Lecture 7: Markov Chain Monte Carlo

Professor Laura Baracaldo

Announcements

- Reading: Chapter 7 of Bayes Rules
- Homework 4 (the last one!): out 09/07, deadline on Wednesday, 09/13
- Last quiz on Monday

Monte Carlo estimation

- $\bar{\theta} = \sum_{s=1}^{S} \theta^{(s)}/S \to \mathrm{E}[\theta|y_1, \ldots, y_n]$
- $\sum_{s=1}^{S} \left(\theta^{(s)} - \bar{\theta}\right)^2/(S-1) \to \mathrm{Var}[\theta|y_1, \ldots, y_n]$
- $\#\left(\theta^{(s)} \leq c\right)/S \to \Pr(\theta \leq c|y_1, \ldots, y_n)$
- 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
- In high dimensions, these approaches aren't sufficient

Markov Chain Monte Carlo

- We want independent random samples, $\theta^{(s)}$ from $p(\theta \mid y_1, \ldots y_n)$
- But there is no good way to get independent samples
- Alternative, create a sequence of **correlated** samples that converge to the correct distribution
- Markov Chain Monte Carlo gives us a way to generate correlated samples from a distribution

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,

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

- If the samples are *positively correlated*,

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

- 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 $t \geq 0$

- 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

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

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

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 $\pi = (\pi_1, \dots \pi_M)$ 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

$$\mathbf{Q} = \begin{array}{|l|lll|} \hline & \text{Lower} & \text{Middle} & \text{Upper} \\ \hline \text{Lower} & 0.40 & 0.50 & 0.10 \\ \text{Middle} & 0.05 & 0.70 & 0.25 \\ \text{Upper} & 0.05 & 0.50 & 0.45 \\ \hline \end{array}$$

Multi-step Transition Probabilities

2-step transition probabilities

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

4-step transition probabilities

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

Multi-step Transition Probabilities

4-step transition probabilities

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

8-step transition probabilities

$$\mathbf{Q}^8 = \mathbf{Q}^4 \times \mathbf{Q}^4 = \begin{bmatrix} 0.0772 & 0.6250 & 0.2978 \\ 0.0769 & 0.6250 & 0.2981 \\ 0.0769 & 0.6250 & 0.2981 \end{bmatrix}$$

The limiting distribution

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

- The equation $\pi Q = \pi$ implies that the (row) vector π is a left eigenvector of Q with eigenvalue equal to 1
- Reminder: $Ax = \lambda x$ implies that x, a column vector, is a (right) eigenvector with eigenvalue λ

The limiting distribution

```
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

- Incredible idea: create a Markov Chain with the desired limiting distribution
    - Want the limiting distribution to be the posterior distribution
- Unlike the previous examples, we will mostly work with *infinite* state space
- Want $p(\theta^{(t+1)} \mid \theta^{(t)})$ to have limiting distribution $p(\theta \mid y)$
    - If $p(\theta^{(t+1)} \mid \theta^{(t)})$ is constructed correctly, and we run the chain long enough, $\theta^{(t)}$ will be distributed approximately according to $p(\theta \mid y)$

The Independence 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, $J(\theta)$. Best if $J(\theta)$ is close to $p(\theta \mid y)$.
- If $p(\theta \mid y) > 0$ then we need $J(\theta) > 0$
- At each iteration we have a choice:
    - Accept the new proposed sample
    - Or keep the previous sample for another iteration

The Independence Sampler

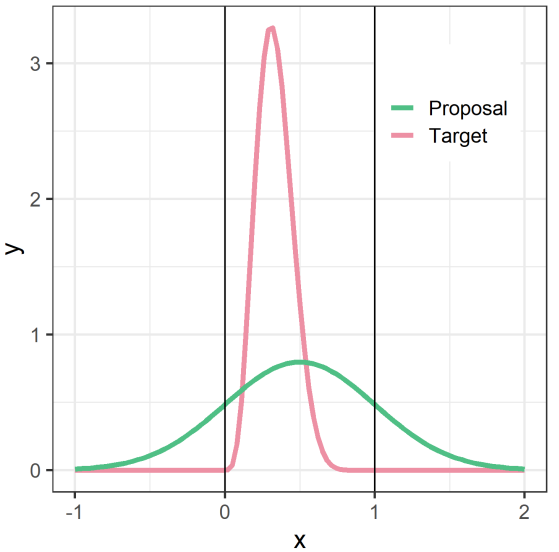
1. Initialize θ_0 to be the starting point for you Markov Chain
2. Choose a *proposal distribution*, $J(\theta^*)$
    - Propose a candidate value for the next sample
    - Best performance if density is very similar to target
3. Generate the candidate θ^* from the proposal distribution, J
4. Compute $r = \min(1, \frac{p(\theta^*|y)}{p(\theta_t|y)})$
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

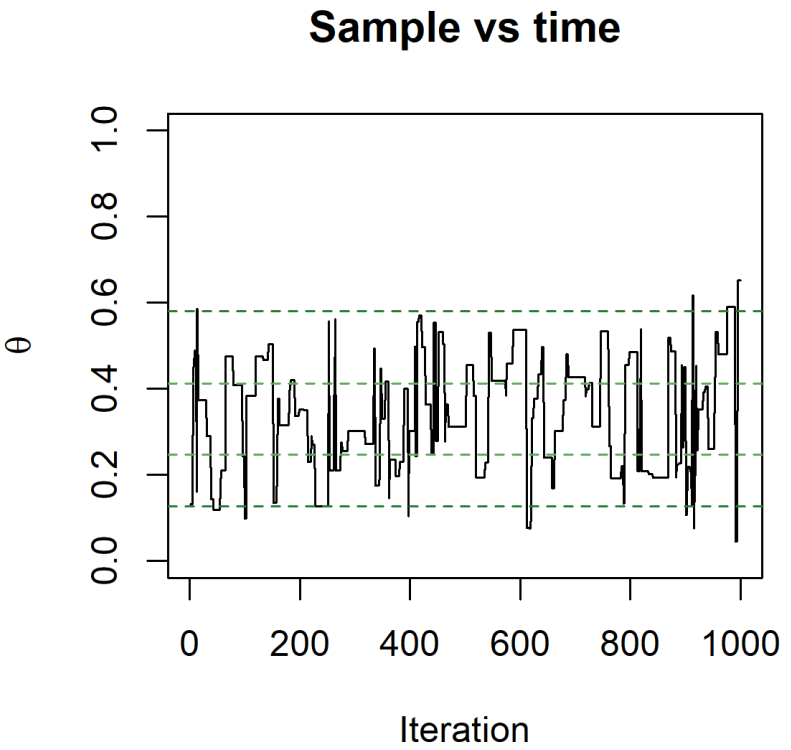
- If $p(\theta^* \mid y) > p(\theta_t \mid y)$ accept with probability 1
  - The proposed sample has higher posterior density than the previous sample
  - Always accept if we increase the posterior probability density
- If $p(\theta^* \mid y) < p(\theta_t \mid y)$ accept with probability $r < 1$
  - 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 $J(\theta^*) \sim N(0.5, 1)$

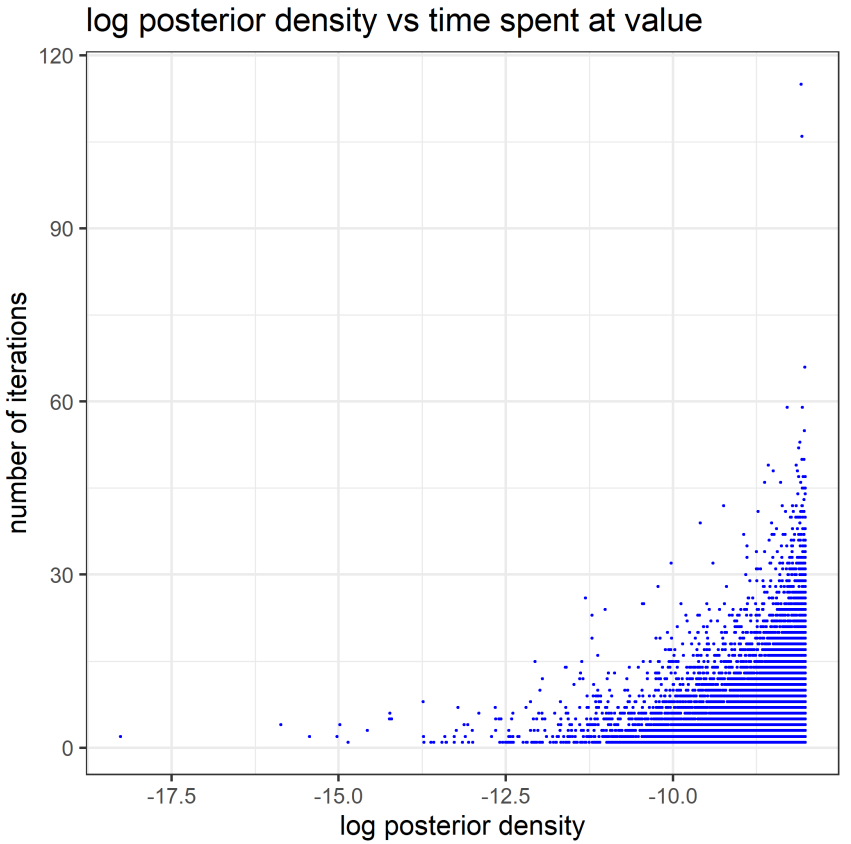


Independence Sampler



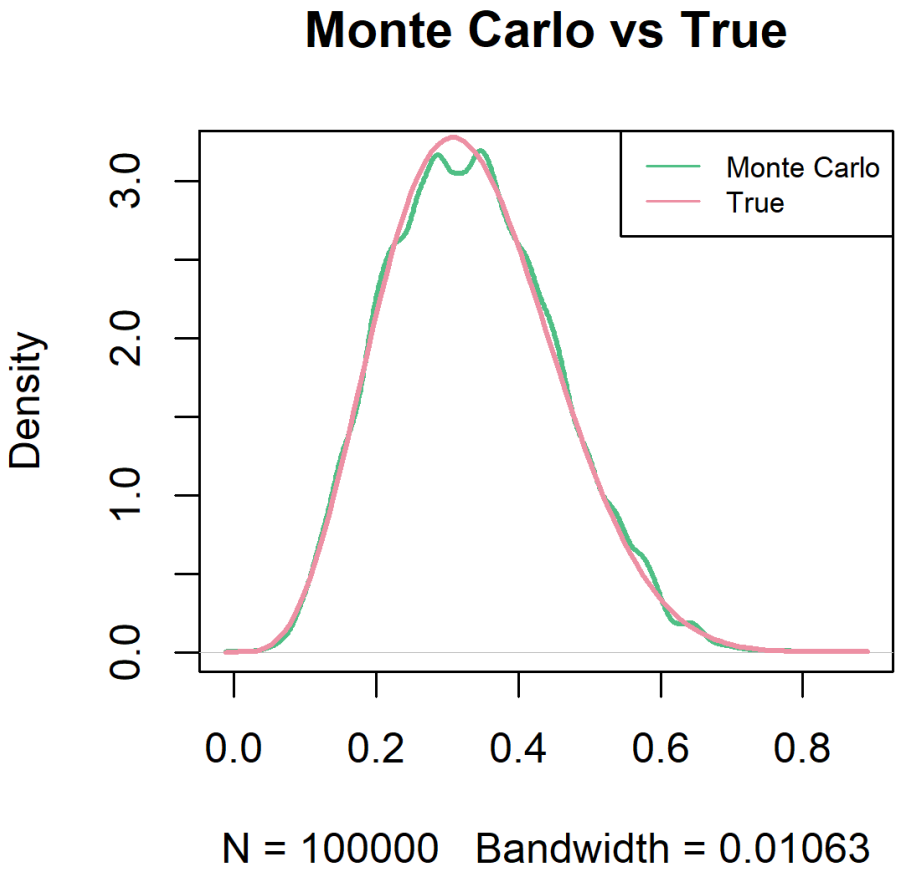
Note and source of confusion: samples are correlated over time for the "independence sampler".

Weighting by waiting



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)

Metropolis Algorithm

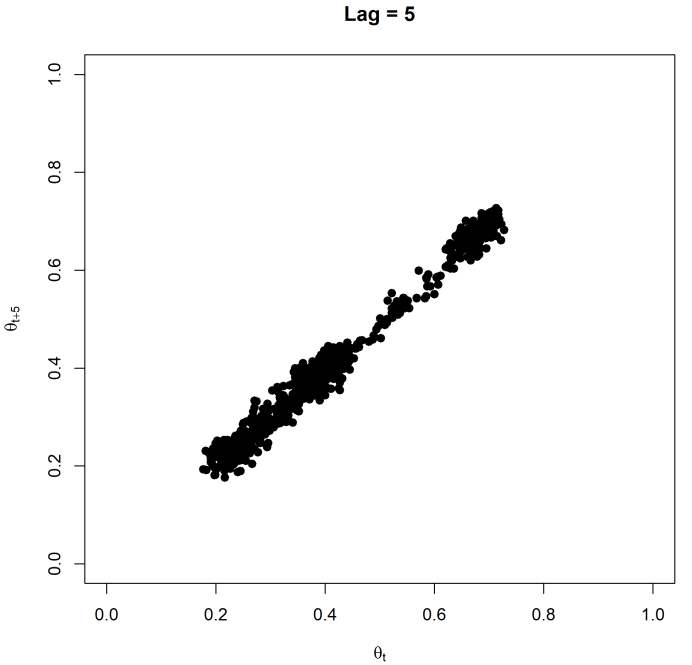
- Let $P(\theta \mid y)$ be a Beta(5, 10) posterior distribution
- 1-d sampling: lets try sampling from the Beta using the Metropolis algorithm
- Initialize θ_0 to 0.9
  - Note that the probability of drawing a value larger than 0.9 from a Beta(5, 10) is smaller than 1e-8
  - Our initial value is far from the high posterior density
  - In the long run this won't matter
- Define transition kernel $J(\theta_{t+1} \mid \theta_t)$ as $\theta^* \sim N(\theta_t, \tau^2)$
  - How does choice of τ^2 effect performance of MC sampler?

Demo

Autocorrelation of the Markov Chain

$\tau^2 = 0.01$ ("small" proposal variance)

Plot θ^t vs θ^{t+5} for all values of t

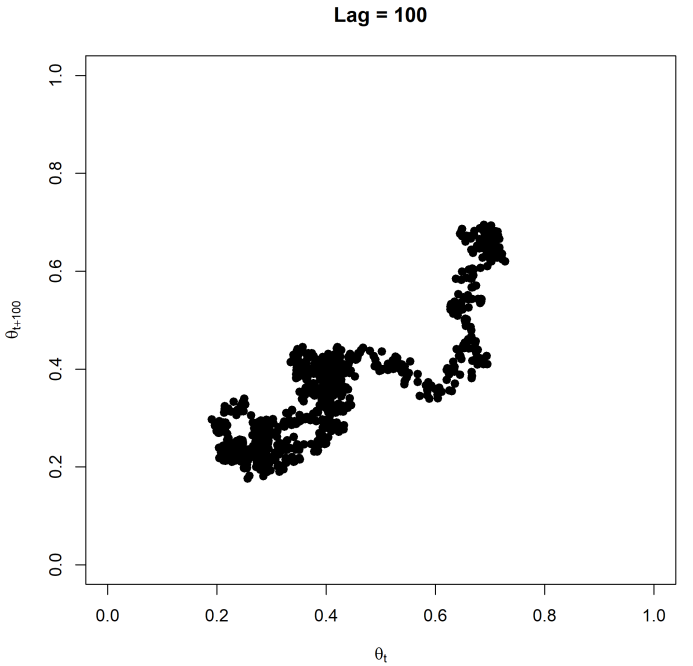


Correlation = 0.9871179

Autocorrelation of the Markov Chain

$\tau^2 = 0.01$ ("small" jump)

Plot θ^t vs θ^{t+100} for all values of t

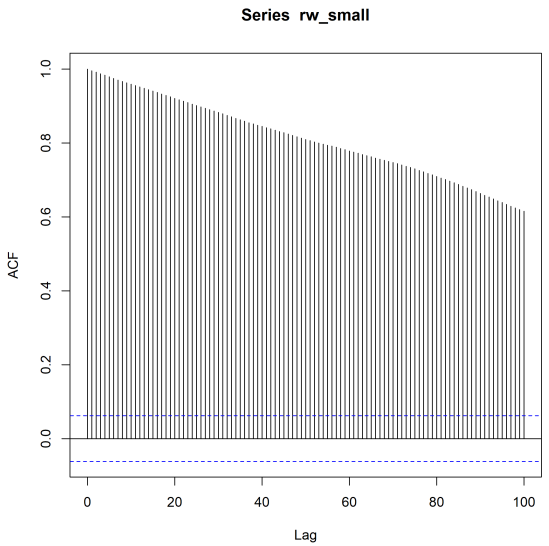


Correlation = 0.792115

The Metropolis Algorithm

$\tau^2 = 0.01$ ("small" jump)

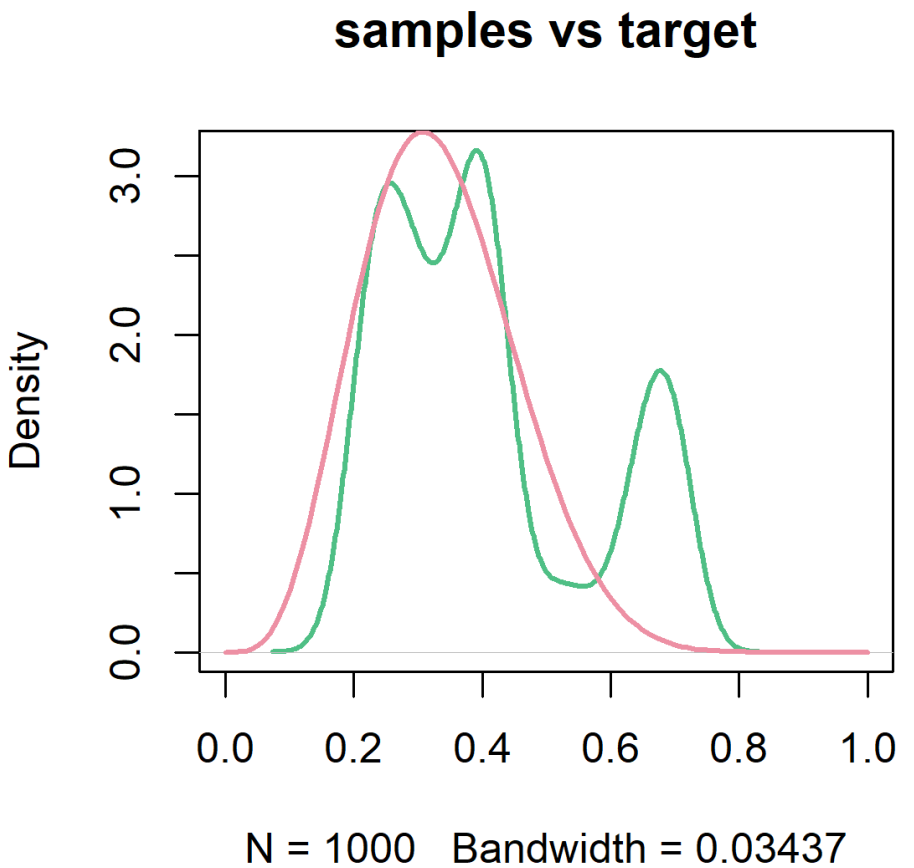
```
acf(rw_small, lag=100)
```




```
print(sprintf("Effective sample size: %.2f, Rejection Rate: %.2f",
              effectiveSize(rw_small), rejectionRate(as.mcmc(rw_small))))
```

The Metropolis Algorithm

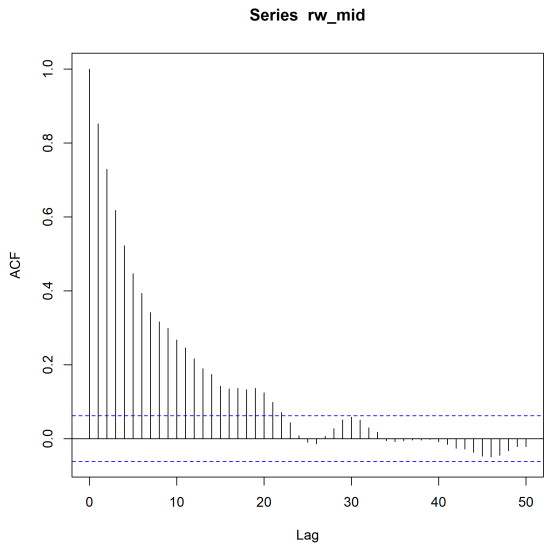
$\tau^2 = 0.01$ ("small" jump)



The Metropolis Algorithm

$\tau^2 = 0.1$ ("medium" proposal variance)

```
acf(rw_mid, lag=50)
```

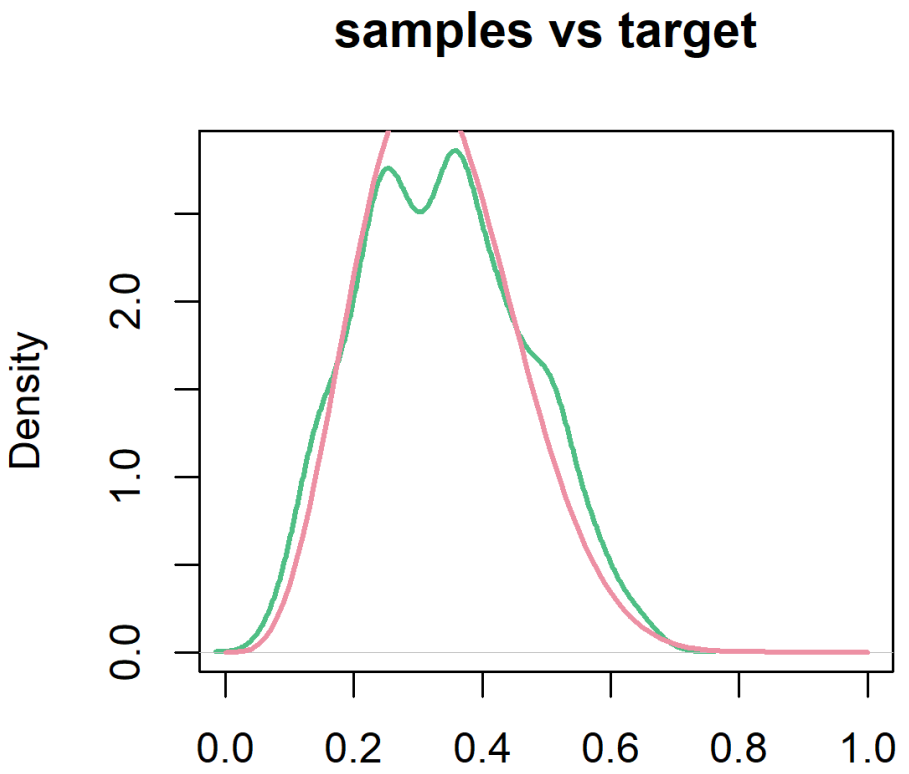



```
print(sprintf("Effective sample size: %.2f, Rejection Rate: %.2f",
              effectiveSize(rw_mid), rejectionRate(as.mcmc(rw_mid))))
```

```
## [1] "Effective sample size: 79.95, Rejection Rate: 0.28"
```

The Metropolis Algorithm

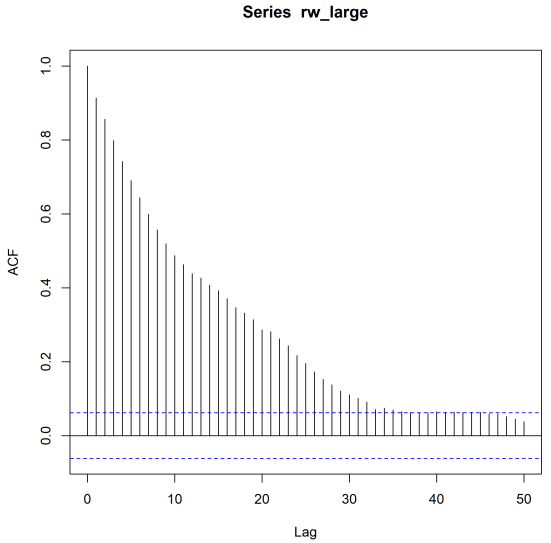
$\tau^2 = 0.1$ ("medium" proposal variance)



The Metropolis Algorithm

$\tau^2 = 2$ ("large" jump)

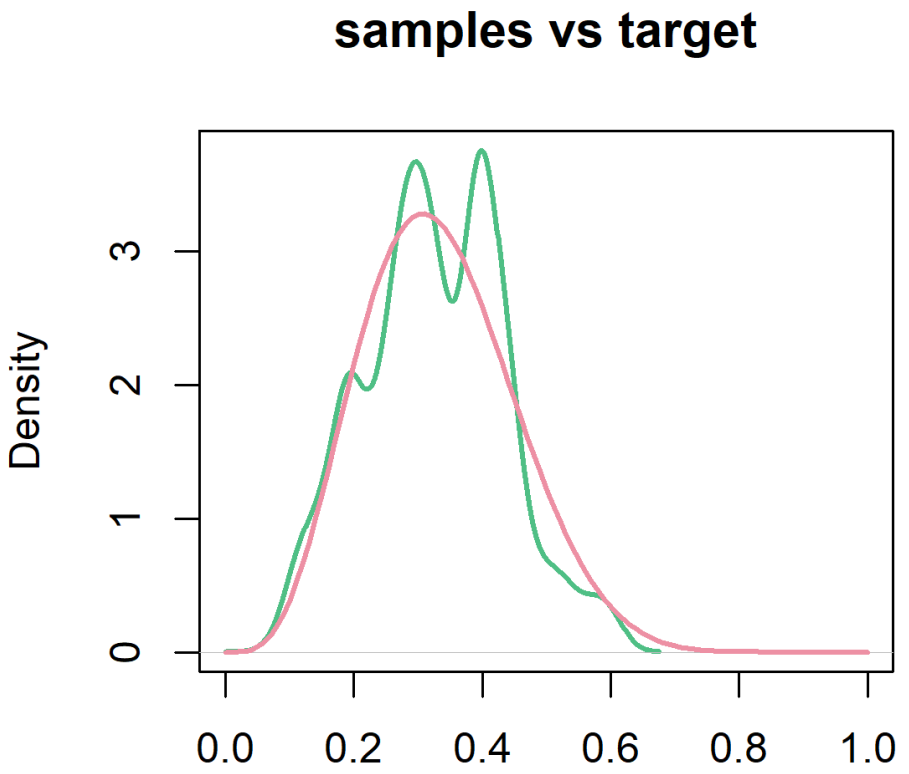
```
acf(rw_large, lag=50)
```



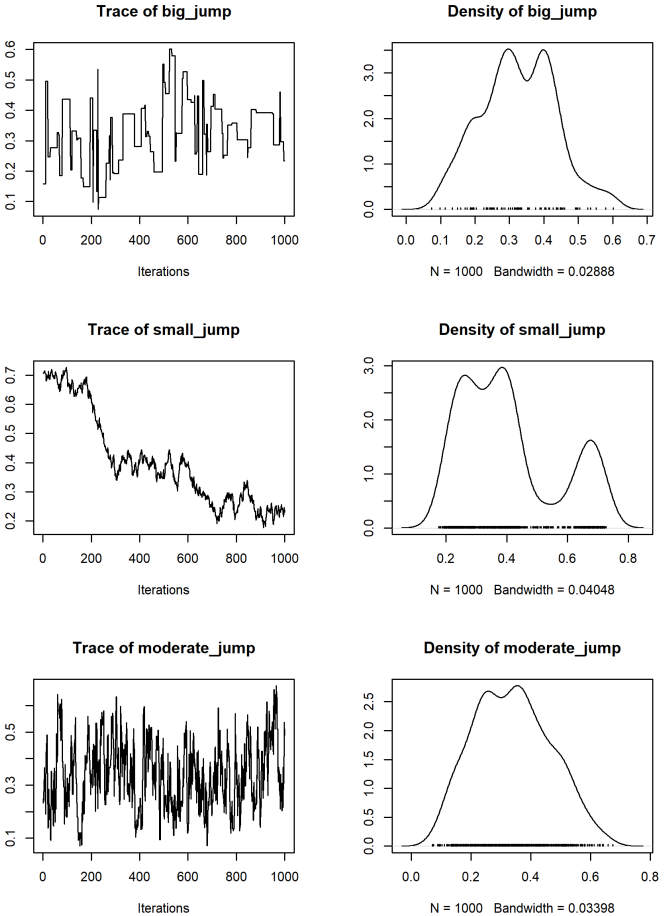

```
print(sprintf("Effective sample size: %.2f, Rejection Rate: %.2f",
              effectiveSize(rw_large), rejectionRate(as.mcmc(rw_large))))
```

The Metropolis Algorithm

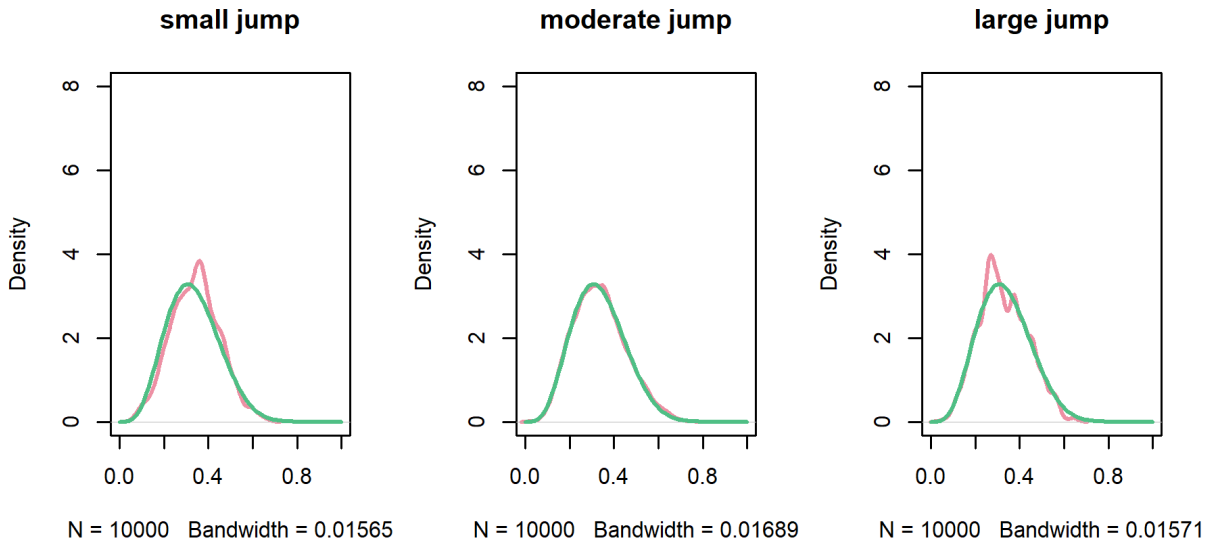
$\tau^2 = 2$ ("large" proposal variance)



Small, Moderate and Large Proposal variance



10,000 Samples




```
## [1] "Effective sample size (small proposal variance): 20.57"

## [1] "Effective sample size (medium proposal variance): 887.30"

## [1] "Effective sample size (large proposal variance): 491.57"
```

MCMC diagnostics

- Diagnose the chain performance by examining:
  - Rejection rate
  - Autocorrelation
  - Effective sample size

MCMC diagnostics

- **Rejection rate**: if rejection rate is high, the proposal density is proposing "too far away" from the current sample
- Means we are often keeping the previous sample
- Sampler is "sticky". Traceplots look like cityscapes.

MCMC diagnostics

- **Effective sample size:** correlated chain of samples is equivalent to this number of independent samples
    - High rejection rate implies a lot of duplicate samples so effective size is smaller than number of iterations
    - High autocorrelation means neighboring samples are very similar (even if not exactly the same)

MCMC diagnostics

- **Autocorrelation**: if samples are highly correlated, the proposal density is proposing "too close" to the current sample
    - Highly correlated implies the Markov chain is mixing slowly
- The mixing time of a Markov chain is the time until the Markov chain is "close" to its limiting distribution.

Metropolis Algorithm

- In the Beta example, the accept ratio, $\min(1, \frac{p(\theta^*|y)}{p(\theta_t|y)})$ is zero when $\theta^* > 1$ or $\theta^* < 0$
- If τ^2 (proposal variance) too large, you will often reject your proposal
    - Proposing far from your current location may move you too far out of the high density areas
    - This makes for a "sticky" chain (stay at current sample for a long time)
- τ^2 too small, the chain explore the parameter space slowly
- A rule of thumb is to aim for 30%-40% acceptance rate for random walk samplers.
    - This balances "stickiness" and slow convergence

Metropolis-Hastings Algorithm

- The Metropolis-*Hastings* algorithm allows us to use non-symmetric proposals
- The Hastings correction is needed when $J(\theta^* \mid \theta_t) \neq J(\theta^t \mid \theta^*)$

$$r = \min(1, \frac{p(\theta^* \mid y)}{p(\theta_t \mid y)} \frac{J(\theta^t \mid \theta^*)}{J(\theta^* \mid \theta_t)})$$

- For symmetric proposals $\frac{J(\theta^t|\theta^*)}{J(\theta^*|\theta_t)} = 1$

MCMC for multivariate distributions

- Modeling wing length of different specifies of midge (small, two-winged flies)
- Reminder: $Y_i \sim N(\mu, \sigma^2)$
- $P(\mu \mid \sigma^2)$, $\mu \sim N(\mu_0, \frac{\sigma^2}{\kappa_0})$
- $P(\sigma^2) \propto \frac{1}{\sigma^2}$ (improper prior)

MCMC for multivariate distributions

Example: midge wing length

- Modeling wing length of different specifies of midge (small, two-winged flies)
- From prior studies: mean wing length close to 1.9mm with sd close to 0.1mm
- $\mu_0 = 1.9$, $\sigma_0^2 = 0.01$
- Choose $\kappa_0 = 1$
- We will run 2 separate chains at different starting locations
- $J(\mu*, \log(\sigma*)) \sim N((\mu^t, \log(\sigma^t)), \begin{pmatrix} \tau_\mu^2 & 0 \\ 0 & \tau_{log\sigma}^2 \end{pmatrix})$

Initialization and Convergence

- In the long run, it doesn't matter where you initialize your sampler
- In practice, we can only run an algorithm for a finite amount of time
    - Need to check with the sampler has converged to the limiting distribution
    - Exclude samples close to the initial values since these are unlikely to be representative samples
    - We call the time to convergence **burn in** and throw away and samples generated during this time.
- How do we know when a sampler has converged to the limiting distribution?

Running multiple chains

- How do we know when a sampler has converged to the limiting distribution?
    - Hard to know for sure.
- Idea: run multiple chains at very different initial locations
    - If the chains are very far apart we haven't converged
    - If the chains end up in the same place, we have confidence that its reached convergence

Diagnosing mixing with Rhat

$$B = \frac{n}{m-1} \sum_{j=1}^{m} \left(\bar{\psi}_{.j} - \bar{\psi}_{..} \right)^2$$

$$W = \frac{1}{m} \sum_{j=1}^{m} \boldsymbol{s}_j^2$$