Decision and Risk Analysis

Lecture 7: **A Simple Manual Example of MCMC**

Metropolis-Hastings for a Bernoulli Parameter

DAEN 427 / ISEN 427 / ISEN 627 - Fall 2026

Alexander Abuabara



We want to understand how MCMC can generate samples from a posterior distribution.

Suppose we observe four Bernoulli trials:

$$y = (1, 1, 1, 0)$$

That is:

$$3 \text{ successes}, \qquad 1 \text{ failure}$$

The unknown parameter is:

$$\theta = \Pr(y = 1)$$

What can we learn about the unknown success probability θ?

1. Prior: All probabilities between 0 and 1 are initially equally plausible.

$$\theta \sim \text{Beta}(1, 1)$$

2. Data and likelihood:

$$Y \mid \theta \sim \text{Binomial}(4, \theta), \qquad y = 3$$

3. Bayesian updating:

$$\begin{aligned} p(\theta \mid y) &\propto p(y \mid \theta)\, p(\theta) \\ &\propto \theta^3(1-\theta) \cdot 1 \end{aligned}$$

4. Posterior distribution:

$$\boxed{\theta \mid y \sim \text{Beta}(4, 2)}$$

Before observing data, all success probabilities were equally plausible. After observing 3 successes in 4 trials, larger probabilities become more plausible, but uncertainty remains.

Metropolis Rule (Symmetric Proposal)

Start at some current value:

$$\theta_t$$

Propose a new value:

$$\theta^*$$

For a symmetric proposal, $q(\theta^* \mid \theta_t) = q(\theta_t \mid \theta^*)$, compute the acceptance ratio:

$$r = \frac{f(\theta^*)}{f(\theta_t)}$$

(check *why the algorithm works)*

Then compute:

$$\alpha = \min(1, r)$$

Accept the proposed value if:

$$u < \alpha, \qquad u \sim \text{Uniform}(0, 1)$$

Step 1: Move from 0.50 to 0.60

Start at:

$$\theta_0 = 0.50$$

Compute:

$$f(0.50) = 0.50^3(1 - 0.50) = 0.125 \times 0.50 = 0.0625$$

Propose:

$$\theta^* = 0.60$$

Compute:

$$f(0.60) = 0.60^3(1 - 0.60) = 0.216 \times 0.40 = 0.0864$$

Acceptance ratio:

$$r = \frac{0.0864}{0.0625} = 1.3824$$

So:

$$\alpha = 1$$

Accept:

$$0.50 \rightarrow 0.60$$

Step 2: Move from 0.60 to 0.80

Current value:

$$\theta_t = 0.60$$

Propose:

$$\theta^* = 0.80$$

Compute:

$$f(0.80) = 0.80^3(1 - 0.80) = 0.512 \times 0.20 = 0.1024$$

Acceptance ratio:

$$r = \frac{0.1024}{0.0864} = 1.185$$

Thus:

$$\alpha = 1$$

Accept:

$$0.60 \rightarrow 0.80$$

Step 3: A Rejected Proposal

Current value:

$$\theta_t = 0.80$$

Propose:

$$\theta^* = 0.95$$

Compute:

$$f(0.95) = 0.95^3(1 - 0.95) \approx 0.8574 \times 0.05 = 0.0429$$

Acceptance ratio:

$$r = \frac{0.0429}{0.1024} \approx 0.419$$

So:

$$\alpha = 0.419$$

Suppose we draw: *(random value 0-1)*

$$u = 0.70$$

Since:

$$0.70 > 0.419$$

reject the proposal and stay at:

$$\theta = 0.80$$

Step 4: Move from 0.80 to 0.70

Current value:

$$\theta_t = 0.80$$

Propose:

$$\theta^* = 0.70$$

Compute:

$$f(0.70) = 0.70^3(1 - 0.70) = 0.343 \times 0.30 = 0.1029$$

Acceptance ratio:

$$r = \frac{0.1029}{0.1024} \approx 1.005$$

Thus:

$$\alpha = 1$$

Accept:

$$0.80 \to 0.70$$

Manual Chain

The chain we recorded is:

$$0.50,\ 0.60,\ 0.80,\ 0.80,\ 0.70$$

Notice the repeated value:

$$0.80,\ 0.80$$

This happens because a rejected proposal keeps the chain at the current state.

After many iterations, the histogram of the chain approximates the posterior:

$$p(\theta \mid y)$$

Summary Table

Step	Current	Proposal	f(Current)	f(Proposal)	α	Result
1	0.50	0.60	0.0625	0.0864	1.000	Accept
2	0.60	0.80	0.0864	0.1024	1.000	Accept
3	0.80	0.95	0.1024	0.0429	0.419	Reject
4	0.80	0.70	0.1024	0.1029	1.000	Accept

Key Idea

MCMC does not only search for the best value.

It constructs a chain whose long-run behavior matches the posterior distribution.

Regions with higher posterior probability are visited more often.

Regions with lower posterior probability are visited less often, but not necessarily never.

That is why MCMC provides uncertainty, not just a point estimate.

In the simple example, the unknown was one parameter:

$$\theta$$

In Bayesian linear regression, the unknowns are usually:

$$(\beta_0, \beta_1, \sigma)$$

MCMC samples from:

$$p(\beta_0, \beta_1, \sigma \mid x, y)$$

Each draw gives one plausible regression model:

$$y = \beta_0 + \beta_1 x + \varepsilon$$

The collection of draws gives the full posterior uncertainty.

Monte Carlo: Draw Samples Directly

1. Draw many independent values $x_1, \ldots, x_N$ from the distribution we want to study, $\pi(x)$.
2. For each value, calculate the result we care about, $g(x_i)$.
3. Average the results: $\frac{1}{N} \sum_{i=1}^{N} g(x_i)$.

Advantage: Easy to understand; each draw is independent.

Challenge: It may be hard to draw directly from a complicated distribution (e.g., complex shapes, high dimensions).

Main difference: Monte Carlo draws independent values directly; Metropolis–Hastings builds a chain in which each value depends on the previous one.

Metropolis–Hastings: How It Works

1. Start with a current value x_t.
2. Suggest a new value x^* using a **proposal distribution** $q(x^* \mid x_t)$.
3. Calculate how likely we are to accept it:

$$\alpha = \min\left\{1, \frac{\pi(x^*)\, q(x_t \mid x^*)}{\pi(x_t)\, q(x^* \mid x_t)}\right\}.$$

4. Draw u between 0 and 1. If $u < \alpha$, move to x^*; otherwise, stay at x_t.
5. Repeat to create a chain of samples.

$\pi(x)$**:** The distribution we want to sample.
$q(y \mid x)$**:** The chance (or density) of proposing y when at x.

We use a proposal distribution because knowing how likely a value is under a posterior distribution is not the same as knowing how to generate random values from that posterior.

The Metropolis–Hastings acceptance rule includes a **proposal ratio**:

$$\alpha = \min\left\{1, \frac{\pi(x^*)}{\pi(x_t)} \times \underbrace{\frac{q(x_t \mid x^*)}{q(x^* \mid x_t)}}_{\text{proposal ratio}}\right\}.$$

If the proposal is **symmetric**, moving from x_t to x^* is just as likely as proposing the reverse move:

$$q(x_t \mid x^*) = q(x^* \mid x_t).$$

So we divide two equal numbers: the proposal ratio is 1.

$$\boxed{\alpha = \min\left\{1, \frac{\pi(x^*)}{\pi(x_t)}\right\}}$$

Example: If each direction has proposal density 0.2, then $0.2/0.2 = 1$.

This simpler rule is called the **Metropolis algorithm**.

Metropolis vs. Metropolis–Hastings

- **Metropolis:** The proposal is symmetric: it is equally likely to suggest a move in either direction. The q terms cancel, so only the target distribution appears in the acceptance ratio.
- **Metropolis–Hastings:** Also works with *asymmetric* proposals. A move in one direction may be more likely than the reverse, so the q ratio must stay in the formula.

Remember!! Metropolis is a special case of Metropolis–Hastings.

Our earlier numerical example used the simpler Metropolis rule.

Independence Sampling

An **independence sampler** is a type of Metropolis–Hastings algorithm.

- It suggests new values from the **same fixed proposal distribution**, no matter where the chain currently is:

$$q(x^* \mid x_t) = q(x^*).$$

- The acceptance rule becomes

$$\alpha = \min\left\{1, \frac{\pi(x^*)\, q(x_t)}{\pi(x_t)\, q(x^*)}\right\}.$$

- It is **not usually Metropolis**, because the forward and reverse proposal densities may differ.

Important: Even though proposals do not depend on the current state, accepted samples can still depend on earlier samples.

Conclusion

- **The posterior distribution** is defined by Bayes' rule, combining the prior and likelihood.
- **MCMC** generates a sequence of dependent samples whose long-run distribution is the posterior.
- **Metropolis-Hastings** is one MCMC algorithm:
  1. Propose a new value.
  2. Accept or reject it using the posterior.
  3. Repeat many times.
- **Monte Carlo** uses the resulting samples to approximate posterior means, credible intervals, and probabilities.

MCMC does not create the posterior distribution. It generates samples that allow us to **explore and summarize the posterior**.

Homework

1. What are the steps to the Monte Carlo Algorithm?
2. Name one pro and one con for the Monte Carlo Algorithm.
3. What are the steps to the Metropolis-Hastings Algorithm?
4. What's the difference between the Monte Carlo and Metropolis-Hastings algorithms?
5. What's the difference between the Metropolis and Metropolis-Hastings algorithms?
6. Identify whether the below statements about independence sampling are True or False. If False, explain.
   - 6.1 The proposal model depends on the current value of the chain.
   - 6.2 The proposal model is the same every time.
   - 6.3 It is a special case of the Metropolis-Hastings algorithm.
   - 6.4 It is a special case of Metropolis algorithm.