monte Carlo Integration on the Markov Chain

Once we have a Markov chain that has converged to the stationary distribution, then the draws in our chain appear to be like draws from $p(\boldsymbol{\theta}|y)$, so it seems like we should be able to use Monte Carlo Integration methods to find quantities of interest.

One problem: our draws are not independent, which we required for Monte Carlo Integration to work (remember SLLN).

Luckily, we have the **Ergodic Theorem**.

Ergodic Theorem

Ergodic Theorem

Let $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(M)}$ be M values from a Markov chain that is *aperiodic*, *irreducible*, and *positive recurrent* (then the chain is ergodic), and $E[g(\boldsymbol{\theta})] < \infty$.

Ergodic Theorem

Let $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(M)}$ be M values from a Markov chain that is *aperiodic*, *irreducible*, and *positive recurrent* (then the chain is ergodic), and $E[g(\boldsymbol{\theta})] < \infty$.

Then with probability 1,

$$\frac{1}{M}\sum_{i=1}^{M} g(\boldsymbol{\theta}_i) \rightarrow \int_{\Theta} g(\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}$$

as $M \rightarrow \infty$, where π is the stationary distribution.

Ergodic Theorem

Let $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(M)}$ be M values from a Markov chain that is *aperiodic*, *irreducible*, and *positive recurrent* (then the chain is ergodic), and $E[g(\boldsymbol{\theta})] < \infty$.

Then with probability 1,

$$\frac{1}{M}\sum_{i=1}^{M} g(\boldsymbol{\theta}_i) \rightarrow \int_{\Theta} g(\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}$$

as $M \rightarrow \infty$, where π is the stationary distribution.

This is the Markov chain analog to the SLLN,

Ergodic Theorem

Let $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(M)}$ be M values from a Markov chain that is *aperiodic*, *irreducible*, and *positive recurrent* (then the chain is ergodic), and $E[g(\boldsymbol{\theta})] < \infty$.

Then with probability 1,

$$\frac{1}{M}\sum_{i=1}^{M} g(\boldsymbol{\theta}_i) \rightarrow \int_{\Theta} g(\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}$$

as $M \rightarrow \infty$, where π is the stationary distribution.

This is the Markov chain analog to the SLLN, and it allows us to ignore the dependence between draws of the Markov chain when we calculate quantities of interest from the draws.

Ergodic Theorem

Let $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(M)}$ be M values from a Markov chain that is *aperiodic*, *irreducible*, and *positive recurrent* (then the chain is ergodic), and $E[g(\boldsymbol{\theta})] < \infty$.

Then with probability 1,

$$\frac{1}{M}\sum_{i=1}^{M} g(\boldsymbol{\theta}_i) \rightarrow \int_{\Theta} g(\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}$$

as $M \rightarrow \infty$, where π is the stationary distribution.

This is the Markov chain analog to the SLLN, and it allows us to ignore the dependence between draws of the Markov chain when we calculate quantities of interest from the draws.

But what does it mean for a chain to be *aperiodic*, *irreducible*, and *positive recurrent*, and therefore ergodic?

Aperiodicity

Aperiodicity

A Markov chain is **aperiodic** if the only length of time for which the chain repeats some cycle of values is the trivial case with cycle length equal to one.

Aperiodicity

A Markov chain is **aperiodic** if the only length of time for which the chain repeats some cycle of values is the trivial case with cycle length equal to one.

Let A, B, and C denote the states (analogous to the possible values of θ) in a 3-state Markov chain.

Aperiodicity

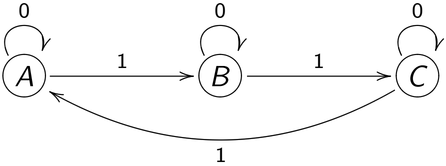
A Markov chain is **aperiodic** if the only length of time for which the chain repeats some cycle of values is the trivial case with cycle length equal to one.

Let A, B, and C denote the states (analogous to the possible values of $\boldsymbol{\theta}$) in a 3-state Markov chain. The following chain is *periodic* with period 3, where the period is the number of steps that it takes to return to a certain state.

Aperiodicity

A Markov chain is **aperiodic** if the only length of time for which the chain repeats some cycle of values is the trivial case with cycle length equal to one.

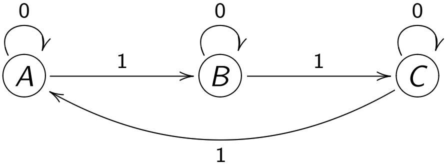
Let A, B, and C denote the states (analogous to the possible values of θ) in a 3-state Markov chain. The following chain is *periodic* with period 3, where the period is the number of steps that it takes to return to a certain state.



Aperiodicity

A Markov chain is **aperiodic** if the only length of time for which the chain repeats some cycle of values is the trivial case with cycle length equal to one.

Let A, B, and C denote the states (analogous to the possible values of θ) in a 3-state Markov chain. The following chain is *periodic* with period 3, where the period is the number of steps that it takes to return to a certain state.



As long as the chain is not repeating an identical cycle, then the chain is **aperiodic**.

Irreducibility

Irreducibility

A Markov chain is **irreducible** if it is possible go from any state to any other state (not necessarily in one step).

Irreducibility

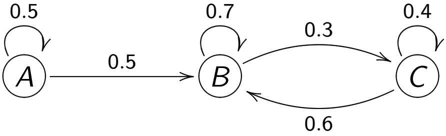
A Markov chain is **irreducible** if it is possible go from any state to any other state (not necessarily in one step).

The following chain is *reducible*, or not irreducible.

Irreducibility

A Markov chain is **irreducible** if it is possible go from any state to any other state (not necessarily in one step).

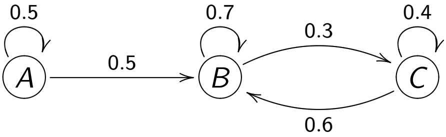
The following chain is *reducible*, or not irreducible.



Irreducibility

A Markov chain is **irreducible** if it is possible go from any state to any other state (not necessarily in one step).

The following chain is *reducible*, or not irreducible.



The chain is not irreducible because we cannot get to A from B or C regardless of the number of steps we take.

Positive Recurrence

Positive Recurrence

A Markov chain is *recurrent* if for any given state i, if the chain starts at i, it will eventually return to i with probability 1.

Positive Recurrence

A Markov chain is *recurrent* if for any given state i, if the chain starts at i, it will eventually return to i with probability 1.

A Markov chain is **positive recurrent** if the expected return time to state i is finite;

Positive Recurrence

A Markov chain is *recurrent* if for any given state i, if the chain starts at i, it will eventually return to i with probability 1.

A Markov chain is **positive recurrent** if the expected return time to state i is finite; otherwise it is *null recurrent*.

Positive Recurrence

A Markov chain is *recurrent* if for any given state i, if the chain starts at i, it will eventually return to i with probability 1.

A Markov chain is **positive recurrent** if the expected return time to state i is finite; otherwise it is *null recurrent*.

So if our Markov chain is **aperiodic**, **irreducible**, and **positive recurrent** (all the ones we use in Bayesian statistics usually are), then it is ergodic and the ergodic theorem allows us to do Monte Carlo Integration by calculating quantities of interest from our draws, ignoring the dependence between draws.

Thinning the Chain

Thinning the Chain

In order to break the dependence between draws in the Markov chain, some have suggested only keeping every dth draw of the chain.

Thinning the Chain

In order to break the dependence between draws in the Markov chain, some have suggested only keeping every dth draw of the chain.

This is known as **thinning**.

Thinning the Chain

In order to break the dependence between draws in the Markov chain, some have suggested only keeping every dth draw of the chain.

This is known as **thinning**.

Pros:

Thinning the Chain

In order to break the dependence between draws in the Markov chain, some have suggested only keeping every dth draw of the chain.

This is known as **thinning**.

Pros:

- Perhaps gets you a little closer to i.i.d. draws.

Thinning the Chain

In order to break the dependence between draws in the Markov chain, some have suggested only keeping every dth draw of the chain.

This is known as **thinning**.

Pros:

- Perhaps gets you a little closer to i.i.d. draws.
- Saves memory since you only store a fraction of the draws.

Thinning the Chain

In order to break the dependence between draws in the Markov chain, some have suggested only keeping every dth draw of the chain.

This is known as **thinning**.

Pros:

- Perhaps gets you a little closer to i.i.d. draws.
- Saves memory since you only store a fraction of the draws.

Cons:

Thinning the Chain

In order to break the dependence between draws in the Markov chain, some have suggested only keeping every dth draw of the chain.

This is known as **thinning**.

Pros:

- Perhaps gets you a little closer to i.i.d. draws.
- Saves memory since you only store a fraction of the draws.

Cons:

- Unnecessary with ergodic theorem.

Thinning the Chain

In order to break the dependence between draws in the Markov chain, some have suggested only keeping every dth draw of the chain.

This is known as **thinning**.

Pros:

- Perhaps gets you a little closer to i.i.d. draws.
- Saves memory since you only store a fraction of the draws.

Cons:

- Unnecessary with ergodic theorem.
- Shown to increase the variance of your Monte Carlo estimates.

So Really, What is MCMC?

So Really, What is MCMC?

MCMC is a class of methods in which we can simulate draws that are slightly dependent and are approximately from a (posterior) distribution.

So Really, What is MCMC?

MCMC is a class of methods in which we can simulate draws that are slightly dependent and are approximately from a (posterior) distribution.

We then take those draws and calculate quantities of interest for the (posterior) distribution.

So Really, What is MCMC?

MCMC is a class of methods in which we can simulate draws that are slightly dependent and are approximately from a (posterior) distribution.

We then take those draws and calculate quantities of interest for the (posterior) distribution.

In Bayesian statistics, there are generally two MCMC algorithms that we use: the Gibbs Sampler and the Metropolis-Hastings algorithm.

Outline

Introduction to Markov Chain Monte Carlo

Gibbs Sampling

The Metropolis-Hastings Algorithm

Gibbs Sampling

Gibbs Sampling

Suppose we have a joint distribution $p(\theta_1, \ldots, \theta_k)$ that we want to sample from (for example, a posterior distribution).

Gibbs Sampling

Suppose we have a joint distribution $p(\theta_1, \ldots, \theta_k)$ that we want to sample from (for example, a posterior distribution).

We can use the Gibbs sampler to sample from the joint distribution if we knew the **full conditional** distributions for each parameter.

Gibbs Sampling

Suppose we have a joint distribution $p(\theta_1, \ldots, \theta_k)$ that we want to sample from (for example, a posterior distribution).

We can use the Gibbs sampler to sample from the joint distribution if we knew the **full conditional** distributions for each parameter.

For each parameter, the **full conditional** distribution is the distribution of the parameter conditional on the known information and all the other parameters:

Gibbs Sampling

Suppose we have a joint distribution $p(\theta_1, \ldots, \theta_k)$ that we want to sample from (for example, a posterior distribution).

We can use the Gibbs sampler to sample from the joint distribution if we knew the **full conditional** distributions for each parameter.

For each parameter, the **full conditional** distribution is the distribution of the parameter conditional on the known information and all the other parameters: $p(\theta_j | \theta_{-j}, y)$

Gibbs Sampling

Suppose we have a joint distribution $p(\theta_1, \ldots, \theta_k)$ that we want to sample from (for example, a posterior distribution).

We can use the Gibbs sampler to sample from the joint distribution if we knew the **full conditional** distributions for each parameter.

For each parameter, the **full conditional** distribution is the distribution of the parameter conditional on the known information and all the other parameters: $p(\theta_j | \theta_{-j}, y)$

How can we know the joint distribution simply by knowing the full conditional distributions?

The Hammersley-Clifford Theorem (for two blocks)

The Hammersley-Clifford Theorem (for two blocks)

Suppose we have a joint density $f(x, y)$.

The Hammersley-Clifford Theorem (for two blocks)

Suppose we have a joint density $f(x, y)$. The theorem proves that we can write out the joint density in terms of the conditional densities $f(x|y)$ and $f(y|x)$:

The Hammersley-Clifford Theorem (for two blocks)

Suppose we have a joint density $f(x, y)$. The theorem proves that we can write out the joint density in terms of the conditional densities $f(x|y)$ and $f(y|x)$:

$$f(x, y) = \frac{f(y|x)}{\int \frac{f(y|x)}{f(x|y)} dy}$$

The Hammersley-Clifford Theorem (for two blocks)

Suppose we have a joint density $f(x, y)$. The theorem proves that we can write out the joint density in terms of the conditional densities $f(x|y)$ and $f(y|x)$:

$$f(x, y) = \frac{f(y|x)}{\int \frac{f(y|x)}{f(x|y)} dy}$$

We can write the denominator as

$$\int \frac{f(y|x)}{f(x|y)} dy \quad = \quad \int \frac{\frac{f(x,y)}{f(x)}}{\frac{f(x,y)}{f(y)}} dy$$

The Hammersley-Clifford Theorem (for two blocks)

Suppose we have a joint density $f(x, y)$. The theorem proves that we can write out the joint density in terms of the conditional densities $f(x|y)$ and $f(y|x)$:

$$f(x,y) = \frac{f(y|x)}{\int \frac{f(y|x)}{f(x|y)} dy}$$

We can write the denominator as

$$\begin{aligned} \int \frac{f(y|x)}{f(x|y)} dy &= \int \frac{\frac{f(x,y)}{f(x)}}{\frac{f(x,y)}{f(y)}} dy \\ &= \int \frac{f(y)}{f(x)} dy \end{aligned}$$

The Hammersley-Clifford Theorem (for two blocks)

Suppose we have a joint density $f(x,y)$. The theorem proves that we can write out the joint density in terms of the conditional densities $f(x|y)$ and $f(y|x)$:

$$f(x,y) = \frac{f(y|x)}{\int \frac{f(y|x)}{f(x|y)} dy}$$

We can write the denominator as

$$\begin{aligned}
\int \frac{f(y|x)}{f(x|y)} dy &= \int \frac{\frac{f(x,y)}{f(x)}}{\frac{f(x,y)}{f(y)}} dy \\
&= \int \frac{f(y)}{f(x)} dy \\
&= \frac{1}{f(x)}
\end{aligned}$$

Thus, our right-hand side is

$$\frac{f(y|x)}{\frac{1}{f(x)}} = f(y|x)f(x)$$

Thus, our right-hand side is

$$\begin{aligned} \frac{f(y|x)}{\frac{1}{f(x)}} &= f(y|x)f(x) \\ &= f(x,y) \end{aligned}$$

Thus, our right-hand side is

$$\begin{aligned} \frac{f(y|x)}{\frac{1}{f(x)}} &= f(y|x)f(x) \\ &= f(x,y) \end{aligned}$$

The theorem shows that knowledge of the conditional densities allows us to get the joint density.

Thus, our right-hand side is

$$\begin{aligned} \frac{f(y|x)}{\frac{1}{f(x)}} &= f(y|x)f(x) \\ &= f(x,y) \end{aligned}$$

The theorem shows that knowledge of the conditional densities allows us to get the joint density.

This works for more than two blocks of parameters.

Thus, our right-hand side is

$$\begin{aligned} \frac{f(y|x)}{\frac{1}{f(x)}} &= f(y|x)f(x) \\ &= f(x,y) \end{aligned}$$

The theorem shows that knowledge of the conditional densities allows us to get the joint density.

This works for more than two blocks of parameters.

But how do we figure out the full conditionals?

Steps to Calculating Full Conditional Distributions

Steps to Calculating Full Conditional Distributions

Suppose we have a posterior $p(\boldsymbol{\theta}|\mathbf{y})$.

Steps to Calculating Full Conditional Distributions

Suppose we have a posterior $p(\boldsymbol{\theta}|\mathbf{y})$. To calculate the full conditionals for each θ, do the following:

Steps to Calculating Full Conditional Distributions

Suppose we have a posterior $p(\boldsymbol{\theta}|\mathbf{y})$. To calculate the full conditionals for each θ, do the following:

1. Write out the full posterior ignoring constants of proportionality.

Steps to Calculating Full Conditional Distributions

Suppose we have a posterior $p(\boldsymbol{\theta}|\mathbf{y})$. To calculate the full conditionals for each θ, do the following:

1. Write out the full posterior ignoring constants of proportionality.
2. Pick a block of parameters (for example, θ_1) and drop everything that doesn't depend on θ_1.

Steps to Calculating Full Conditional Distributions

Suppose we have a posterior $p(\boldsymbol{\theta}|\mathbf{y})$. To calculate the full conditionals for each θ, do the following:

1. Write out the full posterior ignoring constants of proportionality.
2. Pick a block of parameters (for example, θ_1) and drop everything that doesn't depend on θ_1.
3. Use your knowledge of distributions to figure out what the normalizing constant is (and thus what the full conditional distribution $p(\theta_1|\theta_{-1}, \mathbf{y})$ is).

Steps to Calculating Full Conditional Distributions

Suppose we have a posterior $p(\boldsymbol{\theta}|\mathbf{y})$. To calculate the full conditionals for each θ, do the following:

1. Write out the full posterior ignoring constants of proportionality.
2. Pick a block of parameters (for example, θ_1) and drop everything that doesn't depend on θ_1.
3. Use your knowledge of distributions to figure out what the normalizing constant is (and thus what the full conditional distribution $p(\theta_1|\theta_{-1}, \mathbf{y})$ is).
4. Repeat steps 2 and 3 for all parameter blocks.

Gibbs Sampler Steps

Gibbs Sampler Steps

Let's suppose that we are interested in sampling from the posterior $p(\boldsymbol{\theta}|\mathbf{y})$, where $\boldsymbol{\theta}$ is a vector of three parameters, $\theta_1, \theta_2, \theta_3$.

Gibbs Sampler Steps

Let's suppose that we are interested in sampling from the posterior $p(\boldsymbol{\theta}|\mathbf{y})$, where $\boldsymbol{\theta}$ is a vector of three parameters, $\theta_1, \theta_2, \theta_3$.

The steps to a Gibbs Sampler (and the analogous steps in the MCMC process) are

Gibbs Sampler Steps

Let's suppose that we are interested in sampling from the posterior $p(\boldsymbol{\theta}|\mathbf{y})$, where $\boldsymbol{\theta}$ is a vector of three parameters, $\theta_1, \theta_2, \theta_3$.

The steps to a Gibbs Sampler (and the analogous steps in the MCMC process) are

1. Pick a vector of starting values $\boldsymbol{\theta}^{(0)}$. (Defining a starting distribution $\prod^{(0)}$ and drawing $\boldsymbol{\theta}^{(0)}$ from it.)

Gibbs Sampler Steps

Let's suppose that we are interested in sampling from the posterior $p(\boldsymbol{\theta}|\mathbf{y})$, where $\boldsymbol{\theta}$ is a vector of three parameters, $\theta_1, \theta_2, \theta_3$.

The steps to a Gibbs Sampler (and the analogous steps in the MCMC process) are

1. Pick a vector of starting values $\boldsymbol{\theta}^{(0)}$. (Defining a starting distribution $\prod^{(0)}$ and drawing $\boldsymbol{\theta}^{(0)}$ from it.)
2. Start with any θ (order does not matter, but I'll start with θ_1 for convenience).

Gibbs Sampler Steps

Let's suppose that we are interested in sampling from the posterior $p(\boldsymbol{\theta}|\mathbf{y})$, where $\boldsymbol{\theta}$ is a vector of three parameters, $\theta_1, \theta_2, \theta_3$.

The steps to a Gibbs Sampler (and the analogous steps in the MCMC process) are

1. Pick a vector of starting values $\boldsymbol{\theta}^{(0)}$. (Defining a starting distribution $\prod^{(0)}$ and drawing $\boldsymbol{\theta}^{(0)}$ from it.)
2. Start with any θ (order does not matter, but I'll start with θ_1 for convenience). Draw a value $\theta_1^{(1)}$ from the full conditional $p(\theta_1|\theta_2^{(0)}, \theta_3^{(0)}, \mathbf{y})$.

3. Draw a value $\theta_2^{(1)}$ (again order does not matter) from the full conditional $p(\theta_2|\theta_1^{(1)}, \theta_3^{(0)}, \mathbf{y})$.

3. Draw a value $\theta_2^{(1)}$ (again order does not matter) from the full conditional $p(\theta_2|\theta_1^{(1)}, \theta_3^{(0)}, \mathbf{y})$. Note that we must use the updated value of $\theta_1^{(1)}$.

3. Draw a value $\theta_2^{(1)}$ (again order does not matter) from the full conditional $p(\theta_2|\theta_1^{(1)}, \theta_3^{(0)}, \mathbf{y})$. Note that we must use the updated value of $\theta_1^{(1)}$.

4. Draw a value $\theta_3^{(1)}$ from the full conditional $p(\theta_3|\theta_1^{(1)}, \theta_2^{(1)}, \mathbf{y})$ using both updated values.

3. Draw a value $\theta_2^{(1)}$ (again order does not matter) from the full conditional $p(\theta_2|\theta_1^{(1)}, \theta_3^{(0)}, \mathbf{y})$. Note that we must use the updated value of $\theta_1^{(1)}$.

4. Draw a value $\theta_3^{(1)}$ from the full conditional $p(\theta_3|\theta_1^{(1)}, \theta_2^{(1)}, \mathbf{y})$ using both updated values. (Steps 2-4 are analogous to multiplying $\prod^{(0)}$ and $\mathbf{P}$ to get $\prod^{(1)}$ and then drawing $\boldsymbol{\theta}^{(1)}$ from $\prod^{(1)}$.)

3. Draw a value $\theta_2^{(1)}$ (again order does not matter) from the full conditional $p(\theta_2|\theta_1^{(1)}, \theta_3^{(0)}, \mathbf{y})$. Note that we must use the updated value of $\theta_1^{(1)}$.

4. Draw a value $\theta_3^{(1)}$ from the full conditional $p(\theta_3|\theta_1^{(1)}, \theta_2^{(1)}, \mathbf{y})$ using both updated values. (Steps 2-4 are analogous to multiplying $\prod^{(0)}$ and **P** to get $\prod^{(1)}$ and then drawing $\boldsymbol{\theta}^{(1)}$ from $\prod^{(1)}$.)

5. Draw $\boldsymbol{\theta}^{(2)}$ using $\boldsymbol{\theta}^{(1)}$ and continually using the most updated values.

3. Draw a value $\theta_2^{(1)}$ (again order does not matter) from the full conditional $p(\theta_2|\theta_1^{(1)}, \theta_3^{(0)}, \mathbf{y})$. Note that we must use the updated value of $\theta_1^{(1)}$.

4. Draw a value $\theta_3^{(1)}$ from the full conditional $p(\theta_3|\theta_1^{(1)}, \theta_2^{(1)}, \mathbf{y})$ using both updated values. (Steps 2-4 are analogous to multiplying $\prod^{(0)}$ and $\mathbf{P}$ to get $\prod^{(1)}$ and then drawing $\boldsymbol{\theta}^{(1)}$ from $\prod^{(1)}$.)

5. Draw $\boldsymbol{\theta}^{(2)}$ using $\boldsymbol{\theta}^{(1)}$ and continually using the most updated values.

6. Repeat until we get M draws, with each draw being a vector $\boldsymbol{\theta}^{(t)}$.

3. Draw a value $\theta_2^{(1)}$ (again order does not matter) from the full conditional $p(\theta_2|\theta_1^{(1)}, \theta_3^{(0)}, \mathbf{y})$. Note that we must use the updated value of $\theta_1^{(1)}$.

4. Draw a value $\theta_3^{(1)}$ from the full conditional $p(\theta_3|\theta_1^{(1)}, \theta_2^{(1)}, \mathbf{y})$ using both updated values. (Steps 2-4 are analogous to multiplying $\prod^{(0)}$ and $\mathbf{P}$ to get $\prod^{(1)}$ and then drawing $\boldsymbol{\theta}^{(1)}$ from $\prod^{(1)}$.)

5. Draw $\boldsymbol{\theta}^{(2)}$ using $\boldsymbol{\theta}^{(1)}$ and continually using the most updated values.

6. Repeat until we get M draws, with each draw being a vector $\boldsymbol{\theta}^{(t)}$.

7. Optional burn-in and/or thinning.

3. Draw a value $\theta_2^{(1)}$ (again order does not matter) from the full conditional $p(\theta_2|\theta_1^{(1)}, \theta_3^{(0)}, \mathbf{y})$. Note that we must use the updated value of $\theta_1^{(1)}$.

4. Draw a value $\theta_3^{(1)}$ from the full conditional $p(\theta_3|\theta_1^{(1)}, \theta_2^{(1)}, \mathbf{y})$ using both updated values. (Steps 2-4 are analogous to multiplying $\prod^{(0)}$ and $\mathbf{P}$ to get $\prod^{(1)}$ and then drawing $\boldsymbol{\theta}^{(1)}$ from $\prod^{(1)}$.)

5. Draw $\boldsymbol{\theta}^{(2)}$ using $\boldsymbol{\theta}^{(1)}$ and continually using the most updated values.

6. Repeat until we get M draws, with each draw being a vector $\boldsymbol{\theta}^{(t)}$.

7. Optional burn-in and/or thinning.

Our result is a Markov chain with a bunch of draws of $\boldsymbol{\theta}$ that are approximately from our posterior.

3. Draw a value $\theta_2^{(1)}$ (again order does not matter) from the full conditional $p(\theta_2|\theta_1^{(1)}, \theta_3^{(0)}, \mathbf{y})$. Note that we must use the updated value of $\theta_1^{(1)}$.
4. Draw a value $\theta_3^{(1)}$ from the full conditional $p(\theta_3|\theta_1^{(1)}, \theta_2^{(1)}, \mathbf{y})$ using both updated values. (Steps 2-4 are analogous to multiplying $\prod^{(0)}$ and **P** to get $\prod^{(1)}$ and then drawing $\boldsymbol{\theta}^{(1)}$ from $\prod^{(1)}$.)
5. Draw $\boldsymbol{\theta}^{(2)}$ using $\boldsymbol{\theta}^{(1)}$ and continually using the most updated values.
6. Repeat until we get M draws, with each draw being a vector $\boldsymbol{\theta}^{(t)}$.
7. Optional burn-in and/or thinning.

Our result is a Markov chain with a bunch of draws of $\boldsymbol{\theta}$ that are approximately from our posterior. We can do Monte Carlo Integration on those draws to get quantities of interest.

An Example (Robert and Casella, 10.17)[1]

[1] Robert, Christian P. and George Casella. 2004. *Monte Carlo Statistical Methods, 2nd edition*. Springer.

An Example (Robert and Casella, 10.17)[1]

Suppose we have data of the number of failures (y_i) for each of 10 pumps in a nuclear plant.

[1] Robert, Christian P. and George Casella. 2004. *Monte Carlo Statistical Methods, 2nd edition.* Springer.

An Example (Robert and Casella, 10.17)[1]

Suppose we have data of the number of failures (y_i) for each of 10 pumps in a nuclear plant.

We also have the times (t_i) at which each pump was observed.

[1] Robert, Christian P. and George Casella. 2004. *Monte Carlo Statistical Methods, 2nd edition.* Springer.

An Example (Robert and Casella, 10.17)[1]

Suppose we have data of the number of failures (y_i) for each of 10 pumps in a nuclear plant.

We also have the times (t_i) at which each pump was observed.

```
> y <- c(5, 1, 5, 14, 3, 19, 1, 1, 4, 22)
> t <- c(94, 16, 63, 126, 5, 31, 1, 1, 2, 10)
> rbind(y, t)

  [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
y    5    1    5   14    3   19    1    1    4    22
t   94   16   63  126    5   31    1    1    2    10
```

[1] Robert, Christian P. and George Casella. 2004. *Monte Carlo Statistical Methods, 2nd edition*. Springer.

An Example (Robert and Casella, 10.17)[1]

Suppose we have data of the number of failures (y_i) for each of 10 pumps in a nuclear plant.

We also have the times (t_i) at which each pump was observed.

```
> y <- c(5, 1, 5, 14, 3, 19, 1, 1, 4, 22)
> t <- c(94, 16, 63, 126, 5, 31, 1, 1, 2, 10)
> rbind(y, t)

  [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
y    5    1    5   14    3   19    1    1    4    22
t   94   16   63  126    5   31    1    1    2    10
```

We want to model the number of failures with a Poisson likelihood, where the expected number of failure λ_i differs for each pump.

[1] Robert, Christian P. and George Casella. 2004. *Monte Carlo Statistical Methods, 2nd edition.* Springer.

An Example (Robert and Casella, 10.17)[1]

Suppose we have data of the number of failures (y_i) for each of 10 pumps in a nuclear plant.

We also have the times (t_i) at which each pump was observed.

```
> y <- c(5, 1, 5, 14, 3, 19, 1, 1, 4, 22)
> t <- c(94, 16, 63, 126, 5, 31, 1, 1, 2, 10)
> rbind(y, t)

  [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
y    5    1    5   14    3   19    1    1    4    22
t   94   16   63  126    5   31    1    1    2    10
```

We want to model the number of failures with a Poisson likelihood, where the expected number of failure λ_i differs for each pump. Since the time which we observed each pump is different, we need to scale each λ_i by its observed time t_i.

[1] Robert, Christian P. and George Casella. 2004. *Monte Carlo Statistical Methods, 2nd edition*. Springer.

An Example (Robert and Casella, 10.17)[1]

Suppose we have data of the number of failures (y_i) for each of 10 pumps in a nuclear plant.

We also have the times (t_i) at which each pump was observed.

```
> y <- c(5, 1, 5, 14, 3, 19, 1, 1, 4, 22)
> t <- c(94, 16, 63, 126, 5, 31, 1, 1, 2, 10)
> rbind(y, t)

  [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
y    5    1    5   14    3   19    1    1    4    22
t   94   16   63  126    5   31    1    1    2    10
```

We want to model the number of failures with a Poisson likelihood, where the expected number of failure λ_i differs for each pump. Since the time which we observed each pump is different, we need to scale each λ_i by its observed time t_i.

Our likelihood is $\prod_{i=1}^{10} \mathrm{Poisson}(\lambda_i t_i)$.

[1] Robert, Christian P. and George Casella. 2004. *Monte Carlo Statistical Methods, 2nd edition.* Springer.

Let's put a Gamma(α, β) prior on λ_i with $\alpha = 1.8$, so the λ_is are drawn from the same distribution.

Let's put a Gamma(α, β) prior on λ_i with $\alpha = 1.8$, so the λ_is are drawn from the same distribution.

Also, let's put a Gamma(γ, δ) prior on β, with $\gamma = 0.01$ and $\delta = 1$.

Let's put a Gamma(α, β) prior on λ_i with $\alpha = 1.8$, so the λ_is are drawn from the same distribution.

Also, let's put a Gamma(γ, δ) prior on β, with $\gamma = 0.01$ and $\delta = 1$.

So our model has 11 parameters that are unknown (10 λ_is and β).

Let's put a Gamma(α, β) prior on λ_i with $\alpha = 1.8$, so the λ_is are drawn from the same distribution.

Also, let's put a Gamma(γ, δ) prior on β, with $\gamma = 0.01$ and $\delta = 1$.

So our model has 11 parameters that are unknown (10 λ_is and β).

Our posterior is

$$p(\boldsymbol{\lambda}, \beta | \mathbf{y}, \mathbf{t}) \quad \propto \quad \left(\prod_{i=1}^{10} \mathrm{Poisson}(\lambda_i t_i) \times \mathrm{Gamma}(\alpha, \beta) \right) \times \mathrm{Gamma}(\gamma, \delta)$$

Let's put a Gamma(α, β) prior on λ_i with $\alpha = 1.8$, so the λ_is are drawn from the same distribution.

Also, let's put a Gamma(γ, δ) prior on β, with $\gamma = 0.01$ and $\delta = 1$.

So our model has 11 parameters that are unknown (10 λ_is and β).

Our posterior is

$$\begin{aligned} p(\boldsymbol{\lambda}, \beta | \mathbf{y}, \mathbf{t}) &\propto \left(\prod_{i=1}^{10} \mathrm{Poisson}(\lambda_i t_i) \times \mathrm{Gamma}(\alpha, \beta) \right) \times \mathrm{Gamma}(\gamma, \delta) \\ &= \left(\prod_{i=1}^{10} \frac{e^{-\lambda_i t_i}(\lambda_i t_i)^{y_i}}{y_i!} \times \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda_i^{\alpha-1} e^{-\beta\lambda_i} \right) \\ &\quad \times \frac{\delta^\gamma}{\Gamma(\gamma)} \beta^{\gamma-1} e^{-\delta\beta} \end{aligned}$$

$$p(\boldsymbol{\lambda}, \beta|\mathbf{y}, \mathbf{t}) \propto \left(\prod_{i=1}^{10} e^{-\lambda_i t_i} (\lambda_i t_i)^{y_i} \times \beta^{\alpha} \lambda_i^{\alpha-1} e^{-\beta\lambda_i} \right) \times \beta^{\gamma-1} e^{-\delta\beta}$$

$$
\begin{aligned}
p(\boldsymbol{\lambda}, \beta | \mathbf{y}, \mathbf{t}) &\propto \left(\prod_{i=1}^{10} e^{-\lambda_i t_i} (\lambda_i t_i)^{y_i} \times \beta^{\alpha} \lambda_i^{\alpha-1} e^{-\beta \lambda_i} \right) \times \beta^{\gamma-1} e^{-\delta\beta} \\
&= \left(\prod_{i=1}^{10} \lambda_i^{y_i+\alpha-1} e^{-(t_i+\beta)\lambda_i} \right) \beta^{10\alpha+\gamma-1} e^{-\delta\beta}
\end{aligned}
$$

$$\begin{aligned} p(\boldsymbol{\lambda}, \beta | \mathbf{y}, \mathbf{t}) &\propto \left(\prod_{i=1}^{10} e^{-\lambda_i t_i} (\lambda_i t_i)^{y_i} \times \beta^{\alpha} \lambda_i^{\alpha-1} e^{-\beta \lambda_i} \right) \times \beta^{\gamma-1} e^{-\delta\beta} \\ &= \left(\prod_{i=1}^{10} \lambda_i^{y_i+\alpha-1} e^{-(t_i+\beta)\lambda_i} \right) \beta^{10\alpha+\gamma-1} e^{-\delta\beta} \end{aligned}$$

Finding the full conditionals:

$$
\begin{aligned}
p(\boldsymbol{\lambda}, \beta | \mathbf{y}, \mathbf{t}) &\propto \left(\prod_{i=1}^{10} e^{-\lambda_i t_i} (\lambda_i t_i)^{y_i} \times \beta^{\alpha} \lambda_i^{\alpha - 1} e^{-\beta \lambda_i} \right) \times \beta^{\gamma - 1} e^{-\delta \beta} \\
&= \left(\prod_{i=1}^{10} \lambda_i^{y_i + \alpha - 1} e^{-(t_i + \beta)\lambda_i} \right) \beta^{10\alpha + \gamma - 1} e^{-\delta \beta}
\end{aligned}
$$

Finding the full conditionals:

$$
p(\lambda_i | \lambda_{-i}, \beta, \mathbf{y}, \mathbf{t}) \propto \lambda_i^{y_i + \alpha - 1} e^{-(t_i + \beta)\lambda_i}
$$

$$
\begin{aligned}
p(\boldsymbol{\lambda}, \beta | \mathbf{y}, \mathbf{t}) &\propto \left(\prod_{i=1}^{10} e^{-\lambda_i t_i} (\lambda_i t_i)^{y_i} \times \beta^{\alpha} \lambda_i^{\alpha-1} e^{-\beta\lambda_i} \right) \times \beta^{\gamma-1} e^{-\delta\beta} \\
&= \left(\prod_{i=1}^{10} \lambda_i^{y_i+\alpha-1} e^{-(t_i+\beta)\lambda_i} \right) \beta^{10\alpha+\gamma-1} e^{-\delta\beta}
\end{aligned}
$$

Finding the full conditionals:

$$
\begin{aligned}
p(\lambda_i | \lambda_{-i}, \beta, \mathbf{y}, \mathbf{t}) &\propto \lambda_i^{y_i+\alpha-1} e^{-(t_i+\beta)\lambda_i} \\
p(\beta | \boldsymbol{\lambda}, \mathbf{y}, \mathbf{t}) &\propto e^{-\beta(\delta + \sum_{i=1}^{10} \lambda_i)} \beta^{10\alpha+\gamma-1}
\end{aligned}
$$

$$
\begin{aligned}
p(\boldsymbol{\lambda}, \beta | \mathbf{y}, \mathbf{t}) &\propto \left(\prod_{i=1}^{10} e^{-\lambda_i t_i} (\lambda_i t_i)^{y_i} \times \beta^{\alpha} \lambda_i^{\alpha-1} e^{-\beta\lambda_i} \right) \times \beta^{\gamma-1} e^{-\delta\beta} \\
&= \left(\prod_{i=1}^{10} \lambda_i^{y_i+\alpha-1} e^{-(t_i+\beta)\lambda_i} \right) \beta^{10\alpha+\gamma-1} e^{-\delta\beta}
\end{aligned}
$$

Finding the full conditionals:

$$
\begin{aligned}
p(\lambda_i | \lambda_{-i}, \beta, \mathbf{y}, \mathbf{t}) &\propto \lambda_i^{y_i+\alpha-1} e^{-(t_i+\beta)\lambda_i} \\
p(\beta | \boldsymbol{\lambda}, \mathbf{y}, \mathbf{t}) &\propto e^{-\beta(\delta + \sum_{i=1}^{10} \lambda_i)} \beta^{10\alpha+\gamma-1}
\end{aligned}
$$

$p(\lambda_i | \lambda_{-i}, \beta, \mathbf{y}, \mathbf{t})$ is a Gamma$(y_i + \alpha, t_i + \beta)$ distribution.

$$\begin{aligned} p(\boldsymbol{\lambda}, \beta | \mathbf{y}, \mathbf{t}) &\propto \left(\prod_{i=1}^{10} e^{-\lambda_i t_i} (\lambda_i t_i)^{y_i} \times \beta^{\alpha} \lambda_i^{\alpha - 1} e^{-\beta \lambda_i} \right) \times \beta^{\gamma - 1} e^{-\delta \beta} \\ &= \left(\prod_{i=1}^{10} \lambda_i^{y_i + \alpha - 1} e^{-(t_i + \beta) \lambda_i} \right) \beta^{10\alpha + \gamma - 1} e^{-\delta \beta} \end{aligned}$$

Finding the full conditionals:

$$\begin{aligned} p(\lambda_i | \lambda_{-i}, \beta, \mathbf{y}, \mathbf{t}) &\propto \lambda_i^{y_i + \alpha - 1} e^{-(t_i + \beta) \lambda_i} \\ p(\beta | \boldsymbol{\lambda}, \mathbf{y}, \mathbf{t}) &\propto e^{-\beta (\delta + \sum_{i=1}^{10} \lambda_i)} \beta^{10\alpha + \gamma - 1} \end{aligned}$$

$p(\lambda_i | \lambda_{-i}, \beta, \mathbf{y}, \mathbf{t})$ is a Gamma$(y_i + \alpha, t_i + \beta)$ distribution.

$p(\beta | \boldsymbol{\lambda}, \mathbf{y}, \mathbf{t})$ is a Gamma$(10\alpha + \gamma, \delta + \sum_{i=1}^{10} \lambda_i)$ distribution.

Coding the Gibbs Sampler

Coding the Gibbs Sampler

1. Define starting values for β

Coding the Gibbs Sampler

1. Define starting values for β (we only need to define β here because we will draw $\boldsymbol{\lambda}$ first and it only depends on β and other given values).

Coding the Gibbs Sampler

1. Define starting values for β (we only need to define β here because we will draw $\boldsymbol{\lambda}$ first and it only depends on β and other given values).

```
> beta.cur <- 1
```

Coding the Gibbs Sampler

1. Define starting values for β (we only need to define β here because we will draw $\boldsymbol{\lambda}$ first and it only depends on β and other given values).

```
> beta.cur <- 1
```

2. Draw $\boldsymbol{\lambda}^{(1)}$ from its full conditional

Coding the Gibbs Sampler

1. Define starting values for β (we only need to define β here because we will draw $\boldsymbol{\lambda}$ first and it only depends on β and other given values).

   ```
   > beta.cur <- 1
   ```

2. Draw $\boldsymbol{\lambda}^{(1)}$ from its full conditional (we're drawing all the λ_is as a block because they all only depend on β and not each other).

Coding the Gibbs Sampler

1. Define starting values for β (we only need to define β here because we will draw $\boldsymbol{\lambda}$ first and it only depends on β and other given values).

```
> beta.cur <- 1
```

2. Draw $\boldsymbol{\lambda}^{(1)}$ from its full conditional (we're drawing all the λ_is as a block because they all only depend on β and not each other).

```
> lambda.update <- function(alpha, beta, y, t) {
+     rgamma(length(y), y + alpha, t + beta)
+ }
```

Coding the Gibbs Sampler

1. Define starting values for β (we only need to define β here because we will draw $\boldsymbol{\lambda}$ first and it only depends on β and other given values).

```
> beta.cur <- 1
```

2. Draw $\boldsymbol{\lambda}^{(1)}$ from its full conditional (we're drawing all the λ_is as a block because they all only depend on β and not each other).

```
> lambda.update <- function(alpha, beta, y, t) {
+     rgamma(length(y), y + alpha, t + beta)
+ }
```

3. Draw $\beta^{(1)}$ from its full conditional, using $\boldsymbol{\lambda}^{(1)}$.

Coding the Gibbs Sampler

1. Define starting values for β (we only need to define β here because we will draw $\boldsymbol{\lambda}$ first and it only depends on β and other given values).

```
> beta.cur <- 1
```

2. Draw $\boldsymbol{\lambda}^{(1)}$ from its full conditional (we're drawing all the λ_is as a block because they all only depend on β and not each other).

```
> lambda.update <- function(alpha, beta, y, t) {
+     rgamma(length(y), y + alpha, t + beta)
+ }
```

3. Draw $\beta^{(1)}$ from its full conditional, using $\boldsymbol{\lambda}^{(1)}$.

```
> beta.update <- function(alpha, gamma, delta, lambda, y) {
+     rgamma(1, length(y) * alpha + gamma, delta + sum(lambda))
+ }
```

4. Repeat using most updated values until we get M draws.

4. Repeat using most updated values until we get M draws.

5. Optional burn-in and thinning.

4. Repeat using most updated values until we get M draws.

5. Optional burn-in and thinning.

6. Make it into a function.

```
> gibbs <- function(n.sims, beta.start, alpha, gamma, delta,
+     y, t, burnin = 0, thin = 1) {
+     beta.draws <- c()
+     lambda.draws <- matrix(NA, nrow = n.sims, ncol = length(y))
+     beta.cur <- beta.start
+     lambda.update <- function(alpha, beta, y, t) {
+         rgamma(length(y), y + alpha, t + beta)
+     }
+     beta.update <- function(alpha, gamma, delta, lambda,
+         y) {
+         rgamma(1, length(y) * alpha + gamma, delta + sum(lambda))
+     }
+     for (i in 1:n.sims) {
+         lambda.cur <- lambda.update(alpha = alpha, beta = beta.cur,
+             y = y, t = t)
+         beta.cur <- beta.update(alpha = alpha, gamma = gamma,
+             delta = delta, lambda = lambda.cur, y = y)
+         if (i > burnin & (i - burnin)%%thin == 0) {
+             lambda.draws[(i - burnin)/thin, ] <- lambda.cur
+             beta.draws[(i - burnin)/thin] <- beta.cur
+         }
+     }
+     return(list(lambda.draws = lambda.draws, beta.draws = beta.draws))
+ }
```

7. Do Monte Carlo Integration on the resulting Markov chain, which are samples approximately from the posterior.

7. Do Monte Carlo Integration on the resulting Markov chain, which are samples approximately from the posterior.

```
> posterior <- gibbs(n.sims = 10000, beta.start = 1, alpha = 1.8,
+     gamma = 0.01, delta = 1, y = y, t = t)
> colMeans(posterior$lambda.draws)

 [1] 0.07113 0.15098 0.10447 0.12321 0.65680 0.62212 0.86522 0.85465
 [9] 1.35524 1.92694

> mean(posterior$beta.draws)

[1] 2.389

> apply(posterior$lambda.draws, 2, sd)

 [1] 0.02759 0.08974 0.04012 0.03071 0.30899 0.13676 0.55689 0.54814
 [9] 0.60854 0.40812

> sd(posterior$beta.draws)

[1] 0.6986
```

Outline

Introduction to Markov Chain Monte Carlo

Gibbs Sampling

The Metropolis-Hastings Algorithm

Suppose we have a posterior $p(\boldsymbol{\theta}|\mathbf{y})$ that we want to sample from, but

Suppose we have a posterior $p(\boldsymbol{\theta}|\mathbf{y})$ that we want to sample from, but

- the posterior doesn't look like any distribution we know (no conjugacy)

Suppose we have a posterior $p(\boldsymbol{\theta}|\mathbf{y})$ that we want to sample from, but

- the posterior doesn't look like any distribution we know (no conjugacy)
- the posterior consists of more than 2 parameters (grid approximations intractable)

Suppose we have a posterior $p(\boldsymbol{\theta}|\mathbf{y})$ that we want to sample from, but

- the posterior doesn't look like any distribution we know (no conjugacy)
- the posterior consists of more than 2 parameters (grid approximations intractable)
- some (or all) of the full conditionals do not look like any distributions we know (no Gibbs sampling for those whose full conditionals we don't know)

Suppose we have a posterior $p(\boldsymbol{\theta}|\mathbf{y})$ that we want to sample from, but

- the posterior doesn't look like any distribution we know (no conjugacy)
- the posterior consists of more than 2 parameters (grid approximations intractable)
- some (or all) of the full conditionals do not look like any distributions we know (no Gibbs sampling for those whose full conditionals we don't know)

If all else fails, we can use the **Metropolis-Hastings** algorithm, which will always work.

Metropolis-Hastings Algorithm

Metropolis-Hastings Algorithm

The Metropolis-Hastings Algorithm follows the following steps:

Metropolis-Hastings Algorithm

The Metropolis-Hastings Algorithm follows the following steps:

1. Choose a starting value $\boldsymbol{\theta}^{(0)}$.

Metropolis-Hastings Algorithm

The Metropolis-Hastings Algorithm follows the following steps:

1. Choose a starting value $\boldsymbol{\theta}^{(0)}$.
2. At iteration t, draw a candidate $\boldsymbol{\theta}^*$ from a jumping distribution $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$.

Metropolis-Hastings Algorithm

The Metropolis-Hastings Algorithm follows the following steps:

1. Choose a starting value $\boldsymbol{\theta}^{(0)}$.
2. At iteration t, draw a candidate $\boldsymbol{\theta}^*$ from a jumping distribution $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$.
3. Compute an acceptance ratio (probability):

$$r = \frac{p(\boldsymbol{\theta}^*|\mathbf{y})/J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})}{p(\boldsymbol{\theta}^{(t-1)}|\mathbf{y})/J_t(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)}$$

Metropolis-Hastings Algorithm

The Metropolis-Hastings Algorithm follows the following steps:

1. Choose a starting value $\boldsymbol{\theta}^{(0)}$.
2. At iteration t, draw a candidate $\boldsymbol{\theta}^*$ from a jumping distribution $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$.
3. Compute an acceptance ratio (probability):

$$r = \frac{p(\boldsymbol{\theta}^*|\mathbf{y})/J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})}{p(\boldsymbol{\theta}^{(t-1)}|\mathbf{y})/J_t(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)}$$

4. Accept $\boldsymbol{\theta}^*$ as $\boldsymbol{\theta}^{(t)}$ with probability $\min(r, 1)$. If $\boldsymbol{\theta}^*$ is not accepted, then $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)}$.

Metropolis-Hastings Algorithm

The Metropolis-Hastings Algorithm follows the following steps:

1. Choose a starting value $\boldsymbol{\theta}^{(0)}$.
2. At iteration t, draw a candidate $\boldsymbol{\theta}^*$ from a jumping distribution $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$.
3. Compute an acceptance ratio (probability):

$$r = \frac{p(\boldsymbol{\theta}^*|\mathbf{y})/J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})}{p(\boldsymbol{\theta}^{(t-1)}|\mathbf{y})/J_t(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)}$$

4. Accept $\boldsymbol{\theta}^*$ as $\boldsymbol{\theta}^{(t)}$ with probability $\min(r,1)$. If $\boldsymbol{\theta}^*$ is not accepted, then $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)}$.
5. Repeat steps 2-4 M times to get M draws from $p(\boldsymbol{\theta}|\mathbf{y})$, with optional burn-in and/or thinning.

Step 1: Choose a starting value $\boldsymbol{\theta}^{(0)}$.

Step 1: Choose a starting value $\boldsymbol{\theta}^{(0)}$.

This is equivalent to drawing from our initial stationary distribution.

Step 1: Choose a starting value $\boldsymbol{\theta}^{(0)}$.

This is equivalent to drawing from our initial stationary distribution.

The important thing to remember is that $\boldsymbol{\theta}^{(0)}$ must have positive probability.

$$p(\boldsymbol{\theta}^{(0)}|\mathbf{y}) > 0$$

Step 1: Choose a starting value $\boldsymbol{\theta}^{(0)}$.

This is equivalent to drawing from our initial stationary distribution.

The important thing to remember is that $\boldsymbol{\theta}^{(0)}$ must have positive probability.

$$p(\boldsymbol{\theta}^{(0)}|\mathbf{y}) > 0$$

Otherwise, we are starting with a value that cannot be drawn.

Step 2: Draw $\boldsymbol{\theta}^*$ from $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$.

Step 2: Draw $\boldsymbol{\theta}^*$ from $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$.

The jumping distribution $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$ determines where we move to in the next iteration of the Markov chain (analogous to the transition kernel).

Step 2: Draw $\boldsymbol{\theta}^*$ from $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$.

The jumping distribution $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$ determines where we move to in the next iteration of the Markov chain (analogous to the transition kernel). The support of the jumping distribution must contain the support of the posterior.

Step 2: Draw $\boldsymbol{\theta}^*$ from $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$.

The jumping distribution $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$ determines where we move to in the next iteration of the Markov chain (analogous to the transition kernel). The support of the jumping distribution must contain the support of the posterior.

The original **Metropolis algorithm** required that $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$ be a symmetric distribution (such as the normal distribution),

Step 2: Draw $\boldsymbol{\theta}^*$ from $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$.

The jumping distribution $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$ determines where we move to in the next iteration of the Markov chain (analogous to the transition kernel). The support of the jumping distribution must contain the support of the posterior.

The original **Metropolis algorithm** required that $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$ be a symmetric distribution (such as the normal distribution), that is

$$J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)}) = J_t(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)$$

Step 2: Draw $\boldsymbol{\theta}^*$ from $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$.

The jumping distribution $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$ determines where we move to in the next iteration of the Markov chain (analogous to the transition kernel). The support of the jumping distribution must contain the support of the posterior.

The original **Metropolis algorithm** required that $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$ be a symmetric distribution (such as the normal distribution), that is

$$J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)}) = J_t(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)$$

We now know with the Metropolis-Hastings algorithm that symmetry is unnecessary.

Step 2: Draw $\boldsymbol{\theta}^*$ from $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$.

The jumping distribution $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$ determines where we move to in the next iteration of the Markov chain (analogous to the transition kernel). The support of the jumping distribution must contain the support of the posterior.

The original **Metropolis algorithm** required that $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$ be a symmetric distribution (such as the normal distribution), that is

$$J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)}) = J_t(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)$$

We now know with the Metropolis-Hastings algorithm that symmetry is unnecessary.

If we have a symmetric jumping distribution that is dependent on $\boldsymbol{\theta}^{(t-1)}$, then we have what is known as **random walk Metropolis sampling**.

If our jumping distribution does not depend on $\boldsymbol{\theta}^{(t-1)}$,

If our jumping distribution does not depend on $\boldsymbol{\theta}^{(t-1)}$,

$$J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)}) = J_t(\boldsymbol{\theta}^*)$$

If our jumping distribution does not depend on $\boldsymbol{\theta}^{(t-1)}$,

$$J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)}) = J_t(\boldsymbol{\theta}^*)$$

then we have what is known as **independent Metropolis-Hastings sampling**.

If our jumping distribution does not depend on $\boldsymbol{\theta}^{(t-1)}$,

$$J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)}) = J_t(\boldsymbol{\theta}^*)$$

then we have what is known as **independent Metropolis-Hastings sampling**.

Basically all our candidate draws $\boldsymbol{\theta}^*$ are drawn from the same distribution, regardless of where the previous draw was.

If our jumping distribution does not depend on $\boldsymbol{\theta}^{(t-1)}$,

$$J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)}) = J_t(\boldsymbol{\theta}^*)$$

then we have what is known as **independent Metropolis-Hastings sampling**.

Basically all our candidate draws $\boldsymbol{\theta}^*$ are drawn from the same distribution, regardless of where the previous draw was.

This can be extremely efficient or extremely inefficient, depending on how close the jumping distribution is to the posterior.

If our jumping distribution does not depend on $\boldsymbol{\theta}^{(t-1)}$,

$$J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)}) = J_t(\boldsymbol{\theta}^*)$$

then we have what is known as **independent Metropolis-Hastings sampling**.

Basically all our candidate draws $\boldsymbol{\theta}^*$ are drawn from the same distribution, regardless of where the previous draw was.

This can be extremely efficient or extremely inefficient, depending on how close the jumping distribution is to the posterior.

Generally speaking, chain will behave well only if the jumping distribution has heavier tails than the posterior.

Step 3: Compute acceptance ratio r.

Step 3: Compute acceptance ratio r.

$$r = \frac{p(\boldsymbol{\theta}^*|\mathbf{y})/J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})}{p(\boldsymbol{\theta}^{(t-1)}|\mathbf{y})/J_t(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)}$$

Step 3: Compute acceptance ratio r.

$$r = \frac{p(\boldsymbol{\theta}^*|\mathbf{y})/J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})}{p(\boldsymbol{\theta}^{(t-1)}|\mathbf{y})/J_t(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)}$$

In the case where our jumping distribution is symmetric,

$$r = \frac{p(\boldsymbol{\theta}^*|\mathbf{y})}{p(\boldsymbol{\theta}^{(t-1)}|\mathbf{y})}$$

Step 3: Compute acceptance ratio r.

$$r = \frac{p(\boldsymbol{\theta}^*|\mathbf{y})/J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})}{p(\boldsymbol{\theta}^{(t-1)}|\mathbf{y})/J_t(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)}$$

In the case where our jumping distribution is symmetric,

$$r = \frac{p(\boldsymbol{\theta}^*|\mathbf{y})}{p(\boldsymbol{\theta}^{(t-1)}|\mathbf{y})}$$

If our candidate draw has higher probability than our current draw, then our candidate is better so we definitely accept it.

Step 3: Compute acceptance ratio r.

$$r = \frac{p(\boldsymbol{\theta}^*|\mathbf{y})/J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})}{p(\boldsymbol{\theta}^{(t-1)}|\mathbf{y})/J_t(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)}$$

In the case where our jumping distribution is symmetric,

$$r = \frac{p(\boldsymbol{\theta}^*|\mathbf{y})}{p(\boldsymbol{\theta}^{(t-1)}|\mathbf{y})}$$

If our candidate draw has higher probability than our current draw, then our candidate is better so we definitely accept it. Otherwise, our candidate is accepted according to the ratio of the probabilities of the candidate and current draws.

Step 3: Compute acceptance ratio r.

$$r = \frac{p(\boldsymbol{\theta}^*|\mathbf{y})/J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})}{p(\boldsymbol{\theta}^{(t-1)}|\mathbf{y})/J_t(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)}$$

In the case where our jumping distribution is symmetric,

$$r = \frac{p(\boldsymbol{\theta}^*|\mathbf{y})}{p(\boldsymbol{\theta}^{(t-1)}|\mathbf{y})}$$

If our candidate draw has higher probability than our current draw, then our candidate is better so we definitely accept it. Otherwise, our candidate is accepted according to the ratio of the probabilities of the candidate and current draws.

Note that since r is a ratio, we only need $p(\boldsymbol{\theta}|\mathbf{y})$ *up to a constant of proportionality* since $p(\mathbf{y})$ cancels out in both the numerator and denominator.

In the case where our jumping distribution is not symmetric,

$$r = \frac{p(\boldsymbol{\theta}^*|\mathbf{y})/J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})}{p(\boldsymbol{\theta}^{(t-1)}|\mathbf{y})/J_t(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)}$$

In the case where our jumping distribution is not symmetric,

$$r = \frac{p(\boldsymbol{\theta}^*|\mathbf{y})/J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})}{p(\boldsymbol{\theta}^{(t-1)}|\mathbf{y})/J_t(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)}$$

We need to weight our evaluations of the draws at the posterior densities by how likely we are to draw each draw.

In the case where our jumping distribution is not symmetric,

$$r = \frac{p(\boldsymbol{\theta}^*|\mathbf{y})/J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})}{p(\boldsymbol{\theta}^{(t-1)}|\mathbf{y})/J_t(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)}$$

We need to weight our evaluations of the draws at the posterior densities by how likely we are to draw each draw.

For example, if we are very likely to jump to some $\boldsymbol{\theta}^*$, then $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$ is likely to be high, so we should accept less of them than some other $\boldsymbol{\theta}^*$ that we are less likely to jump to.

In the case where our jumping distribution is not symmetric,

$$r = \frac{p(\boldsymbol{\theta}^*|\mathbf{y})/J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})}{p(\boldsymbol{\theta}^{(t-1)}|\mathbf{y})/J_t(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)}$$

We need to weight our evaluations of the draws at the posterior densities by how likely we are to draw each draw.

For example, if we are very likely to jump to some $\boldsymbol{\theta}^*$, then $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$ is likely to be high, so we should accept less of them than some other $\boldsymbol{\theta}^*$ that we are less likely to jump to.

In the case of independent Metropolis-Hastings sampling,

$$r = \frac{p(\boldsymbol{\theta}^*|\mathbf{y})/J_t(\boldsymbol{\theta}^*)}{p(\boldsymbol{\theta}^{(t-1)}|\mathbf{y})/J_t(\boldsymbol{\theta}^{(t-1)})}$$

Step 4: Decide whether to accept $\boldsymbol{\theta}^*$.

Step 4: Decide whether to accept $\boldsymbol{\theta}^*$.

Accept $\boldsymbol{\theta}^*$ as $\boldsymbol{\theta}^{(t)}$ with probability $\min(r, 1)$. If $\boldsymbol{\theta}^*$ is not accepted, then $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)}$.

Step 4: Decide whether to accept $\boldsymbol{\theta}^*$.

Accept $\boldsymbol{\theta}^*$ as $\boldsymbol{\theta}^{(t)}$ with probability $\min(r, 1)$. If $\boldsymbol{\theta}^*$ is not accepted, then $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)}$.

1. For each $\boldsymbol{\theta}^*$, draw a value u from the Uniform(0,1) distribution.

Step 4: Decide whether to accept $\boldsymbol{\theta}^*$.

Accept $\boldsymbol{\theta}^*$ as $\boldsymbol{\theta}^{(t)}$ with probability $\min(r, 1)$. If $\boldsymbol{\theta}^*$ is not accepted, then $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)}$.

1. For each $\boldsymbol{\theta}^*$, draw a value u from the Uniform(0,1) distribution.
2. If $u \leq r$, accept $\boldsymbol{\theta}^*$ as $\boldsymbol{\theta}^{(t)}$. Otherwise, use $\boldsymbol{\theta}^{(t-1)}$ as $\boldsymbol{\theta}^{(t)}$

Step 4: Decide whether to accept $\boldsymbol{\theta}^*$.

Accept $\boldsymbol{\theta}^*$ as $\boldsymbol{\theta}^{(t)}$ with probability $\min(r, 1)$. If $\boldsymbol{\theta}^*$ is not accepted, then $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)}$.

1. For each $\boldsymbol{\theta}^*$, draw a value u from the Uniform(0,1) distribution.
2. If $u \leq r$, accept $\boldsymbol{\theta}^*$ as $\boldsymbol{\theta}^{(t)}$. Otherwise, use $\boldsymbol{\theta}^{(t-1)}$ as $\boldsymbol{\theta}^{(t)}$

Candidate draws with higher density than the current draw are always accepted.

Step 4: Decide whether to accept $\boldsymbol{\theta}^*$.

Accept $\boldsymbol{\theta}^*$ as $\boldsymbol{\theta}^{(t)}$ with probability $\min(r, 1)$. If $\boldsymbol{\theta}^*$ is not accepted, then $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)}$.

1. For each $\boldsymbol{\theta}^*$, draw a value u from the Uniform(0,1) distribution.
2. If $u \leq r$, accept $\boldsymbol{\theta}^*$ as $\boldsymbol{\theta}^{(t)}$. Otherwise, use $\boldsymbol{\theta}^{(t-1)}$ as $\boldsymbol{\theta}^{(t)}$

Candidate draws with higher density than the current draw are always accepted.

Unlike in rejection sampling, each iteration always produces a draw, either $\boldsymbol{\theta}^*$ or $\boldsymbol{\theta}^{(t-1)}$.

Acceptance Rates

Acceptance Rates

It is important to monitor the *acceptance rate* (the fraction of candidate draws that are accepted) of your Metropolis-Hastings algorithm.

Acceptance Rates

It is important to monitor the *acceptance rate* (the fraction of candidate draws that are accepted) of your Metropolis-Hastings algorithm.

If your acceptance rate is too high, the chain is probably not mixing well (not moving around the parameter space quickly enough).

Acceptance Rates

It is important to monitor the *acceptance rate* (the fraction of candidate draws that are accepted) of your Metropolis-Hastings algorithm.

If your acceptance rate is too high, the chain is probably not mixing well (not moving around the parameter space quickly enough).

If your acceptance rate is too low, your algorithm is too inefficient (rejecting too many candidate draws).

Acceptance Rates

It is important to monitor the *acceptance rate* (the fraction of candidate draws that are accepted) of your Metropolis-Hastings algorithm.

If your acceptance rate is too high, the chain is probably not mixing well (not moving around the parameter space quickly enough).

If your acceptance rate is too low, your algorithm is too inefficient (rejecting too many candidate draws).

What is too high and too low depends on your specific algorithm, but generally

Acceptance Rates

It is important to monitor the *acceptance rate* (the fraction of candidate draws that are accepted) of your Metropolis-Hastings algorithm.

If your acceptance rate is too high, the chain is probably not mixing well (not moving around the parameter space quickly enough).

If your acceptance rate is too low, your algorithm is too inefficient (rejecting too many candidate draws).

What is too high and too low depends on your specific algorithm, but generally

- random walk: somewhere between 0.25 and 0.50 is recommended

Acceptance Rates

It is important to monitor the *acceptance rate* (the fraction of candidate draws that are accepted) of your Metropolis-Hastings algorithm.

If your acceptance rate is too high, the chain is probably not mixing well (not moving around the parameter space quickly enough).

If your acceptance rate is too low, your algorithm is too inefficient (rejecting too many candidate draws).

What is too high and too low depends on your specific algorithm, but generally

- random walk: somewhere between 0.25 and 0.50 is recommended
- independent: something close to 1 is preferred

A Simple Example

A Simple Example

Using a random walk Metropolis algorithm to sample from a Gamma(1.7, 4.4) distribution with a Normal jumping distribution with standard deviation of 2.

A Simple Example

Using a random walk Metropolis algorithm to sample from a Gamma(1.7, 4.4) distribution with a Normal jumping distribution with standard deviation of 2.

```
> mh.gamma <- function(n.sims, start, burnin, cand.sd, shape, rate) {
+     theta.cur <- start
+     draws <- c()
+     theta.update <- function(theta.cur, shape, rate) {
+         theta.can <- rnorm(1, mean = theta.cur, sd = cand.sd)
+         accept.prob <- dgamma(theta.can, shape = shape, rate = rate)/dgamma(theta.cur,
+             shape = shape, rate = rate)
+         if (runif(1) <= accept.prob)
+             theta.can
+         else theta.cur
+     }
+     for (i in 1:n.sims) {
+         draws[i] <- theta.cur <- theta.update(theta.cur, shape = shape,
+             rate = rate)
+     }
+     return(draws[(burnin + 1):n.sims])
+ }
> mh.draws <- mh.gamma(10000, start = 1, burnin = 1000, cand.sd = 2,
+     shape = 1.7, rate = 4.4)
```