Lecture 7: Markov Chain Monte Carlo

Professor Alexander Franks

2023-02-16

Announcements

- HW 4 out, due 3/5

- Quiz 3 out, Due tomorrow at 2PM.

Monte Carlo estimation

- $\bar{\theta} = \sum_{s=1}^{S} \theta^{(s)}/S \xrightarrow{\text{LLN}} \mathrm{E}[\theta|y_1, \ldots, y_n]$
- $\sum_{s=1}^{S} \left(\theta^{(s)} - \bar{\theta}\right)^2/(S-1) \xrightarrow{\text{LLN}} \mathrm{Var}[\theta|y_1, \ldots, y_n]$
- $\#\left(\theta^{(s)} \leq c\right)/S \to \Pr(\theta \leq c|y_1, \ldots, y_n)$
- the α-percentile of $\left\{\theta^{(1)}, \ldots, \theta^{(S)}\right\} \to \theta_\alpha$

Sampling from the posterior distributions

- The Monte Carlo methods we discussed previously assumed we could easily get samples from the posterior, e.g. with `rnorm`

- In general, sampling from a general probability distribution is hard

- Want to call `rcomplicatedistribution()` but don't have it

    - Inversion sampling is limited
    - Grid sampling is reasonable in 1 or 2 dimensions

Rejection sampling.

- In high dimensions, these approaches aren't sufficient

Markov Chain Monte Carlo

- We want independent random samples, $\theta^{(s)}$ from $p(\theta \mid y_1, \ldots y_n)$

Too Hard

- But there is no good way to get independent samples
- Alternative, create a sequence of **correlated** samples that converge to the correct distribution (time series)
- Markov Chain Monte Carlo gives us a way to generate correlated samples from a distribution

$\theta_1, \theta_2, \theta_3 \ldots \theta_t$ (correlated)

$$E\left[\frac{1}{t}\sum \theta_i\right] = \frac{1}{t}\sum_i^t E(\theta_i) = E[\theta | y]$$

Linearity of Expected.

Monte Carlo Error

- Reminder: $\bar{\theta} = \sum_{s=1}^{S} \theta^{(s)} / S$ and S is the number of samples.
- If the samples are independent,

$$\mathrm{Var}(\bar{\theta}) = \frac{1}{S^2} \sum_{s=1}^{S} \mathrm{Var}(\theta^{(s)}) = \frac{\mathrm{Var}(\theta \mid y_1, \ldots y_n)}{S}$$

- If the samples are *positively correlated*,

$$\mathrm{Var}(\bar{\theta}) = \frac{1}{S^2} \sum_{s,t} \mathrm{Cov}(\theta^{(s)}, \theta^{(t)}) > \frac{\mathrm{Var}(\theta \mid y_1, \ldots y_n)}{S}$$

- MCMC methods have higher Monte Carlo error due to positive dependence between samples.
- Hope to minimize dependence and thus MC error

Basics of Markov Chains

Markov Chains: Big Picture

- For standard Monte Carlo, we make use of the law of large number to approximate posterior quantities

- The law of large numbers can still apply to random variables that are not independent

- We have a sequence of random variables indexed in time, θ_t

- We'll be using a *discrete-time* Markov Chain: $t \in 0, 1, \ldots T$

- The observations, $\theta^{(t)}$ can be discrete or continous ("discrete-state" or "continuous-state" Markov Chain)

Discrete-state Markov Chains

- Let $\theta^{(t)} \in 1, 2, \ldots M$ be the state space for the Markov Chain
- A sequence is called a markov chain if

$$Pr(\theta^{(t+1)} \mid \theta^{(t)}, \theta^{(t-1)} \ldots \theta^{(1)}) = Pr(\theta^{(t+1)} \mid \theta^{(t)})$$

for all $t \geq 0$ (Memory of 1 time period)

- The **Markov property**: given the entire past history, $\theta^{(1)}, \ldots \theta^{(t)}$, the most recent $\theta^{(t+1)}$ depends only on the immediate past, $\theta^{(t)}$

The Transition Matrix

- Define $q_{ij} = Pr(\theta^{(t+1)} \mid \theta^{(t)})$ is the transition probability from state i to state j
- The $M \times M$ matrix $Q = (q_{ij})$ is called the *transition matrix* of the Markov Chain

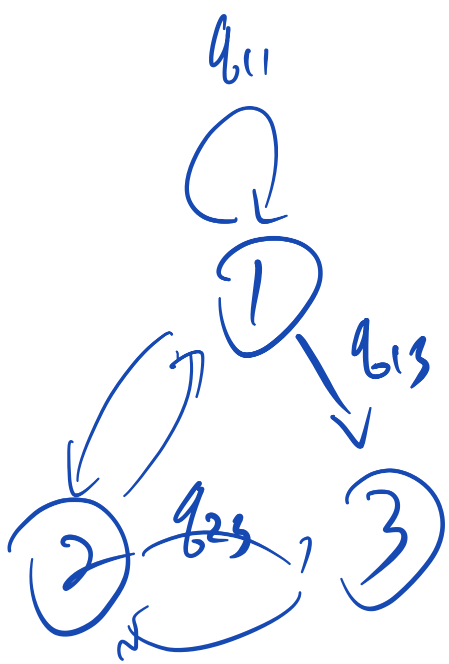
3-state example

end

start

$$Q = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix}$$

Sums within a row = 1



The Transition Matrix

3-state example

$$Q = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix}$$

- The rows of the transition matrix sum to 1
- Note: $Q^n = (q_{ij}^{(n)})$ is the probability of transitioning from i to j in n steps

$(Q \times Q)_{ij}$: Given I'm in i, prob of being in j after 2 steps.

What happens $n \to \infty$

The limiting distribution

- A regular, irreducible Markov chain has a **limiting probability distribution**
    - Cover definitions of regular and irreducible in PSTAT160 (or related)
- Limit distribution describes the long-run fraction of time the Markov Chain spends in each state
    - *Does not* depend on where the chain starts
- Let $\pi = (\pi_1, \ldots \pi_M)$ be a row vector of probabilities associated with each state, such that $\sum_{i=1}^{M} = \pi_i = 1$
    - The limiting distribution converges to π, which is said to be **stationary** because $\pi Q = \pi$
    - If you sample from the limiting distribution and then transition, the result is still distributed according to the limiting distribution

Markov Chain Example

- Sociologists often study social mobility using a Markov chain.
- In this example, the state space is `{low income, middle income, and high income}` of families
- Let $\mathbf{Q}$ be the transition matrix from parents income to childrens income

Income of kids

$\mathbf{Q} =$

	Lower	Middle	Upper
Lower	0.40	0.50	0.10
Middle	0.05	0.70	0.25
Upper	0.05	0.50	0.45

Income of parents

Multi-step Transition Probabilities

2-step transition probabilities

Grandkids

Parents

$$\mathbf{Q}^2 = \mathbf{Q} \times \mathbf{Q} = \begin{vmatrix} 0.1900 & 0.6000 & 0.2100 \\ 0.0675 & 0.6400 & 0.2925 \\ 0.0675 & 0.6000 & 0.3325 \end{vmatrix}$$

4-step transition probabilities

Great-Great Grandkids

$$\mathbf{Q}^4 = \mathbf{Q}^2 \times \mathbf{Q}^2 = \begin{vmatrix} 0.0908 & 0.6240 & 0.2852 \\ 0.0758 & 0.6256 & 0.2986 \\ 0.0758 & 0.6240 & 0.3002 \end{vmatrix}$$

Multi-step Transition Probabilities

4-step transition probabilities

$$\mathbf{Q}^4 = \mathbf{Q}^2 \times \mathbf{Q}^2 = \left|\begin{array}{ccc} 0.0908 & 0.6240 & 0.2852 \\ 0.0758 & 0.6256 & 0.2986 \\ 0.0758 & 0.6240 & 0.3002 \end{array}\right|$$

8-step transition probabilities

$$\mathbf{Q}^8 = \mathbf{Q}^4 \times \mathbf{Q}^4 = \begin{array}{cc} & \begin{array}{ccc} \text{Low} & \text{Med} & \text{H} \end{array} \\ \begin{array}{c} L \\ M \\ H \end{array} & \left|\begin{array}{ccc} 0.0772 & 0.6250 & 0.2978 \\ 0.0769 & 0.6250 & 0.2981 \\ 0.0769 & 0.6250 & 0.2981 \end{array}\right| \end{array}$$

construct a Markov chain

$p(\theta|y)$

Limiting distn.

The limiting distribution

$$\mathbf{Q}^{\infty} = \mathbf{1}\pi = \begin{vmatrix} \pi_1 & \pi_2 & \pi_3 \\ \pi_1 & \pi_2 & \pi_3 \\ \pi_1 & \pi_2 & \pi_3 \end{vmatrix}$$

```
Q <- matrix(c(0.4, 0.05, 0.05,
              0.5, 0.7, 0.5,
              0.1, 0.25, 0.45),
            ncol=3)

p <- eigen(t(Q))$vectors[, 1]
stationary_probs <- p/sum(p)
stationary_probs
```

```
## [1] 0.07692308 0.62500000 0.29807692
```

```
stationary_probs %*% Q
```

```
##            [,1]  [,2]      [,3]
## [1,] 0.07692308 0.625 0.2980769
```

Markov Chain Monte Carlo (MCMC)

- Incredible idea: create a Markov Chain with the desired limiting distribution $\to p(\theta|y)$
  - Want the limiting distribution to be the posterior distribution
- Unlike the previous examples, we will mostly work with *infinite* state space

Transition Density.

- Want $p(\theta^{(t+1)} \mid \theta^{(t)})$ to have limiting distribution $p(\theta \mid y)$
  - If we run the random walk for long enough, $\theta^{(t)}$ will be distributed approximately according to $p(\theta \mid y)$

The Independence Sampler (confusing)

Generalization of the rejection sampler

- The Metropolis algorithm tells us how to construct a transition matrix with the correct limiting distribution
  - The Independence Sampler is a special case of the Metropolis algorithm
- Sample from a proposal, $q(\theta)$. Best if $q(\theta)$ is close to $p(\theta \mid y)$.
- If $p(\theta \mid y) > 0$ then we need $q(\theta) > 0$

Don't need q to envelope $p(\theta/y)$

- At each iteration we have a choice:
  - Accept the new proposed sample
  - Or keep the previous sample for another iteration

$\theta_{t+1} = \theta_t$

The Independence Sampler

1. Initialize θ_0 to be the starting point for you Markov Chain

   & before

2. Choose a *proposal distribution*, $J(\theta^*)$
   - Propose a candidate value for the next sample
   - Best performance if density is very similar to target
3. Generate the candidate θ^* from the proposal distribution, J
4. Compute $r = \min(1, \frac{p(\theta^*|y)}{p(\theta_t|y)})$

   posterior density at proposed value

   posterior evaluated at current.

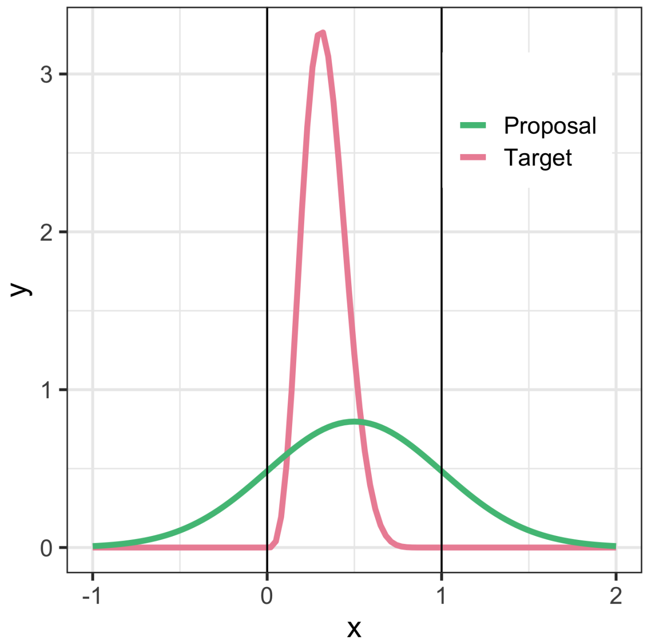
5. Set $\theta_{t+1} \leftarrow \theta^*$ with probability r
   - Generate a uniform random number $u \sim Unif(0,1)$
   - If $u < r$ we accept θ^* as our next sample
   - Else $\theta_{t+1} \leftarrow \theta_t$ (we do not update the sample this time)

Intuition

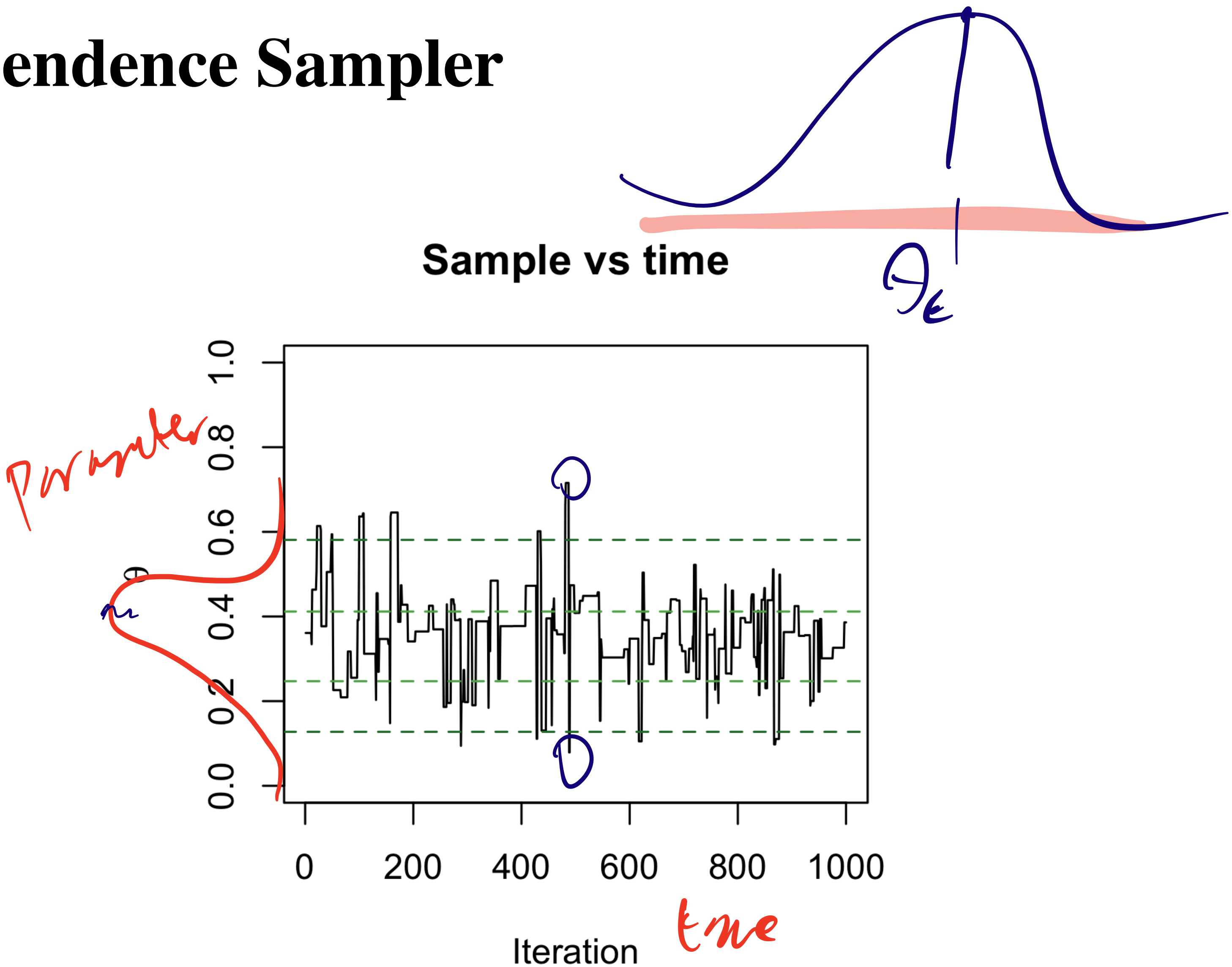
- If $p(\theta^* \mid y) > p(\theta_t \mid y)$ accept with probability 1
  - The proposed sample has higher posterior density than the previous sample
  - Always accept if we increase the posterior probability density
- If $p(\theta^* \mid y) < p(\theta_t \mid y)$ accept with probability $r < 1$
  - Accept with probability less than 1 if probability density would decrease
  - Relative frequency of θ^* vs θ_t in our samples should be $\frac{p(\theta^*|y)}{p(\theta_t|y)}$

An Example

- Let $P(\theta \mid y)$ be a Beta(5, 10) posterior distribution
- Propose from a distribution $J(\theta^{*}) \sim N(0.5, 1)$

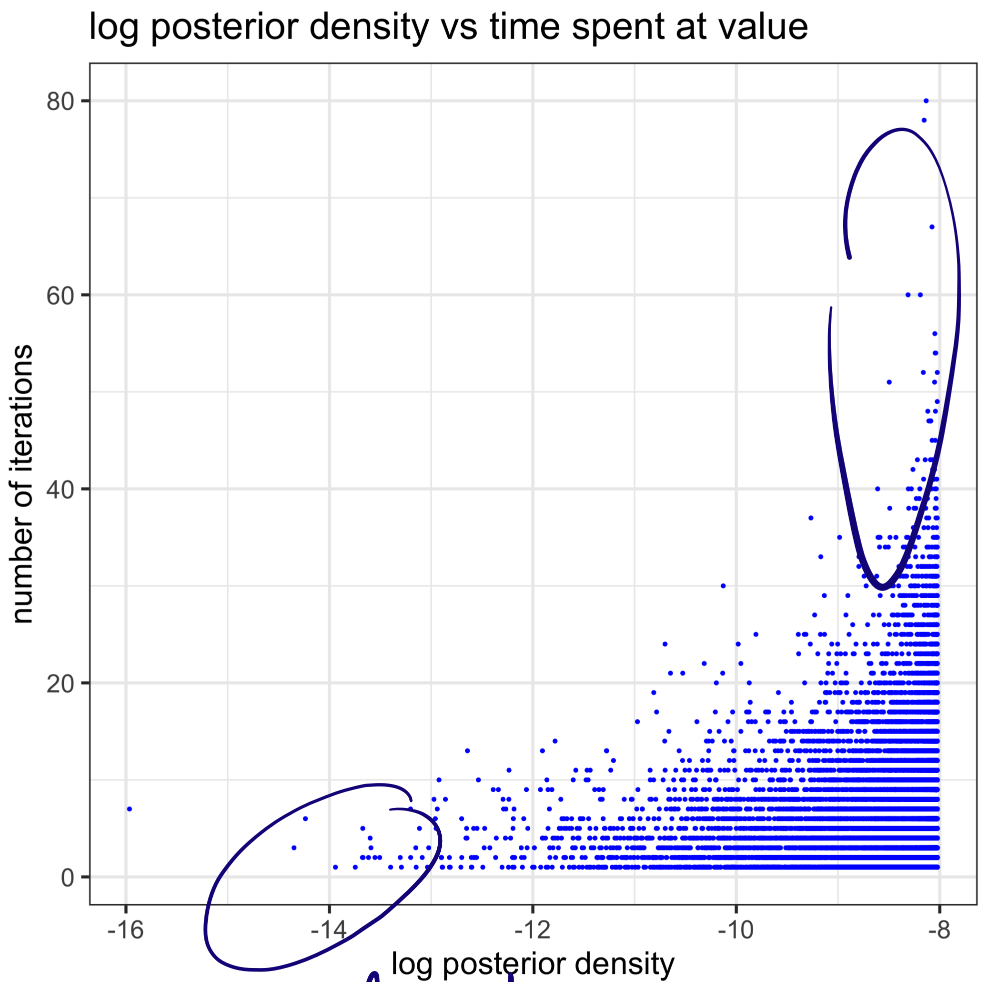


Independence Sampler



Note and source of confusion: samples are correlated over time for the "independence sampler".

Weighting by waiting



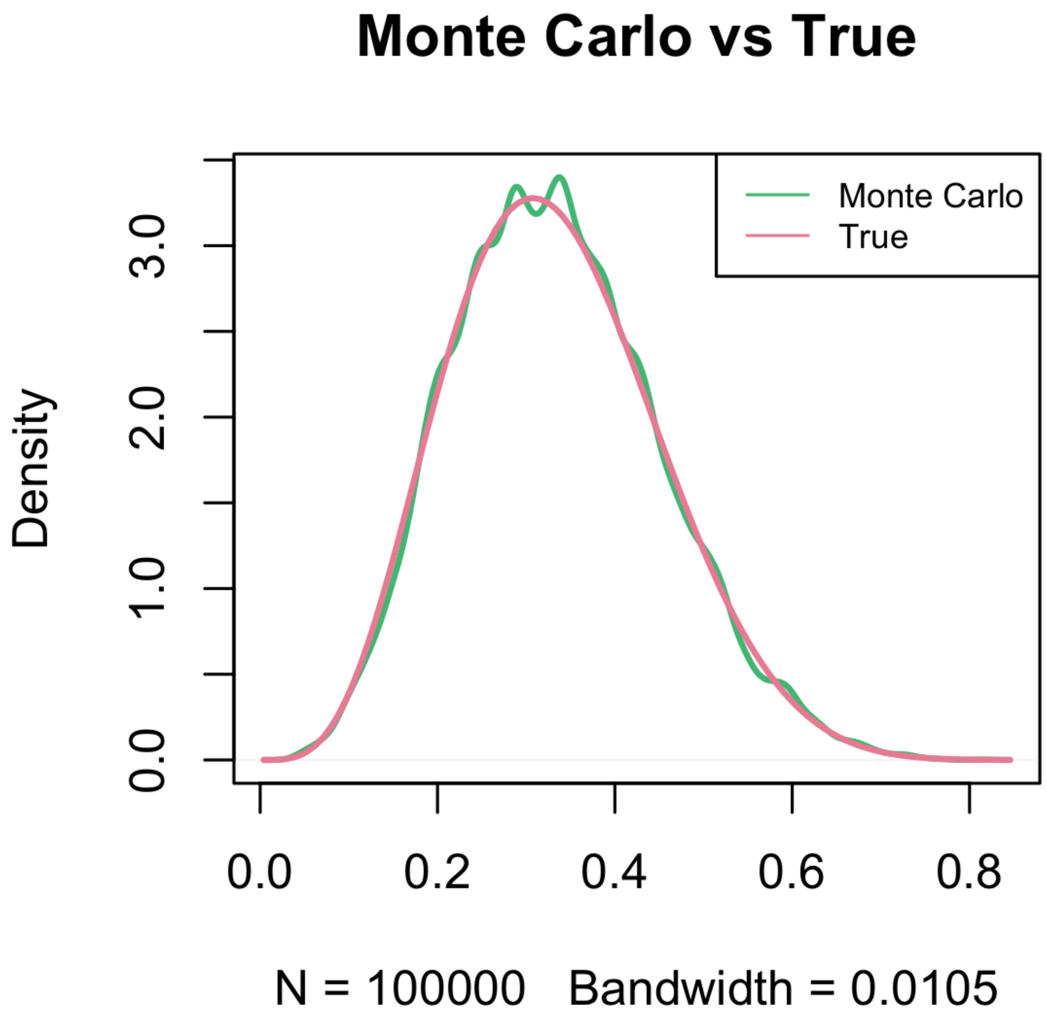
Stay for a while

High Density

Low density

Where did the sampler get stuck? Where does it quickly leave?

Independence Sampler



The Metropolis Algorithm

- Generalize the previous special case
- Allow the proposal distribution to depend on the most recent sample
    - Sometimes called an "Independence sampler": $J(\theta^*)$, e.g. $\theta^* \sim N(0.5, 1)$
    - Metropolis: $J(\theta^* \mid \theta_t)$, e.g. $\theta^* \sim N(\theta_t, 1)$
- Independence sampler: "Independence" refers to the proposal being fixed (the samples are **not** independent)!
- Metropolis sampler: a "moving" proposal distribution

The Metropolis Algorithm

1. Initialize θ_0 to be the starting point for you Markov Chain
2. Choose a proposal distribution, $J(\theta^* \mid \theta_t)$
   - Propose a candidate value for the next sample
   - Must have symmetry: $J(\theta^* \mid \theta_t) = J(\theta_t \mid \theta^*)$
3. Generate the candidate θ^* from the proposal distribution, J
4. Compute $r = \min(1, \frac{p(\theta^*|y)}{p(\theta_t|y)})$
5. Set $\theta_{t+1} \leftarrow \theta^*$ with probability r
   - Generate a uniform random number $u \sim Unif(0, 1)$
   - If $u < r$ we accept θ^* as our next sample
   - Else $\theta_{t+1} \leftarrow \theta_t$ (we do not update the sample this time)