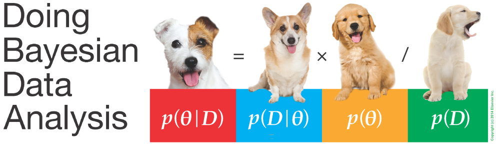
Bayesian Computing for Data Science (DATS 6311)
Markov Chain Monte Carlo

Yuxiao Huang

Data Science, Columbian College of Arts & Sciences
George Washington University
yuxiaohuang@gwu.edu

Summer 2020

Reference



Picture courtesy of the book website

- This set of slices is an excerpt of the book by Professor John K. Kruschke, with some trivial changes by the creator of the slides
- Please find the reference to and website of the book below:
  - Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier
  - `https://sites.google.com/site/doingbayesiandataanalysis/`

Overview

1. Motivation

2. A simple case of the metropolis algorithm

3. The metropolis algorithm more generally

4. Toward gibbs sampling: estimating two coin biases

Motivation

- **Q:** Recall, what methods have been discussed for estimating the parameters?

Motivation

- **Q:** Recall, what methods have been discussed for estimating the parameters?
- **A:**
  - when the posterior is analytically solvable, formal analysis
  - otherwise, grid approximation
- **Q:** Are they sufficient?

Motivation

- **Q:** Recall, what methods have been discussed for estimating the parameters?
- **A:**
    - when the posterior is analytically solvable, formal analysis
    - otherwise, grid approximation
- **Q:** Are they sufficient?
- **A:**
    - suppose there are (just) 6 parameters, for each parameter we use 1000 bins. **Q:** What is the size of the dimension?

Motivation

- **Q:** Recall, what methods have been discussed for estimating the parameters?
- **A:**
    - when the posterior is analytically solvable, formal analysis
    - otherwise, grid approximation
- **Q:** Are they sufficient?
- **A:**
    - suppose there are (just) 6 parameters, for each parameter we use 1000 bins. **Q:** What is the size of the dimension?
    - **A:** 1000^6!
- So, unfortunately, these methods are not sufficient
- This is why we need Markov Chain Monte Carlo (MCMC)
- Actually, MCMC is one of the key reasons that makes Bayesian inference feasible

Approximating a distribution with a large sample

- The idea of MCMC is representing the distribution by samples generated from it
- This idea is shown in Figure 7.1 (see next page)
    - the top left panel shows a beta distribution
    - the other three show the histograms of samples generated from it
- **Q:** What conclusion can you make?

Figure 7.1

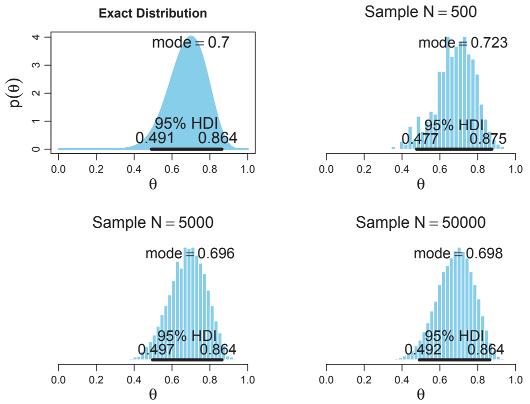


Figure 7.1: Large representative samples approximate the continuous distribution in the upper left panel. The larger the sample, the more accurate the approximation.(This happens to be a beta(θ|15, 7) distribution.) Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Approximating a distribution with a large sample

- The idea of MCMC is representing the posterior distribution by samples generated from it
- This idea is shown in Figure 7.1 (see next page)
    - the top left panel shows a beta distribution
    - the other three show the histograms of samples generated from it
- **Q:** What conclusion can you make?
- **A:** Larger sample size $\rightarrow$ better distribution estimation

Example: a politician stumbles upon the metropolis algorithm

- Suppose a politician lives on a long chain of islands, from east to west
- He has to:
  - stay on the current island
  - move to the adjacent island to the west
  - move to the adjacent island to the east
- He knows:
  - the population of the current island
  - the population of the adjacent islands
- He wants to visit the islands proportionally to their relative population
- **Q:** What would you do if you were the politician?

A simple idea

- **A:** The politician has a simple idea:
  1. he flips a coin to propose going to the east island or west
  2. if the population on the proposed island, P_{proposed}, is larger than that on the current island, P_{current}, he definitely goes to the proposed island
  3. otherwise, he goes there with probability P_{move} such that

$$P_{\text{move}} = \frac{P_{\text{proposed}}}{P_{\text{current}}}$$

- Amazingly, this idea works!

A random walk

- Suppose there are seven islands, indexed by the value θ
- The relative populations of the islands increase linearly such that $P(\theta) = \theta$, as shown in the bottom panel of Figure 7.2 (see next page)
- The middle panel shows one possible trajectory taken by the politician. **Q:** Explain:
  - where did the politician start
  - what happened in the first 5 steps?
  - why does the posterior distribution make sense?

Figure 7.2

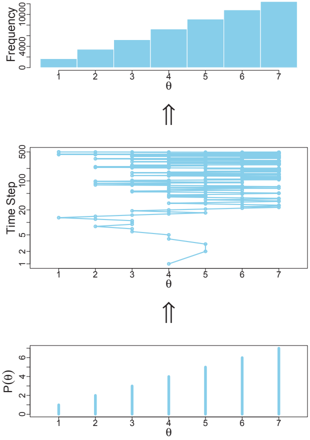


Figure 7.2: Illustration of a simple Metropolis algorithm. The bottom panel shows the values of the target distribution. The middle panel shows one random walk, at each time step proposing to move either one unit right or one unit left, and accepting the proposed move according the heuristic described in the main text. The top panel shows the frequency distribution of the positions in the walk. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

General properties of a random walk

- **Q:** Why is the trajectory in Figure 7.2 just one possible sequence?

General properties of a random walk

- **Q:** Why is the trajectory in Figure 7.2 just one possible sequence?
- **A:** Because of the randomness in the heuristic. **Q:** What are they?

General properties of a random walk

- **Q:** Why is the trajectory in Figure 7.2 just one possible sequence?
- **A:** Because of the randomness in the heuristic. **Q:** What are they?
- **A:**
    - the direction (east or west) of the proposed move is random
    - the acceptance of the proposed move could also be random
- While the trajectories are different, in the long run they all produce the same relative frequency that mimics the target distribution

General properties of a random walk

- Figure 7.3 (see next page) shows the probability of being in each position as a function of time
- In the early time steps, the probability distribution has a bulge over the starting position
- By time $t = 99$, the probability distribution is virtually indistinguishable from the target distribution
- **Q:** How is Figure 7.3 obtained?

Figure 7.3

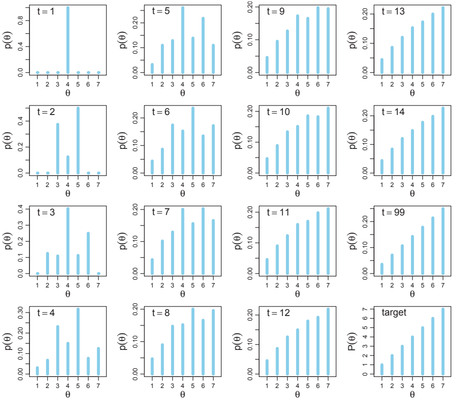


Figure 7.3: The probability of being at position θ, as a function of time t, when a simple Metropolis algorithm is applied to the target distribution in the lower right panel. The time in each panel corresponds to the step in a random walk, an example of which is shown in Figure 7.2. The target distribution is shown in the lower right panel. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Why we care

- **Q:** What assumptions does the random-walk process rely on?

Why we care

- **Q:** What assumptions does the random-walk process rely on?
- **A:**
    - we must be able to generate a random value from the proposal distribution, to create θ_{proposed}
    - we must be able to evaluate the target distribution at any proposed position, to compute

$$P_{\text{move}} = \frac{P(\theta_{\text{proposed}})}{P(\theta_{\text{current}})}$$

    - we must be able to generate a random value from a uniform distribution, to accept or reject the proposal according to P_{move}
- By being able to do those three things, we are able to do indirectly something we could not necessarily do directly: we can generate random samples from the target distribution

Why we care

- This technique is profoundly useful when the target distribution is a posterior distribution, $p(\theta|D)$
- **Q:** Recall, what is difficult when calculating posterior distribution?

Why we care

- This technique is profoundly useful when the target distribution is a posterior distribution, $p(\theta|D)$
- **Q:** Recall, what is difficult when calculating posterior distribution?
- **A:** Calculating the evidence, $p(D)$, because it needs the integral:

$$p(D) = \int_{\theta} p(D|\theta)p(\theta)$$

- **Q:** Do we need to calculate $p(D)$ in the random-walk?

Why we care

- This technique is profoundly useful when the target distribution is a posterior distribution, $p(\theta|D)$
- **Q:** Recall, what is difficult when calculating posterior distribution?
- **A:** Calculating the evidence, $p(D)$, because it needs the integral:

$$p(D) = \int_\theta p(D|\theta)p(\theta)$$

- **Q:** Do we need to calculate $p(D)$ in the random-walk?
- **A:** No, since all we care about is

$$\frac{p(\theta_{\text{propose}}|D)}{p(\theta_{\text{current}}|D)},$$

and $p(D)$ is canceled out:

$$\frac{p(\theta_{\text{propose}}|D)}{p(\theta_{\text{current}}|D)} = \frac{\frac{p(D|\theta_{\text{propose}})p(\theta_{\text{propose}})}{\cancel{p(D)}}}{\frac{p(D|\theta_{\text{current}})p(\theta_{\text{current}})}{\cancel{p(D)}}}$$

Why it works

- We now prove that in the long run, each position will be visited proportionally to its target value
- **Q:** What is the probability of moving from θ to $\theta + 1$, $p(\theta \rightarrow \theta + 1)$?

Why it works

- We now prove that in the long run, each position will be visited proportionally to its target value
- **Q:** What is the probability of moving from θ to $\theta + 1$, $p(\theta \rightarrow \theta + 1)$?
- **A:**

$$p(\theta \rightarrow \theta + 1) = 0.5 \cdot \min\left(\frac{P(\theta + 1)}{P(\theta)}, 1\right)$$

- **Q:** What is the probability of moving from $\theta + 1$ to θ, $p(\theta + 1 \rightarrow \theta)$?

Why it works

- We now prove that in the long run, each position will be visited proportionally to its target value
- **Q:** What is the probability of moving from θ to $\theta + 1$, $p(\theta \rightarrow \theta + 1)$?
- **A:**

$$p(\theta \rightarrow \theta + 1) = 0.5 \cdot \min\left(\frac{P(\theta + 1)}{P(\theta)}, 1\right)$$

- **Q:** What is the probability of moving from $\theta + 1$ to θ, $p(\theta + 1 \rightarrow \theta)$?
- **A:**

$$p(\theta + 1 \rightarrow \theta) = 0.5 \cdot \min\left(\frac{P(\theta)}{P(\theta + 1)}, 1\right)$$

Why it works

- **Q:** What is the ratio of the transition probabilities?

Why it works

- **Q:** What is the ratio of the transition probabilities?
- **A:**

$$\begin{aligned}\frac{p(\theta \rightarrow \theta+1)}{p(\theta+1 \rightarrow \theta)} &= \frac{0.5 \cdot \min\left(\frac{P(\theta+1)}{P(\theta)}, 1\right)}{0.5 \cdot \min\left(\frac{P(\theta)}{P(\theta+1)}, 1\right)} \\ &= \begin{cases} \frac{1}{P(\theta)/P(\theta+1)} & \text{if } P(\theta+1) \geq P(\theta) \\ \frac{P(\theta+1)/P(\theta)}{1} & \text{if } P(\theta+1) < P(\theta) \end{cases} \\ &= \frac{P(\theta+1)}{P(\theta)}\end{aligned}$$

- This means that the relative probability of the transitions exactly matches the relative values of the target distribution
- This is why the frequency exactly matches the target distribution in Figure 7.2 (see next page)

Figure 7.2

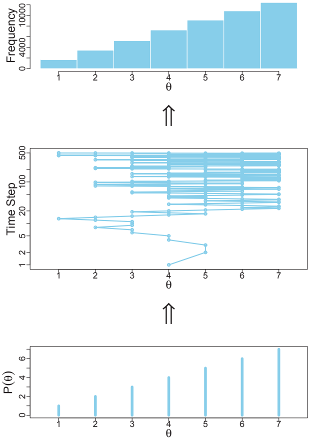


Figure 7.2: Illustration of a simple Metropolis algorithm. The bottom panel shows the values of the target distribution. The middle panel shows one random walk, at each time step proposing to move either one unit right or one unit left, and accepting the proposed move according the heuristic described in the main text. The top panel shows the frequency distribution of the positions in the walk. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Why it works

- We can put those transition probabilities into a matrix
- Each row of the matrix is a possible current position, and each column is a candidate moved-to position
- Below is a submatrix from the full transition matrix T, showing rows $\theta - 2$ to $\theta + 2$, and columns $\theta - 2$ to $\theta + 2$

$$\begin{bmatrix} \ddots & p(\theta-2 \to \theta-1) & 0 & 0 & 0 \\ \ddots & p(\theta-1 \to \theta-1) & p(\theta-1 \to \theta) & 0 & 0 \\ 0 & p(\theta \to \theta-1) & p(\theta \to \theta) & p(\theta \to \theta+1) & 0 \\ 0 & 0 & p(\theta+1 \to \theta) & p(\theta+1 \to \theta+1) & \ddots \\ 0 & 0 & 0 & p(\theta+2 \to \theta+1) & \ddots \end{bmatrix}$$

Why it works

- The previous matrix can be rewritten as

$$\begin{bmatrix}
\ddots & 0.5\min\left(\frac{p(\theta-1)}{p(\theta-2)},1\right) & 0 & 0 & 0 \\
\ddots & \begin{array}{c} 0.5\left[1-\min\left(\frac{p(\theta-2)}{p(\theta-1)},1\right)\right] \\ +0.5\left[1-\min\left(\frac{p(\theta)}{p(\theta-1)},1\right)\right] \end{array} & 0.5\min\left(\frac{p(\theta)}{p(\theta-1)},1\right) & 0 & 0 \\
0 & 0.5\min\left(\frac{p(\theta-1)}{p(\theta)},1\right) & \begin{array}{c} 0.5\left[1-\min\left(\frac{p(\theta-1)}{p(\theta)},1\right)\right] \\ +0.5\left[1-\min\left(\frac{p(\theta+1)}{p(\theta)},1\right)\right] \end{array} & 0.5\min\left(\frac{p(\theta+1)}{p(\theta)},1\right) & 0 \\
0 & 0 & 0.5\min\left(\frac{p(\theta)}{p(\theta+1)},1\right) & \begin{array}{c} 0.5\left[1-\min\left(\frac{p(\theta)}{p(\theta+1)},1\right)\right] \\ +0.5\left[1-\min\left(\frac{p(\theta+2)}{p(\theta+1)},1\right)\right] \end{array} & \ddots \\
0 & 0 & 0 & 0.5\min\left(\frac{p(\theta+1)}{p(\theta+2)},1\right) & \ddots
\end{bmatrix}$$

- **Q:** Why these values?

Why it works

- With current position and transition matrix, we can use matrix multiplication to get the probabilities of each position in the next step
  - let $W = [\ldots, 0, 1, 0, \ldots]$ be the probabilities of each position θ being the initial position, T the transition matrix
  - then the probabilities of each position in the next step is WT
- **Q:** Again, how is Figure 7.3 (see next page) obtained?

Figure 7.3

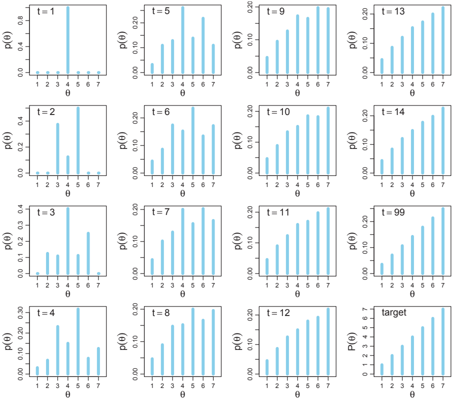


Figure 7.3: The probability of being at position θ, as a function of time t, when a simple Metropolis algorithm is applied to the target distribution in the lower right panel. The time in each panel corresponds to the step in a random walk, an example of which is shown in Figure 7.2. The target distribution is shown in the lower right panel. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Stationary Distribution

- When the vector of position probabilities is the target distribution, it stays that way on the next time step
    - suppose the current position probabilities are the target probabilities:

$$W = \frac{[\ldots, P(\theta - 1), P(\theta), P(\theta + 1), \ldots]}{\sum_{\theta} P(\theta)}$$

then we can prove that

$$WT = W$$

- We call such W the stationary distribution of a Markov Chain with transition matrix T

The metropolis algorithm more generally

- In the previous island-hopping example, we considered:
    - discrete positions
    - only one dimension
    - moves that proposed just one position (west or east)
- It is a special case of a more general procedure known as the Metropolis algorithm, where we consider:
    - continuous values
    - any number of dimensions
    - moves that proposed by more general proposal distributions

The keys in metropolis algorithm

- Target distribution such that:
    - it is over a multidimensional continuous parameter space
    - it is usually the unnormalized posterior distribution on θ (i.e., the product of the likelihood and prior)
- Proposal distribution such that:
    - it can efficiently explore the parameter space
    - we can efficiently generate samples from the distribution
- Transition probability such that:

$$P_{\text{move}} = \min\left(1, \frac{P_{\text{proposed}}}{P_{\text{current}}}\right)$$

Metropolis algorithm applied to bernoulli likelihood and beta prior

- Let us again revisit the coin-flipping example, where we:
  - flip a coin N times and observe z heads
  - use a bernoulli likelihood

$$p(z, N|\theta) = \theta^z (1 - \theta)^{N-z}$$

  - use a beta prior

$$p(\theta) = \mathrm{beta}(\theta|a, b)$$

- We apply the metropolis algorithm to this example, where we:
  - use a normal distribution, $N(0, \sigma)$, as the proposed distribution
  - denote the proposed jump as

$$\Delta\theta \sim N(0, \sigma)$$

The metropolis algorithm in the coin-flipping example

- Repeat until sufficiently representative samples have been generated:
  1. randomly generate a proposed jump, $\Delta\theta \sim N(0, \sigma)$, and denote the proposed value of the parameter as
  $$\theta_{\text{pro}} = \theta_{\text{cur}} + \Delta\theta$$
  2. compute the probability of moving to the proposed value
  $$\begin{aligned} p_{\text{move}} &= \min\left(1, \frac{p(\theta_{\text{pro}}|D)}{p(\theta_{\text{cur}}|D)}\right) = \min\left(1, \frac{p(D|\theta_{\text{pro}})p(\theta_{\text{pro}})}{p(D|\theta_{\text{cur}})p(\theta_{\text{cur}})}\right) \\ &= \min\left(1, \frac{\text{bern}(z, N|\theta_{\text{pro}})\text{beta}(\theta_{\text{pro}}|a, b)}{\text{bern}(z, N|\theta_{\text{cur}})\text{beta}(\theta_{\text{cur}}|a, b)}\right) \\ &= \min\left(1, \frac{\theta_{\text{pro}}^{z}(1-\theta_{\text{pro}})^{(N-z)}\theta_{\text{pro}}^{(a-1)}(1-\theta_{\text{pro}})^{(b-1)}/B(a, b)}{\theta_{\text{cur}}^{z}(1-\theta_{\text{cur}})^{(N-z)}\theta_{\text{cur}}^{(a-1)}(1-\theta_{\text{cur}})^{(b-1)}/B(a, b)}\right) \end{aligned}$$
  3. accept θ_{pro} if a random value sampled from a $[0, 1]$ uniform distribution is less than p_{move}, otherwise reject θ_{pro} and tally θ_{cur} again

The metropolis algorithm in the coin-flipping example

- Figure 7.4 (see next page) shows examples of the metropolis algorithm applied to a case with
  - beta prior: beta$(\theta|1, 1)$
  - bernoulli likelihood: bern$(14, 20|\theta)$
- There are three columns in the figure, for three runs of the metropolis algorithm, using three values for σ (i.e., standard deviation) in the proposal distribution
- **Q:** What are the differences between the columns?

Figure 7.4

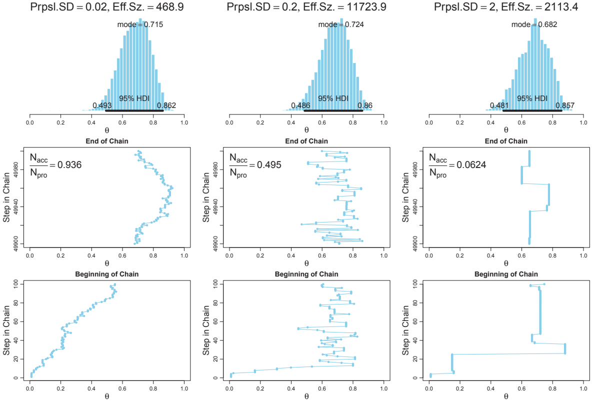


Figure 7.4: Metropolis algorithm applied to Bernoulli likelihood with beta(θ|1, 1) prior and z = 14 with N = 20. For each of the three columns, there are 50,000 steps in the chain, but, for the left column the proposal standard deviation (SD) is 0.02, for the middle column SD=0.2, and for the right column SD=2.0. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

The metropolis algorithm in the coin-flipping example

- Figure 7.4 shows examples of the metropolis algorithm applied to a case with
  - beta prior: beta$(\theta|1, 1)$
  - bernoulli likelihood: bern$(14, 20|\theta)$
- There are three columns in the figure, for three runs of the metropolis algorithm, using three values for σ in the proposal distribution
- **Q:** What are the differences among the columns?
- **A:**
  - left: small σ, small proposed jump, large acceptance ratio, small effective size (e.g., number of representative values)
  - right: large σ, large proposed jump, small acceptance ratio, small effective size
  - middle: medium σ, medium proposed jump, medium acceptance ratio, large effective size

Summary of metropolis algorithm

- Metropolis algorithm provides a high-resolution picture of the posterior distribution, by generating representative parameter values
- Particularly, representative parameter values can be sampled from complicated posterior distributions without solving the integral in Bayes' rule

Prior, likelihood and posterior for two biases

- Suppose we have two coins and our goal is to estimate their biases, θ_1 and θ_2
- Our prior beliefs are the probabilities of the combinations of parameter values, $p(\theta_1, \theta_2)$, such that

$$\iint \mathrm{d}\theta_1 \mathrm{d}\theta_2 \; p(\theta_1, \theta_2) = 1$$

- We assume that our beliefs about θ_1 and θ_2 are independent:

$$p(\theta_1, \theta_2) = p(\theta_1) p(\theta_2),$$

where $p(\theta_1)$ and $p(\theta_2)$ are prior distributions of θ_1 and θ_2

Prior, likelihood and posterior for two biases

- We also assume the independence of the data:
  - the data from coin 1, y_1, depend only on the bias in coin 1, θ_1
  - the data from coin 2, y_2, depend only on the bias in coin 2, θ_2

$$p(y_1|\theta_1, \theta_2) = p(y_1|\theta_1) \quad \text{and} \quad p(y_2|\theta_1, \theta_2) = p(y_2|\theta_2)$$

- We observe some data of the two coins:
  - from coin 1 we observe data D_1 consisting of z_1 heads in N_1 flips
  - from coin 2 we observe data D_2 consisting of z_2 heads in N_2 flips

Prior, likelihood and posterior for two biases

- Because of independence of sampled flips,

$$\begin{aligned} p(D|\theta_1, \theta_2) &= \prod_{y_{1i} \in D_1} p(y_{1i}|\theta_1, \theta_2) \prod_{y_{2i} \in D_2} p(y_{2i}|\theta_1, \theta_2) \\ &= \prod_{y_{1i} \in D_1} \theta_1^{y_{1i}} (1-\theta_1)^{1-y_{1i}} \prod_{y_{2i} \in D_2} \theta_2^{y_{2i}} (1-\theta_2)^{1-y_{2i}} \\ &= \theta_1^{z_1} (1-\theta_1)^{N_1 - z_1} \theta_2^{z_2} (1-\theta_2)^{N_2 - z_2} \end{aligned}$$

- Based on Bayes' rule, the posterior distribution is

$$\begin{aligned} p(\theta_1, \theta_2|D) &= \frac{p(D_1|\theta_1, \theta_2) p(\theta_1, \theta_2)}{p(D)} \\ &= \frac{p(D_1|\theta_1, \theta_2) p(\theta_1, \theta_2)}{\iint \mathrm{d}\theta_1 \mathrm{d}\theta_2 \; p(D|\theta_1, \theta_2) p(\theta_1, \theta_2)} \end{aligned}$$

The posterior via exact formal analysis

- We assume a beta$(\theta_1|a_1, b_1)$ prior on θ_1, and an independent beta$(\theta_2|a_2, b_2)$ prior on θ_2, so that

$$p(\theta_1, \theta_2) = \mathrm{beta}(\theta_1|a_1, b_1) \cdot \mathrm{beta}(\theta_2|a_2, b_2)$$

- Then the posterior distribution, $p(\theta_1, \theta_2|D)$, is

$$\begin{aligned}
&p(\theta_1, \theta_2|D) \\
&= \frac{p(D|\theta_1, \theta_2)p(\theta_1, \theta_2)}{p(D)} \\
&= \frac{\theta_1^{z_1}(1-\theta_1)^{(N_1-z_1)}\theta_2^{z_2}(1-\theta_2)^{(N_2-z_2)}p(\theta_1, \theta_2)}{p(D)} \\
&= \frac{\theta_1^{z_1}(1-\theta_1)^{(N_1-z_1)}\theta_2^{z_2}(1-\theta_2)^{(N_2-z_2)}\theta_1^{(a_1-1)}(1-\theta_1)^{(b_1-1)}\theta_2^{(a_2-1)}(1-\theta_2)^{(b_2-1)}}{p(D)B(a_1, b_1)B(a_2, b_2)} \\
&= \frac{\theta_1^{(z_1+a_1-1)}(1-\theta_1)^{(N_1-z_1+b_1-1)}\theta_2^{(z_2+a_2-1)}(1-\theta_2)^{(N_2-z_2+b_2-1)}}{p(D)B(a_1, b_1)B(a_2, b_2)}
\end{aligned}$$

- **Q:** What is the denominator of the equation?

The posterior via exact formal analysis

- We know that the left side of the previous equation must be a probability density function
- The numerator of the right side has the form of a product of beta distributions:

$$\mathrm{beta}(\theta_1|z_1+a_1, N_1-z_1+b_1)\cdot\mathrm{beta}(\theta_2|z_2+a_2, N_2-z_2+b_2)$$

- Therefore, the denominator of the equation must be the corresponding normalizer for the product of beta distributions:

$$p(D)B(a_1,b_1)B(a_2,b_2) = B(z_1{+}a_1, N_1{-}z_1{+}b_1)B(z_2{+}a_2, N_2{-}z_2{+}b_2)$$

Summary

- If:
  - the prior is a product of independent beta distributions
  - and the likelihood is a product of independent bernoulli distributions
  - then the posterior is a product of independent beta distributions
- In the two coins example, when:
  - the prior is

$$\text{beta}(\theta_1 | a_1, b_1) \cdot \text{beta}(\theta_2 | a_2, b_2)$$

  - and the likelihood is

$$\text{bern}(z_1, N_1 | \theta_1) \cdot \text{bern}(z_2, N_2 | \theta_2)$$

  - then the posterior is

$$\text{beta}(\theta_1 | z_1 + a_1, N_1 - z_1 + b_1) \cdot \text{beta}(\theta_2 | z_2 + a_2, N_2 - z_2 + b_2)$$

The posterior via exact formal analysis

- The two coins example using formal analysis is shown in Figure 7.5 (see next page)
- The left column shows the perspective plot of the distributions, while the right shows the contour plot
- If the goal is a quick intuition about the general layout of the distribution, then a perspective plot is preferred over a contour plot
- If the goal is instead a more detailed visual determination of the parameter values for a particular peak in the distribution, then a contour plot may be preferred

Figure 7.5

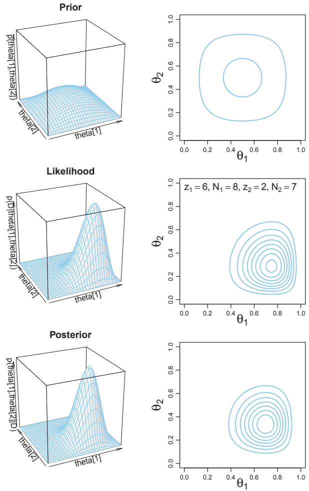


Figure 7.5: Bayesian updating of independent beta($\theta|2, 2$) priors with the data annotated in the middle-right panel. Left panels show perspective surface plots; right panels show contour plots of the same distributions. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

The posterior via the metropolis algorithm

- **Q:** Recall, besides the prior and likelihood, what else do we need to apply the metropolis algorithm to the two-coins example?

The posterior via the metropolis algorithm

- **Q:** Recall, besides the prior and likelihood, what else do we need to apply the metropolis algorithm to the two-coins example?
- **A:** We need a proposal distribution from which we can generate the samples of θ_1 and θ_2
- In this example, we use a bivariate normal with the current position as the central tendency
- **Q:** What does this mean to the proposed jumps?

The posterior via the metropolis algorithm

- **Q:** Recall, besides the prior and likelihood, what else do we need to apply the metropolis algorithm to the two-coins example?
- **A:** We need a proposal distribution from which we can generate the samples of θ_1 and θ_2
- In this example, we use a bivariate normal with the current position as the central tendency
- **Q:** What does this mean to the proposed jumps?
- **A:** The proposed jumps will usually be near the current position

The posterior via the metropolis algorithm

- Figure 7.6 (see next page) shows the metropolis algorithm applied to this example
- The left panel shows results from a proposal distribution with relatively narrow width
  - the acceptance rate is relatively high
  - the effective size is relatively small
- The right panel shows results from a proposal distribution with moderate width
  - the acceptance rate is relatively low
  - the effective size is relatively large

Figure 7.6

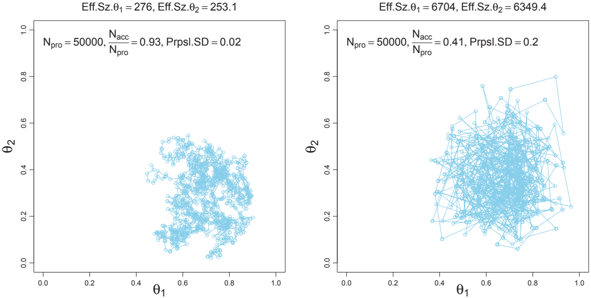


Figure 7.6: Metropolis algorithm applied to the prior and likelihood shown in Figure 7.5, p. 163. Left panel shows chain with narrow proposal distribution and right panel shows chain with moderate-width proposal distribution, as indicated by annotation "Prpsl.SD" in each panel. N_{pro} is the number of proposed jumps, and N_{acc} is the number of accepted proposals. *Many of the plotted points have multiple superimposed symbols where the chain lingered during rejected proposals.* Notice that the effective size of the chain, indicated at the top of the plot, is far less than the length of the chain (N_{pro}). Only 1,000 of the N_{pro} steps are displayed here. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Gibbs sampling

- **Q:** What is the problem of the metropolis algorithm?

Gibbs sampling

- **Q:** What is the problem of the metropolis algorithm?
- **A:**
  - the proposal distribution must be properly tuned
  - if the proposal distribution is too narrow or too broad, the effective size of the chain is far less than the number of proposed jumps
- This is why we need Gibbs sampling, which is usually more efficient than the metropolis algorithm

Gibbs sampling

- Repeat until sufficiently representative samples have been generated:
    - one of the parameters is selected in order

$$\theta_1, \theta_2, \theta_3, \ldots, \theta_1, \theta_2, \theta_3, \ldots$$

    - suppose that parameter θ_i has been selected
    - Gibbs sampling then chooses a new value for that parameter by generating a random value directly from the conditional probability distribution

$$p(\theta_i | \{\theta_{j \neq i}\}, D)$$

    - the new value for θ_i, combined with the unchanged values of $\theta_{j \neq i}$, constitutes the new position in the random walk

Gibbs sampling applied to the two coins example

- Repeat until sufficiently representative samples have been generated:
  - select θ_1 based on the conditional probability distribution

$$
\begin{aligned}
p(\theta_1|\theta_2, D) &= \frac{p(\theta_1, \theta_2|D)}{p(\theta_2|D)} \\
&= \frac{p(\theta_1, \theta_2|D)}{\int \mathrm{d}\theta_1\ p(\theta_1, \theta_2|D)} \\
&= \frac{\mathrm{beta}(\theta_1|z_1 + a_1, N_1 - z_1 + b_1)\mathrm{beta}(\theta_2|z_2 + a_2, N_2 - z_2 + b_2)}{\int \mathrm{d}\theta_1\ \mathrm{beta}(\theta_1|z_1 + a_1, N_1 - z_1 + b_1)\mathrm{beta}(\theta_2|z_2 + a_2, N_2 - z_2 + b_2)} \\
&= \frac{\mathrm{beta}(\theta_1|z_1 + a_1, N_1 - z_1 + b_1)\,\cancel{\mathrm{beta}(\theta_2|z_2 + a_2, N_2 - z_2 + b_2)}}{\cancel{\mathrm{beta}(\theta_2|z_2 + a_2, N_2 - z_2 + b_2)}\ \int \mathrm{d}\theta_1\ \mathrm{beta}(\theta_1|z_1 + a_1, N_1 - z_1 + b_1)} \\
&= \frac{\mathrm{beta}(\theta_1|z_1 + a_1, N_1 - z_1 + b_1)\mathrm{beta}}{\int \mathrm{d}\theta_1\ \mathrm{beta}(\theta_1|z_1 + a_1, N_1 - z_1 + b_1)} \\
&= \mathrm{beta}(\theta_1|z_1 + a_1, N_1 - z_1 + b_1)
\end{aligned}
$$

  - select θ_2 based on the conditional probability distribution and new θ_1

$$
p(\theta_2|\theta_1, D) = \mathrm{beta}(\theta_2|z_2 + a_2, N_2 - z_2 + b_2)
$$

Gibbs sampling applied to the two coins example

- Figure 7.7 (see next page) illustrates such conditional probabilities
- The upper panel shows a slice conditional on θ_2, and the heavy curve illustrates $p(\theta_1|\theta_2, D)$, which is

$$\mathrm{beta}(\theta_1|z_1 + a_1, N_1 - z_1 + b_1)$$

- The lower panel shows a slice conditional on θ_1, and the heavy curve illustrates $p(\theta_2|\theta_1, D)$, which is

$$\mathrm{beta}(\theta_2|z_2 + a_2, N_2 - z_2 + b_2)$$

Figure 7.7

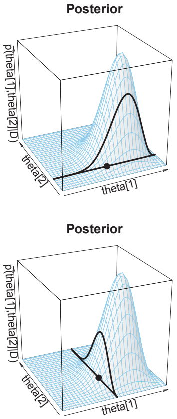


Figure 7.7: Two steps in a Gibbs sampling. In the upper panel, the heavy lines show a slice through the posterior conditionalized on a particular value of θ_2, and the large dot shows a random value of θ_1 sampled from the conditional density. The lower panel shows a random generation of a value for θ_2, conditional on the value for θ_1 determined by the previous step. The heavy lines show a slice through the posterior at the conditional value of θ_1, and the large dot shows the random value of θ_2 sampled from the conditional density. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Gibbs sampling applied to the two coins example

- Figure 7.8 (see next page) shows the result of applying Gibbs sampling to this scenario
- **A:** Why each step in the random walk (in the left panel) is parallel to a parameter axis?

Figure 7.8

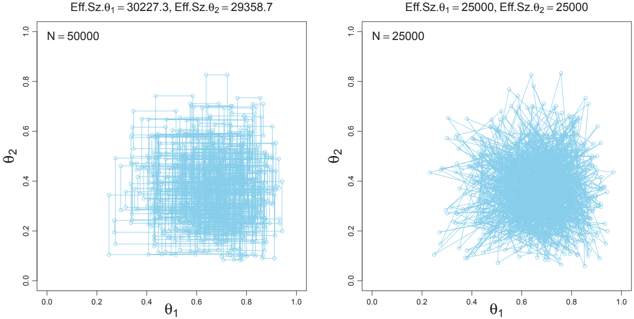


Figure 7.8: Gibbs sampling applied to the posterior shown in Figure 7.5, p. 163. The left panel shows all the intermediate steps of chain, changing one parameter at a time. The right panel shows only the points after complete sweeps through all (two) parameters. Both are valid samples from the posterior distribution. Only 1,000 of the *N* steps are displayed here. Compare with the results of the Metropolis algorithm in Figure 7.6. Notice that the effective size of the Gibbs sample is larger than the effective size of the Metropolis sample for the same length of chain. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Gibbs sampling applied to the two coins example

- Figure 7.8 (see next page) shows the result of applying Gibbs sampling to this scenario
- **Q:** Why each step in the random walk (in the left panel) is parallel to a parameter axis?

Gibbs sampling applied to the two coins example

- Figure 7.8 (see next page) shows the result of applying Gibbs sampling to this scenario
- **Q:** Why each step in the random walk (in the left panel) is parallel to a parameter axis?
- **A:**
    - each step changes only one component parameter
    - the walk cycled through the component parameters
- Notice that the effective size of the Gibbs sampler is much larger than that of the metropolis algorithm shown in Figure 7.6 (see next page)

Figure 7.6

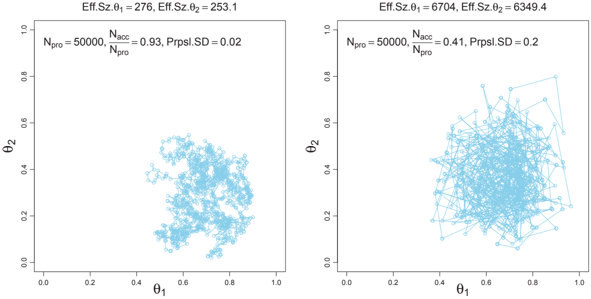


Figure 7.6: Metropolis algorithm applied to the prior and likelihood shown in Figure 7.5, p. 163. Left panel shows chain with narrow proposal distribution and right panel shows chain with moderate-width proposal distribution, as indicated by annotation “Prpsl.SD” in each panel. N_{pro} is the number of proposed jumps, and N_{acc} is the number of accepted proposals. *Many of the plotted points have multiple superimposed symbols where the chain lingered during rejected proposals.* Notice that the effective size of the chain, indicated at the top of the plot, is far less than the length of the chain (N_{pro}). Only 1,000 of the N_{pro} steps are displayed here. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Pros and cons of Gibbs sampling

- Difference between metropolis algorithm and Gibbs sampling
  - in the metropolis algorithm, all the parameters are selected in each step
  - in Gibbs sampling, only one of the parameters is selected in each step
- Pros:
  - Gibbs sampling does not need to tune a proposal distribution
  - Gibbs sampling does not need to reject proposed values
  - thus, the effective size of Gibbs sampling is usually much larger than that of the metropolis algorithm
- Cons:
  - Gibbs sampling needs to know the conditional probabilities
  - Gibbs sampling requires that samples can be generated from such conditional probabilities
  - Gibbs sampling can be stalled by highly correlated parameters (for posterior distributions with shape as a narrow ridge along the diagonal of the parameter space)

Is there a difference between biases?

- Can we obtain the posterior distribution of $\theta_1 - \theta_2$?
  - no matter the metropolis algorithm or Gibbs sampling, in each step we have a combination of parameters, (θ_1, θ_2), where each item is generated by its posterior distribution
  - thus, in each step we have a sample of $\theta_1 - \theta_2$
  - thus, we obtain the distribution of $\theta_1 - \theta_2$
- Figure 7.9 (see next page) shows the histograms of $\theta_1 - \theta_2$ from the posterior distribution
- Upper panels come from results of the metropolis algorithm
- Lower panels come from results of the Gibbs sampling
- A difference of zero is clearly among the 95% most credible differences (i.e., within the 95% HDI), and we would not want to declare that there is a difference

Figure 7.9

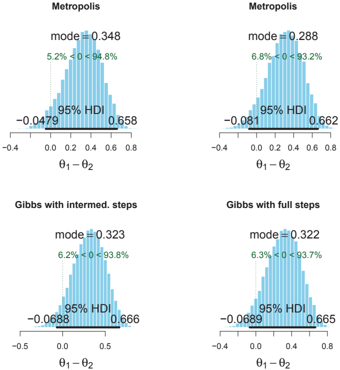


Figure 7.9: Credible posterior differences between biases. Upper panels comes from results of Metropolis algorithm in Figure 7.6; lower panels come from results of Gibbs sampling in Figure 7.8. The four distributions are nearly the same, and in the limit for infinitely long chains, should be identical. For these finite chains, the ones with longer effective size (i.e., the Gibbs sampled) are more accurate on average. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

MCMC representativeness, accuracy, and efficiency

- We have three main goals in generating an MCMC sample from the posterior distribution:
    - the values in the chain must be representative of the posterior distribution
        - they should not be unduly influenced by the arbitrary initial value of the chain
        - they should fully explore the range of the posterior distribution without getting stuck.
    - the chain should be of sufficient size so that estimates are accurate and stable
        - the estimates of the central tendency (such as median or mode), and the limits of the 95% HDI, should not be much different if the MCMC analysis is run again (using different seed states for the pseudorandom number generators)
    - the chain should be generated efficiently, with as few steps as possible

MCMC representativeness

- Checks for unrepresentativeness usually look for:
  - lingering influence of the initial value
  - orphaned chains that have somehow settled into unusual regions of parameter space
- Current practice often focuses on two methods:
  - visual examination of the trajectory
  - consideration of a numerical description of convergence

Visual examination

- A graph of the sampled parameter values as a function of step in the chain is called a trace plot
- One way to enhance the visibility of unrepresentative parts of the chain is to superimpose two or more chains (that have been sampled with independent pseudo-random numbers)
- If the chains are representative, then they should overlap and mix well
- Is the converse true?

Visual examination

- A graph of the sampled parameter values as a function of step in the chain is called a trace plot
- One way to enhance the visibility of unrepresentative parts of the chain is to superimpose two or more chains (that have been sampled with independent pseudo-random numbers)
- If the chains are representative, then they should overlap and mix well
- Is the converse true?
    - No. The chains might overlap, but all be stuck in the same unrepresentative part of parameter space

Visual examination

- Figure 7.10 (see next page) shows the early steps of three MCMC trajectories started at different initial values
- The upper-left panel shows the trace plot
- What can you see from the panel?

Figure 7.10

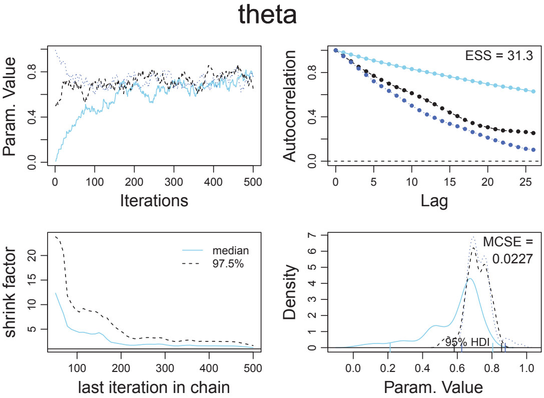


Figure 7.10: Illustration of MCMC diagnostics. Three chains were generated by starting a Metropolis algorithm at different initial values, with proposal SD=0.02 (cf. Fig. 7.4) for data $z = 35$, $N = 50$. Only steps 1–500 are shown here. See Fig. 7.11 for later steps in the chain. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Visual examination

- Figure 7.10 shows the early steps of three MCMC trajectories started at different initial values
- The upper-left panel shows the trace plot
- What can you see from the panel?
    - it takes a few hundred steps for the three chains to converge to the same region of the parameter
    - this suggests that the first several hundred steps of the chain should be excluded from the sample because they are not representative
    - the preliminary steps, during which the chain moves from its unrepresentative initial value to the modal region of the posterior, is called the burn-in period

Visual examination

- Later steps in the chains are shown in Figure 7.11 (see next page)
- The upper-left panel shows the trace plot, where it can be seen that the three chains meander fairly smoothly and overlap each other
    - if any chain were isolated from the others, it would be a sign that convergence had not been achieved
    - if any chain lingered for extended durations at (nearly) the same value, or changed values only very gradually, it might also be a sign of failure to converge
- While the chains converge, they do meander relatively slowly, which is a sign of inefficiency

Figure 7.11

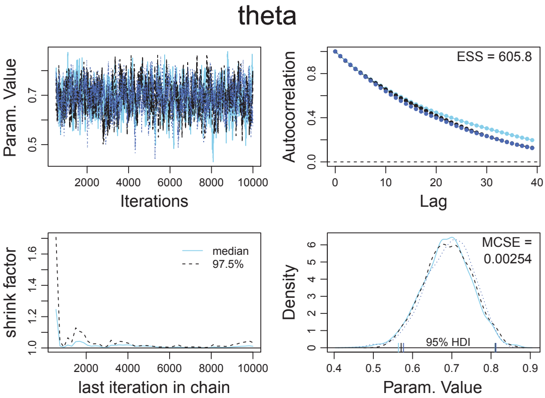


Figure 7.11: Illustration of MCMC diagnostics. Three chains were generated by starting a Metropolis algorithm at different initial values, with proposal SD=0.02 (cf. Fig. 7.4) for data $z = 35$, $N = 50$. Steps 500–10,000 are shown here. See Fig. 7.10 for earlier steps in the chain. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Visual examination

- Another useful visual representation appears in the lower-right panels of Figures 7.10 and 7.11
- These plots show smoothed histograms (a.k.a., density plots) of the parameter values sampled in each chain
- Unlike histograms, which show the exact proportion of points in each histogram bin, density plots average across overlapping intervals to produce a smooth representation of probability density
- As shown in the lower-right panel of the two figures:
  - the density plots of the three chains do not overlap very well during the burn-in period
  - the density plots of the three chains do overlap well after the burn-in period
  - this suggests, but does not guarantee, that the chains are producing representative values from the posterior distribution

Figure 7.10

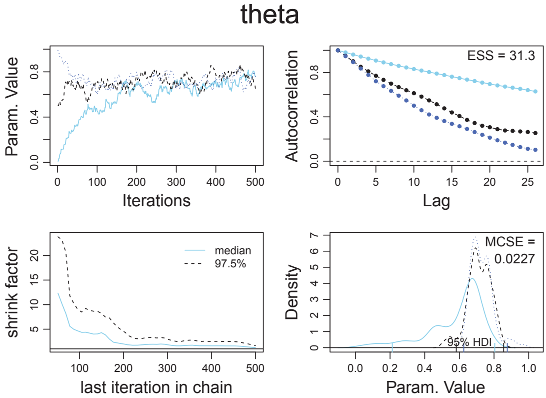


Figure 7.10: Illustration of MCMC diagnostics. Three chains were generated by starting a Metropolis algorithm at different initial values, with proposal SD=0.02 (cf. Fig. 7.4) for data $z = 35$, $N = 50$. Only steps 1–500 are shown here. See Fig. 7.11 for later steps in the chain. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Shrink factor

- Besides the visual checks of convergence, one popular numerical check is a measure of how much variance there is between chains relative to how much variance there is within chains
- The idea is that, if all the chains have settled into a representative sampling, then the average difference between the chains should be the same as the average difference (across steps) within the chains
- But, if one or more chains is orphaned or stuck, it will increase the between-chain variance relative to the within-chain variance

Shrink factor

- As shown in the lower-left panel of Figures 7.10 and 7.11 (see next two pages):
  - during the burn-in period, the measure greatly exceeds 1.0
  - after the burn-in period, the measure quickly gets very close to 1.0
- The specific numerical measure is called the Gelman-Rubin statistic, or the Brooks-Gelman-Rubin statistic, or the "potential scale reduction factor," or simply the "shrink factor"
- As a heuristic, if the shrink factor is greater than 1.1 or so, you should worry that perhaps the chains have not converged adequately

Figure 7.10

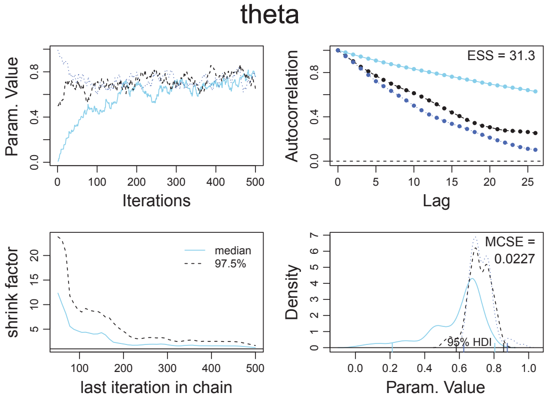


Figure 7.10: Illustration of MCMC diagnostics. Three chains were generated by starting a Metropolis algorithm at different initial values, with proposal SD=0.02 (cf. Fig. 7.4) for data $z = 35$, $N = 50$. Only steps 1–500 are shown here. See Fig. 7.11 for later steps in the chain. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Figure 7.11

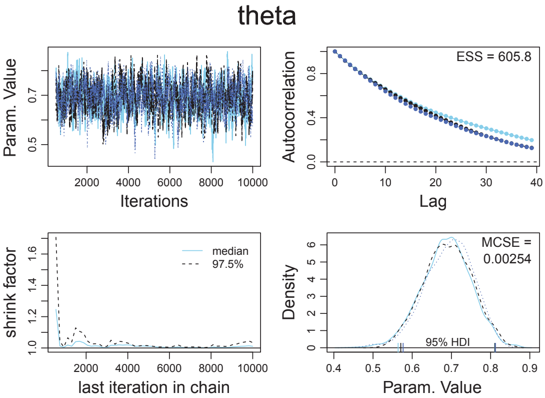


Figure 7.11: Illustration of MCMC diagnostics. Three chains were generated by starting a Metropolis algorithm at different initial values, with proposal SD=0.02 (cf. Fig. 7.4) for data $z = 35$, $N = 50$. Steps 500–10,000 are shown here. See Fig. 7.10 for earlier steps in the chain. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

MCMC accuracy

- The goal here is to have a large enough sample for stable and accurate numerical estimates of the distribution
- Since successive steps in a clumpy chain do not provide independent information about the parameter distribution, we need measures of chain length and accuracy that take into account the clumpiness of the chain
- We will measure clumpiness as autocorrelation, which is simply the correlation of the chain values with the chain values k steps (a.k.a., lags) ahead

Autocorrelation function

- The autocorrelation function is the autocorrelation across a spectrum of candidate lags
- The autocorrelation of the chain values, X_t, with the chain values k steps lags ahead, X_{t+k}, is denoted $\mathrm{ACF}(k)$:

$$\mathrm{ACF}(k) = \frac{E[(\mathrm{X}_t - \mu_t)(\mathrm{X}_{t+k} - \mu_{t+k})]}{\sigma_t \sigma_{t+k}},$$

  where μ and σ are the mean and standard deviation of the values
- The bottom panel of Figure 7.12 (see next page) shows an example of computing the autocorrelation function
- The plot of the ACF reveals that this chain is highly autocorrelated, that is, it is fairly clumpy

Figure 7.12

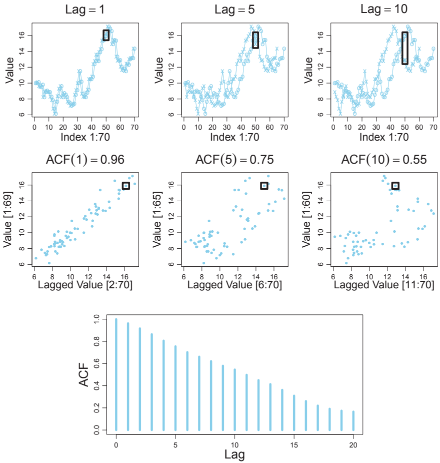


Figure 7.12: Autocorrelation of a chain. Upper panels show examples of lagged chains. Middle panels show scatter plots of chain values against lagged chain values, with their correlation annotated. Lowest panel shows the autocorrelation function (ACF). Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Autocorrelation function

- ACFs were also displayed in Figures 7.10 and 7.11 (see next two pages), in their upper-right panels
- As shown in the figures, the chains are highly autocorrelated, insofar as the autocorrelations remain well above zero for large lags

Figure 7.10

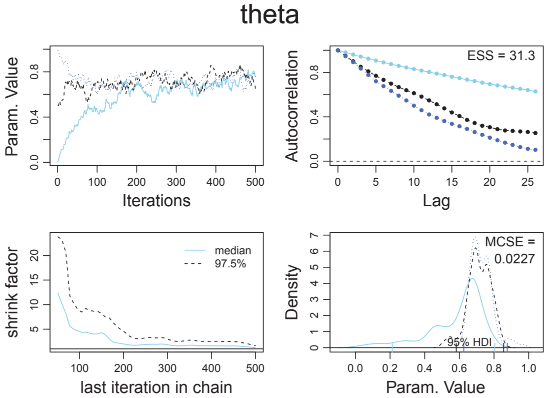


Figure 7.10: Illustration of MCMC diagnostics. Three chains were generated by starting a Metropolis algorithm at different initial values, with proposal SD=0.02 (cf. Fig. 7.4) for data $z = 35$, $N = 50$. Only steps 1–500 are shown here. See Fig. 7.11 for later steps in the chain. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Figure 7.11

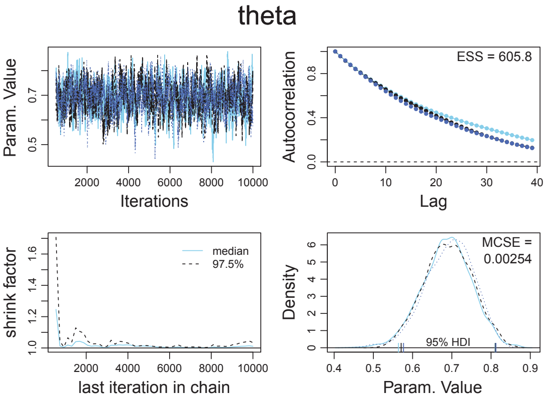


Figure 7.11: Illustration of MCMC diagnostics. Three chains were generated by starting a Metropolis algorithm at different initial values, with proposal SD=0.02 (cf. Fig. 7.4) for data $z = 35$, $N = 50$. Steps 500–10,000 are shown here. See Fig. 7.10 for earlier steps in the chain. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Effective sample size

- We would like some measure of how much independent information there is in autocorrelated chains
- In particular, we can ask, what would be the sample size of a completely non-autocorrelated chain that yielded the same information?
- An answer to this question is provided by a measure called the *effective sample size*, which divides the actual sample size by the amount of autocorrelation:

$$\mathrm{ESS} = \frac{N}{(1 + 2\sum_{k=1}^{\infty} \mathrm{ACF}(k))},$$

  where N is the sample size and $\mathrm{ACF}(k)$ the autocorrelation at lag k
- The upper-right panels of Figures 7.10 and 7.11 (see previous two pages) annotate the ESS across the multiple chains

Effective sample size

- How big should the ESS be for an accurate and stable picture of the posterior distribution?
- The answer depends on which detail of the posterior distribution you want a handle on
- For aspects of the distribution that are strongly influenced by dense regions, such as the median in unimodal distributions, the ESS does not need to be huge
- For aspects of the distribution that are strongly influenced by sparse regions, such as the limits of the 95% HDI, the ESS needs to be relatively large. Why?

Effective sample size

- How big should the ESS be for an accurate and stable picture of the posterior distribution?
- The answer depends on which detail of the posterior distribution you want a handle on
- For aspects of the distribution that are strongly influenced by dense regions, such as the median in unimodal distributions, the ESS does not need to be huge
- For aspects of the distribution that are strongly influenced by sparse regions, such as the limits of the 95% HDI, the ESS needs to be relatively large. Why?
    - sparse regions of the distribution are relatively rarely sampled in the chain, and therefore, a long chain is required to generate a high-resolution picture of sparse regions such as 95% HDI limits
    - for reasonably accurate and stable estimates of the limits of the 95% HDI, an ESS of 10,000 is recommended

Effective sample size

- To get an intuition for the (in-) stability of the estimates of the 95% HDI limits, we will repeatedly generate MCMC chains from a known distribution, which has precisely known true 95% HDI limits
- In particular, a standardized normal distribution has 95% HDI limits at very nearly - 1.96 and $+$ 1.96
- For each MCMC sample, we will estimate the 95% HDI
- Figures 7.13 and 7.14 (see next two pages) show the results

Figure 7.13

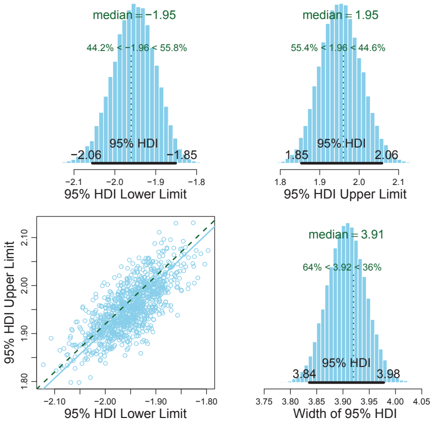


Figure 7.13: Estimated 95% HDI limits for random samples from a standardized normal distribution that have an *ESS of 10,000*. Repeated runs yield a distribution of estimates as shown here; there were 50,000 repetitions. Upper panels show estimate of HDI limits. Lower panels show estimate of HDI width. True values are indicated by dashed lines. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Figure 7.14

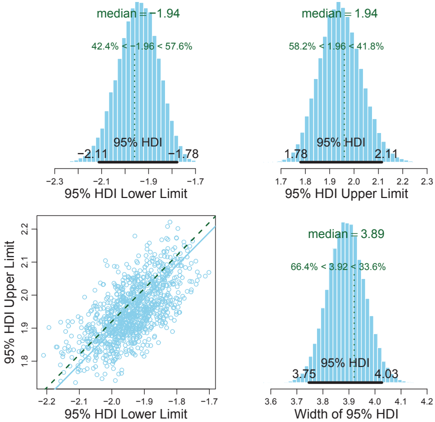


Figure 7.14: Estimated 95% HDI limits for random samples from a standardized normal distribution that have an *ESS of only 2,500*. Repeated runs yield a distribution of estimates as shown here; there were 50,000 repetitions. Upper panels show estimate of HDI limits. Lower panels show estimate of HDI width. True values are indicated by dashed lines. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Standard error

- If we sample N values many times, sometimes the sample mean $\bar{x}$ will be greater than μ and sometimes less than μ
- The Standard deviation (SD) of the sample mean, across repetitions, is called the standard error of the sample mean, and its estimated value is simply

$$\mathrm{SE} = \frac{\mathrm{SD}}{\sqrt{N}}$$

- The bigger the sample, the less noise there is in the estimate of the underlying mean, and the standard error itself provides a quantitative suggestion of how big the estimation noise is

Monte Carlo standard error (MCSE)

- Another useful measure of the effective accuracy of the chain is the Monte Carlo standard error (MCSE)
- A simple version of MCSE is defined as:

$$\mathrm{MCSE} = \frac{\mathrm{SD}}{\sqrt{\mathrm{ESS}}}$$

- The MCSE indicates the estimated SD of the sample mean in the chain, on the scale of the parameter value
- In Figure 7.11 (see next page), for example, despite the small ESS, the mean of the posterior appears to be estimated very stably

Figure 7.11

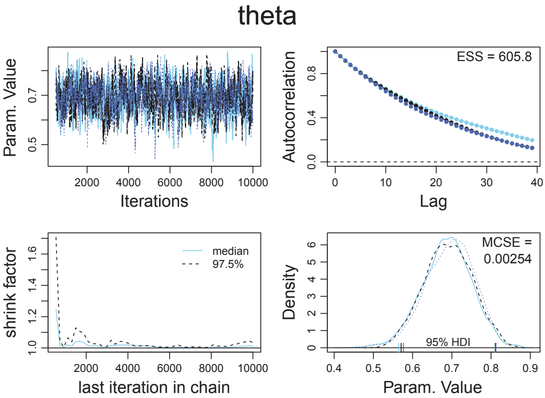


Figure 7.11: Illustration of MCMC diagnostics. Three chains were generated by starting a Metropolis algorithm at different initial values, with proposal SD=0.02 (cf. Fig. 7.4) for data $z = 35$, $N = 50$. Steps 500–10,000 are shown here. See Fig. 7.10 for earlier steps in the chain. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

MCMC efficiency

- It is often the case in realistic applications that there is strong autocorrelation for some parameters, and therefore, an extremely long chain is required to achieve an adequate ESS or MCSE
- Our goal here is to improve the efficiency of MCMC, so we do not exceed our patience and computing power
- There are various ways to (attempt to) improve the efficiency of the MCMC process
    - run chains on parallel hardware
    - adjust the sampling method (e.g., use a Gibbs sampler instead of a metropolis sampler)
    - change the parameterization of the model
    - thinning the chain reduces autocorrelation (and the size of space needed for saving the chain) but does not improve efficiency