# An Introduction to Markov Chain Monte Carlo (MCMC) Sampling Methods

Francisco M. Beltrán

Nov 21, 2016

LLNL-PRES-676046

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

# Outline

- Introduction
  - History
  - Basic theory
- Sampling methods
  - Rejection
  - MCMC
    - Metropolis-Hastings
    - Gibbs steps
- Bayesian Example
- Discussion

# Introduction

- The name ‘Monte Carlo’ came after the casino in Monte Carlo, a small hillside town in Monaco, since some of the first computer simulations were used for gambling.
  - Soon became a technical term for simulation of random process

- It was picked by a physicist Fermi (Italian born American) who was among the first using the sampling techniques in his effort building the first manmade nuclear reactors in 1942.

Place du Casino, Monaco

# Introduction: Cont

- Markov Chain Monte Carlo (MCMC)
  - Invented at Los Alamos soon after Ordinary Monte Carlo
  - Used by Metropolis et al. (1953) to simulate liquid in equilibrium with its gas phase. They realized that they did not need to simulate the whole dynamic but only the some Markov chain having the same equilibrium distribution.
  - Simulations following this scheme were known as the Metropolis algorithm.
- With the availability of computers, MCMC was widely used by physicist and chemists.
- Did not become widely known to statisticians until after 1990.

# Introduction (conti)

- In 1970, W.K. Hastings generalized the Metropolis algorithm to become the Metropolis-Hastings algorithm.
- A special case of the Metropolis algorithm was introduced Geman and Geman (1984) known as the Gibbs sampler.
  - Optimization for finding the posterior mode rather than simulation.
  - Simulates directly from the posterior distribution.
  - Also known as ‘Data Augmentation’
- Gelfand and Smith (1990) popularized the Gibbs sampler in the Bayesian community.

# MCMC Applications

- Plays a significant role in:
  - Statistics
  - Econometrics
  - Physics
  - Computer Science
- Sampling from High Dimensional and complex distributions.
- Essential in Bayesian Statistics.
  - Bayesian Inference is about the quantification and propagation of uncertainty in light of the observations of the system.
  - Defined via a probability.
  - From a Prior to a Posterior
  - Different from classical inference which tends to be concerned with parameter estimation.

# Bayesian Inference and Learning

There are three essential steps to Bayesian data analysis

- Defining the full joint probability distribution for both observable y and parameters x.

$$p(x, y) = p(y \mid x)p(x)$$

- Condition on the data:

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{\int_X p(y \mid x)p(x)dx}$$

- Check your model.

# MCMC: Why go through all the 'trouble'?

- We don't know how to draw from $p(x|y)$.
- MCMC is a general method that simultaneously solves inference on $\{p(x|y), p(x_i|y), p(\hat{y}|y)\}$
- Only requires evaluation of the joint distribution up to a proportionality constant.

**How it works:**

- MCMC methods construct a Markov chain on the state space of x whose steady state distribution is the posterior distribution (target distribution) $p(x|y)$

# Goals of using sampling methods

- MCMC: Sampling using local information
  - Generic ‘problem solving technique’
- Sample from a distribution that is known up to a proportionality constant.

# Rejection Sampling

- Suppose the target distribution p(x) is known up to a proportionality constant

$$p(x) \propto 0.7\exp(-0.2x^2) + 0.3\exp(-0.2(x-10)^2)$$

- Sample from a distribution g(x) such that

$$f(x) \leq Mg(x)$$

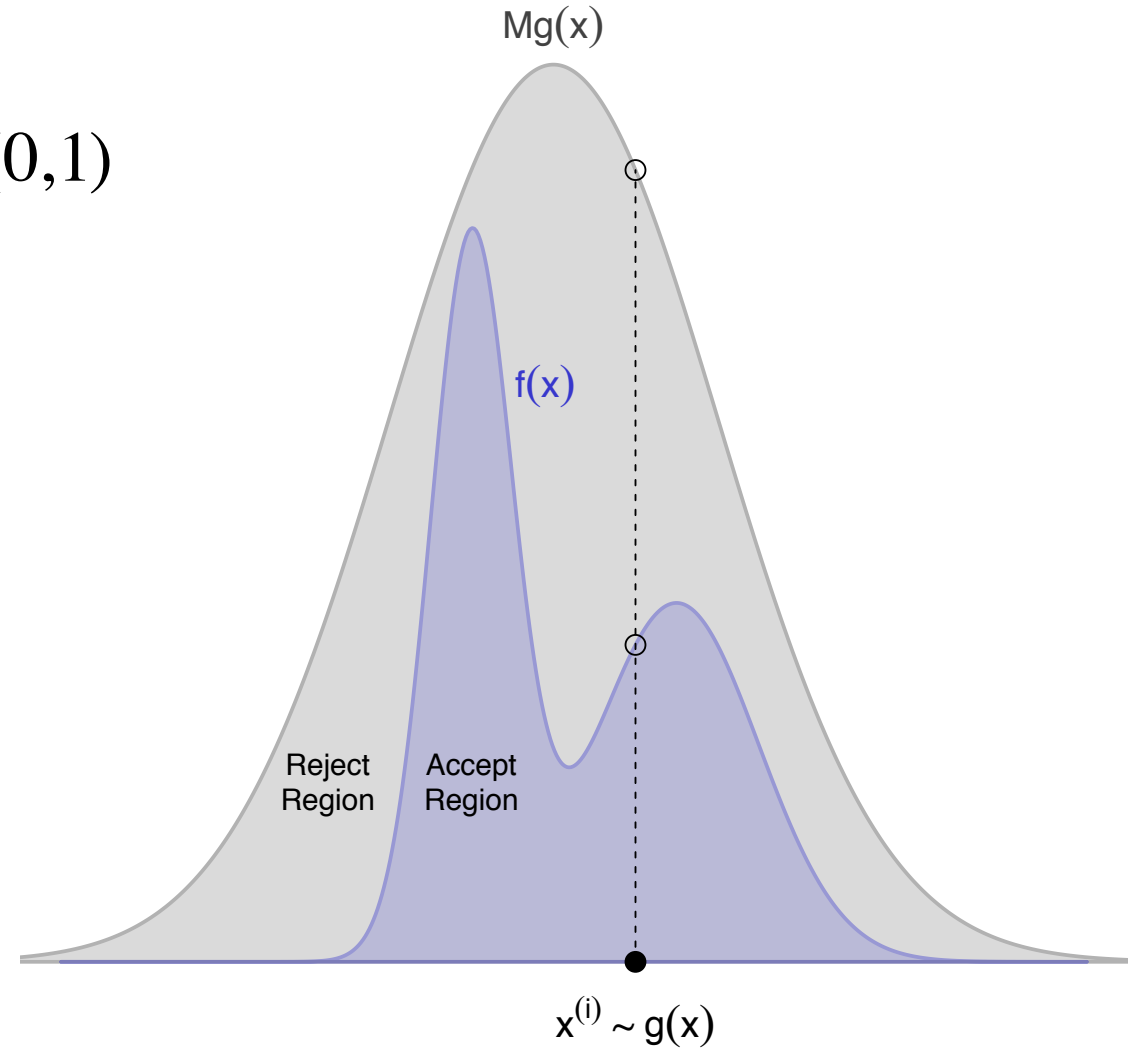
- Where the scaling parameter $M<\infty$

# Rejection Sampling Algorithm

- For i=1, sample

$$x^i \sim g(x) \text{ and } u \sim U(0,1)$$

- If $u < \dfrac{p(x^i)}{Mg(x^i)}$
  - Accept $x^i$
- Else
  - Reject $x^i$
- i = i+1
- Repeat until i=I



# Rejection Sampling: Severe Limitations

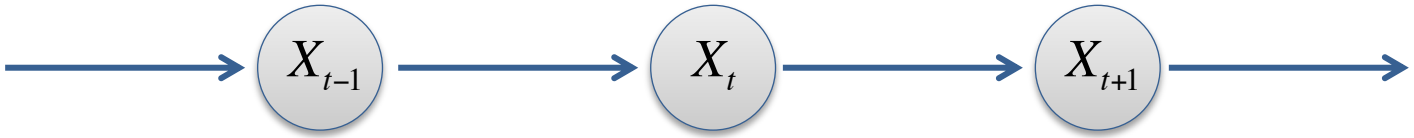
- Not always possible to bound p(x)/g(x) with a reasonable M.
- If M is large the acceptance probability is small.
- In high dimensions, it is exponentially slow to sample points.

# What is a Markov Chain?

A Markov chain is a mathematical model for stochastic systems whose states, discrete or continuous, are governed by a transition probability.

- A 1st order Markov chain depends on the most recent previous state:

$$X_{t+1} \mid X_t, \ldots, X_1 \sim p(X_{t+1} \mid X_t, \ldots, X_1) = p(X_{t+1} \mid X_t)$$




- The Markovian property means "locality" in space or time.

# Markov Chains

- Definition of Markov Chain
  - A sequence $X_1,\ldots,X_t$ of random elements of some set is a Markov Chain if the distribution of $X_{t+1}$ given $X_1,\ldots,X_t$ depends only on $X_t$.
  - Set in which the $X_i$ take values is called the state space of the Markov Chain
  - The Markov Chain has stationary transition probabilities if the conditional distribution of $X_{t+1}$ given $X_t$ does not depend on n.
    - This is the MAIN kind of Markov Chain of interest in MCMC

- Markov Chain:

$$p(X_{t+1} \mid X_t, \ldots, X_1) = p(X_{t+1} \mid X_t)$$

# Markov Chains

- Our goal is to find conditions under which the Markov Chain converges to a unique limit distribution
  - Independent from the starting state distribution
- If the limiting distribution exists it must be a stationary distribution
- Important to design the sampler that converges quickly

# Metropolis-Hastings Algorithm

- General procedure for MCMC simulations from a target dist.
- We are interested in an updating algorithm that preserves a specified distribution (target distribution).
  - Can construct Markov chains to sample that distribution
- Suppose that the specified distribution has an unnormalized density p(x).
  - p(x) is a nonnegative function that integrates to a non-zero finite value
- MH algorithms generate Markov chains which converge to f(x), by successively sampling from an (essentially) arbitrary proposal distribution q(x|x*) (i.e. a Markov transition kernel) and imposing a random rejection step at each transition.

# Metropolis-Hastings Algorithm

1. Start at the current state $x^{(i)}$
2. Propose a move to state $x^{(*)}$ having conditional probability density $q(.|x^{(i)})$ (**Proposal Density**)
3. Calculate the acceptance ratio (Hastings ratio)

$$r(x^{(i)}, x^{(*)}) = \frac{p(x^{(*)}) / q(x^{(*)} \mid x^{(i)})}{p(x^{(i)}) / q(x^{(i)} \mid x^{(*)})}$$

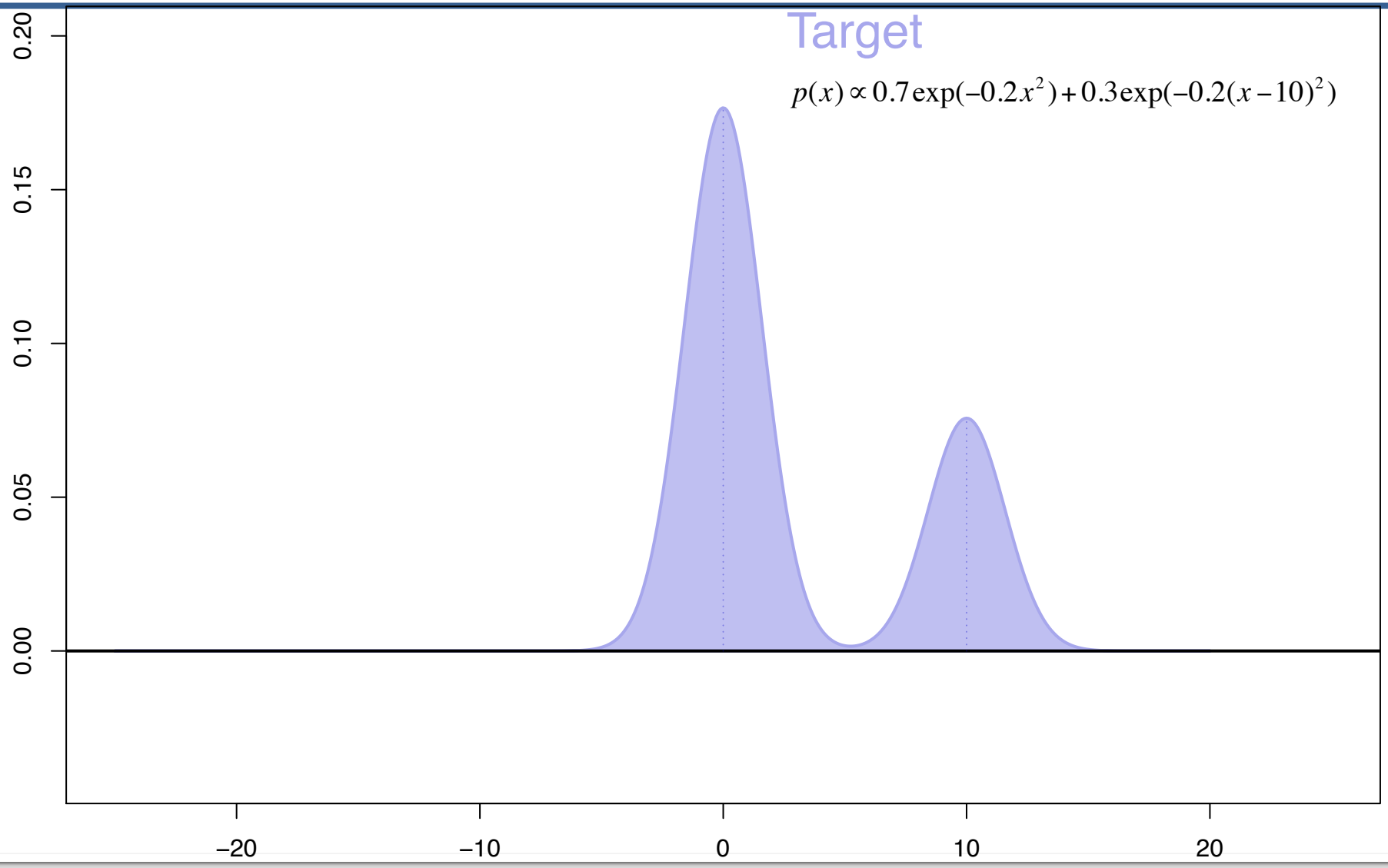
Note: The normalizing constant cancels if q is symmetric.

4. Accept the proposed move to $x^{(*)}$ with probability

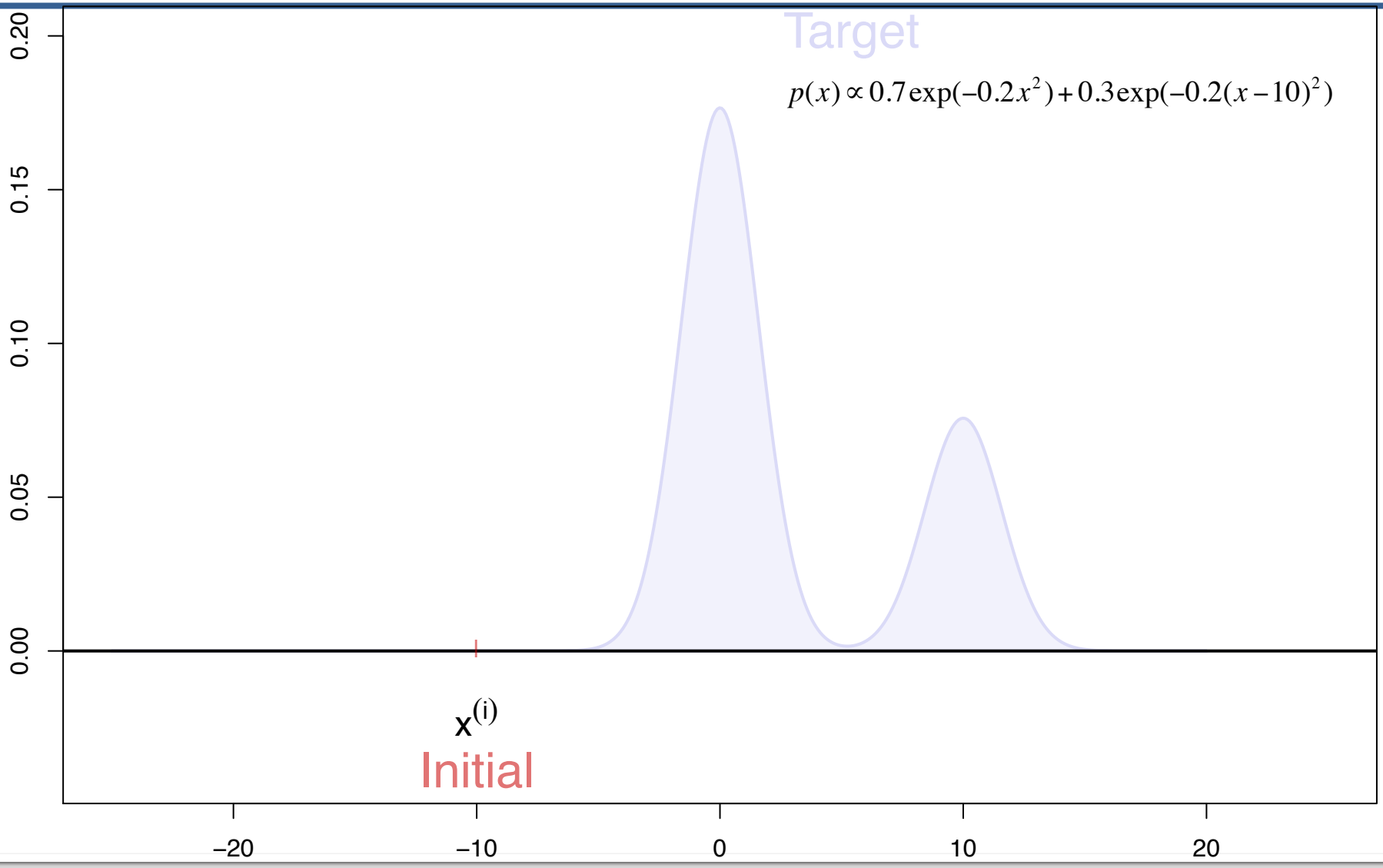
$$\alpha(x^{(i)}, x^{(*)}) = \min\left(1, r(x^{(i)}, x^{(*)})\right)$$

- In the special case of a symmetric proposal density, the ratio is simplified. The proposal is user define and is more art than science.

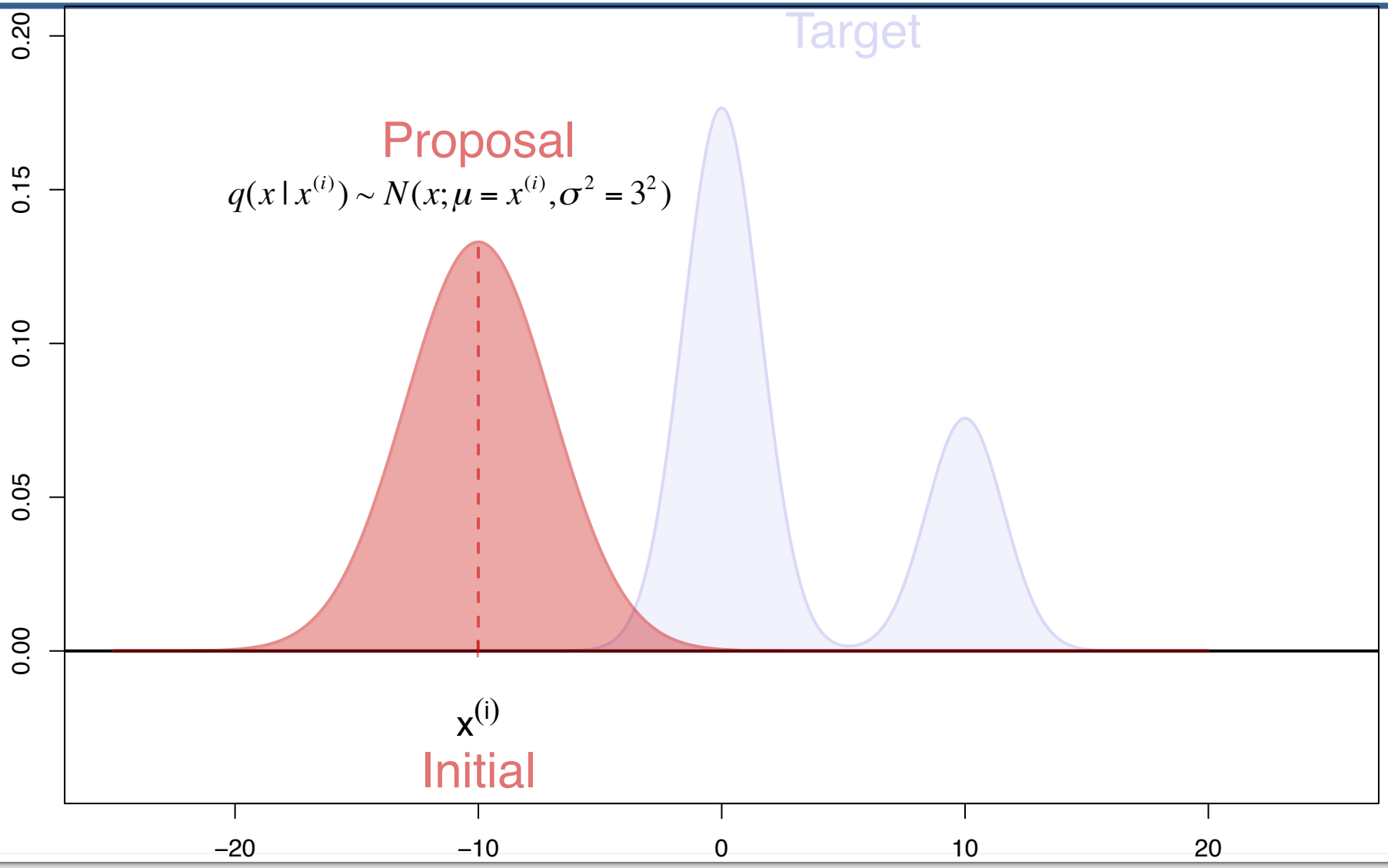
# Metropolis-Hastings Algorithm



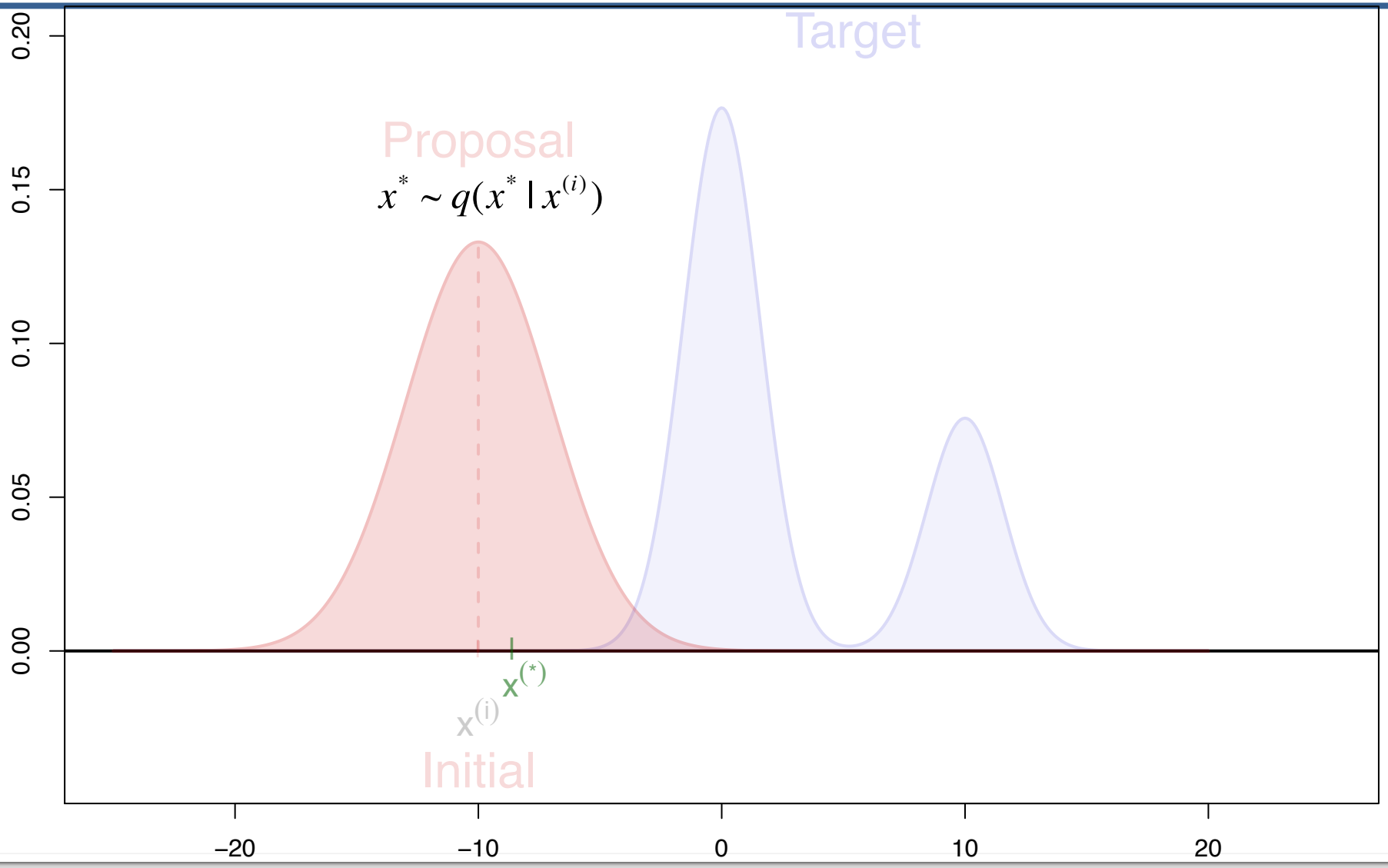
# Metropolis-Hastings Algorithm



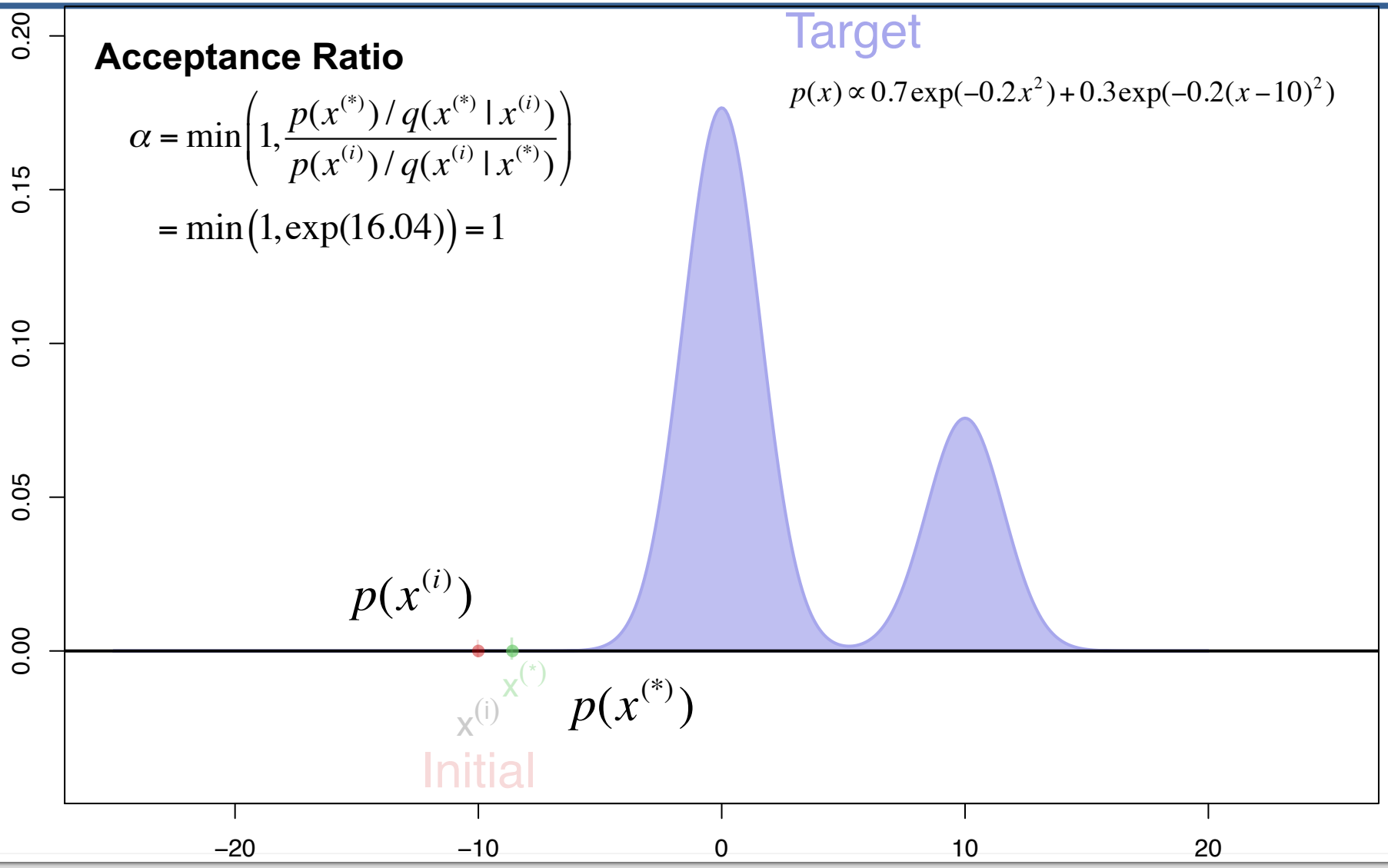
# Metropolis-Hastings Algorithm



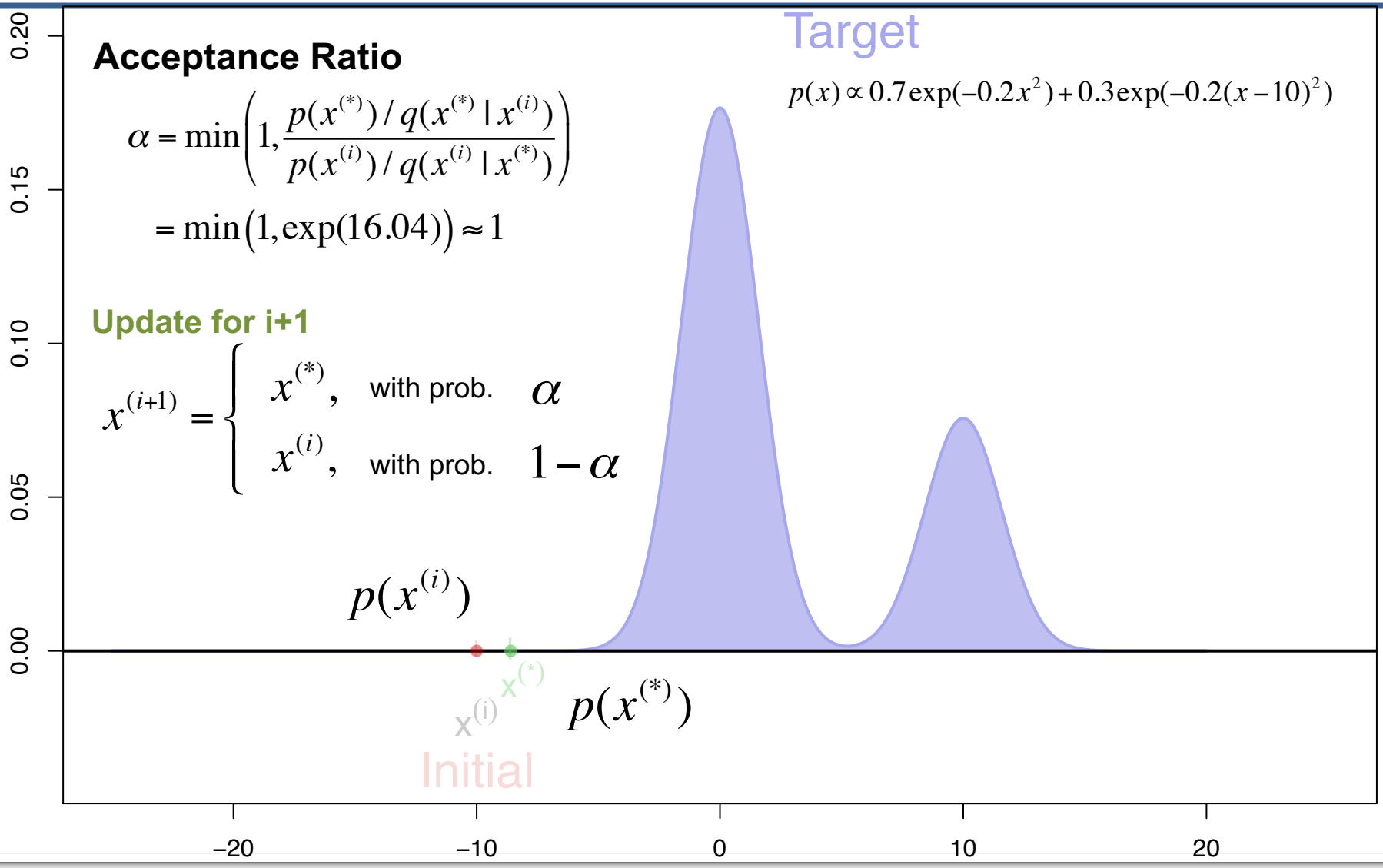
# Metropolis-Hastings Algorithm



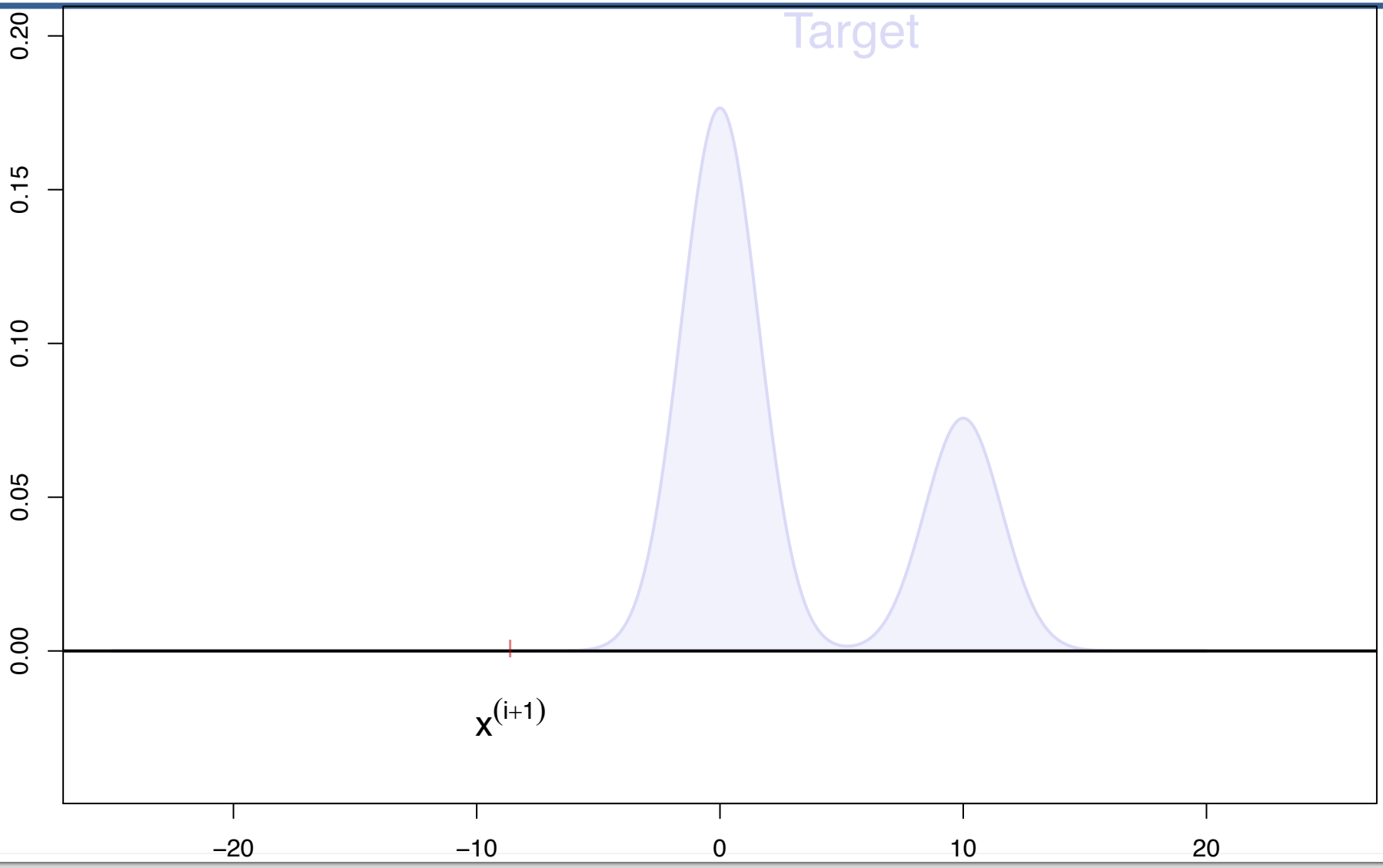
# Metropolis-Hastings Algorithm



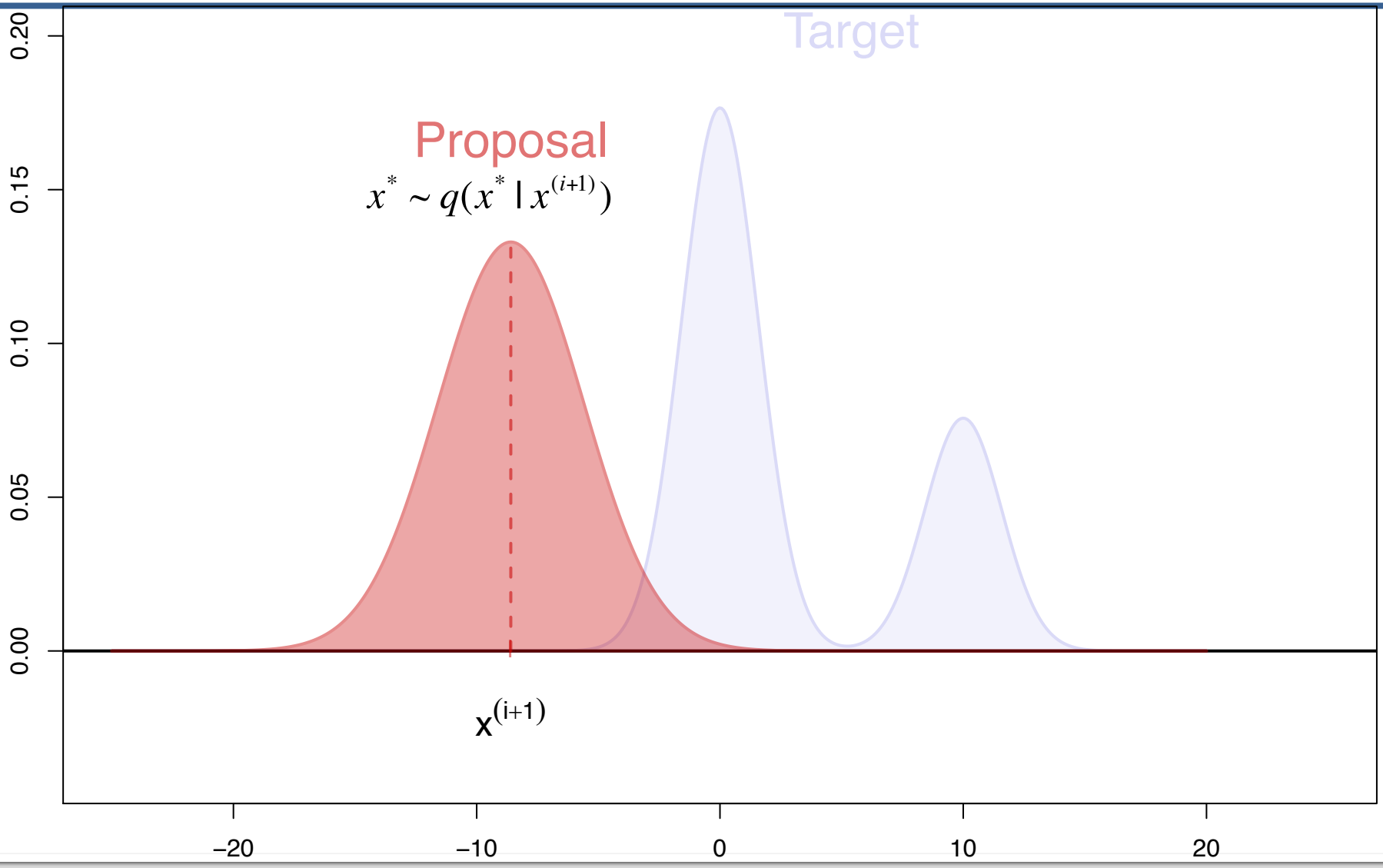
# Metropolis-Hastings Algorithm



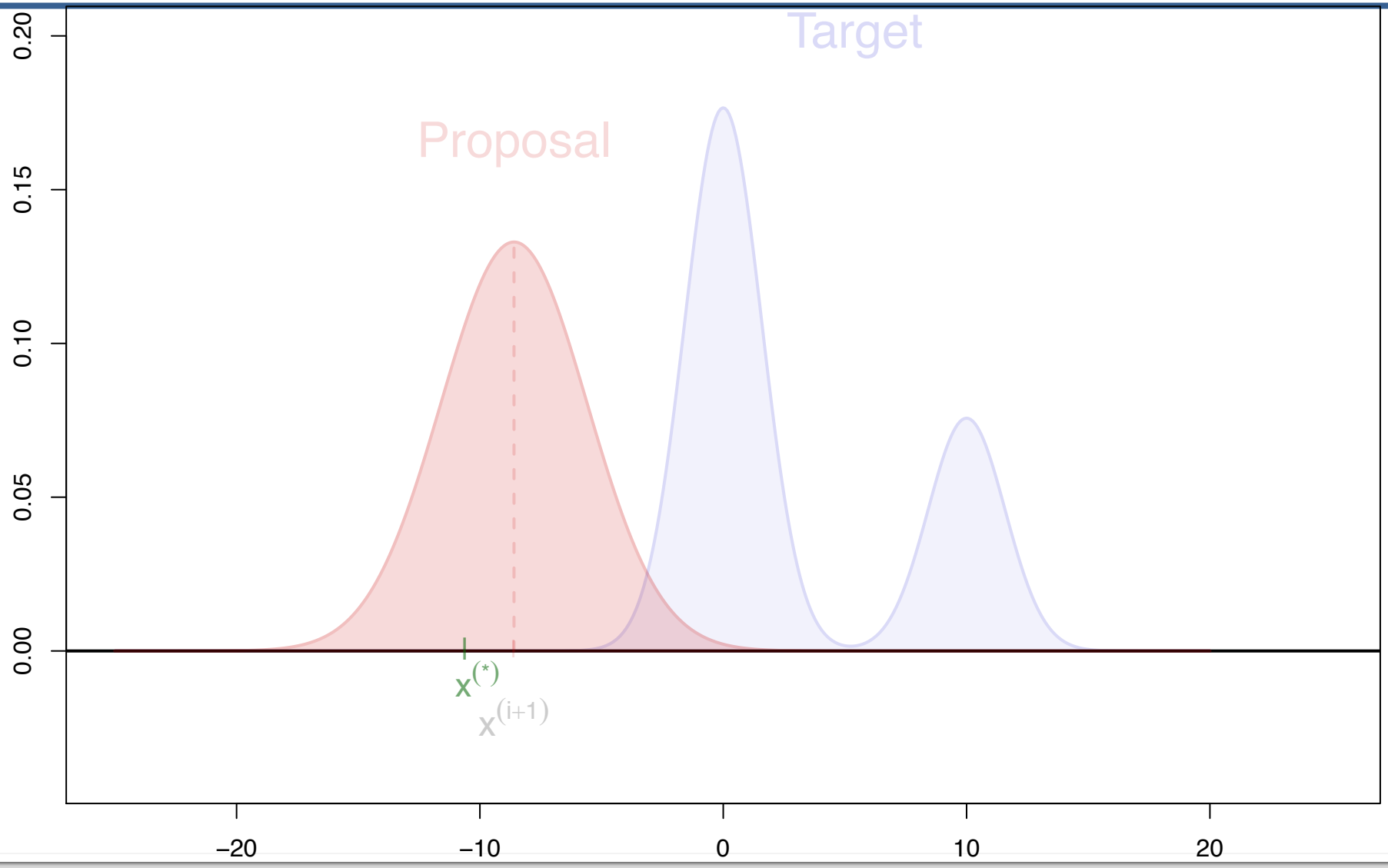
# Metropolis-Hastings Algorithm



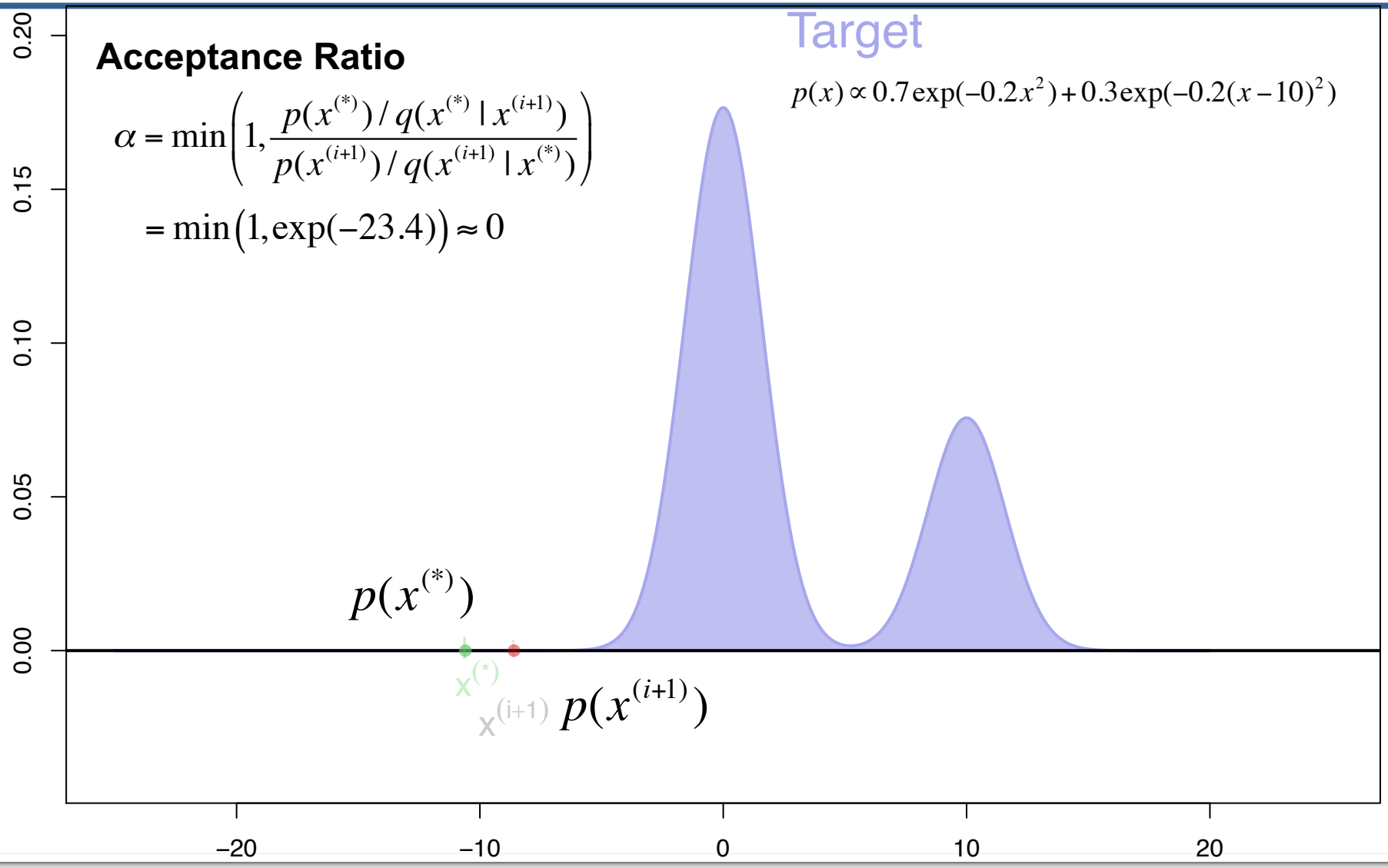
# Metropolis-Hastings Algorithm



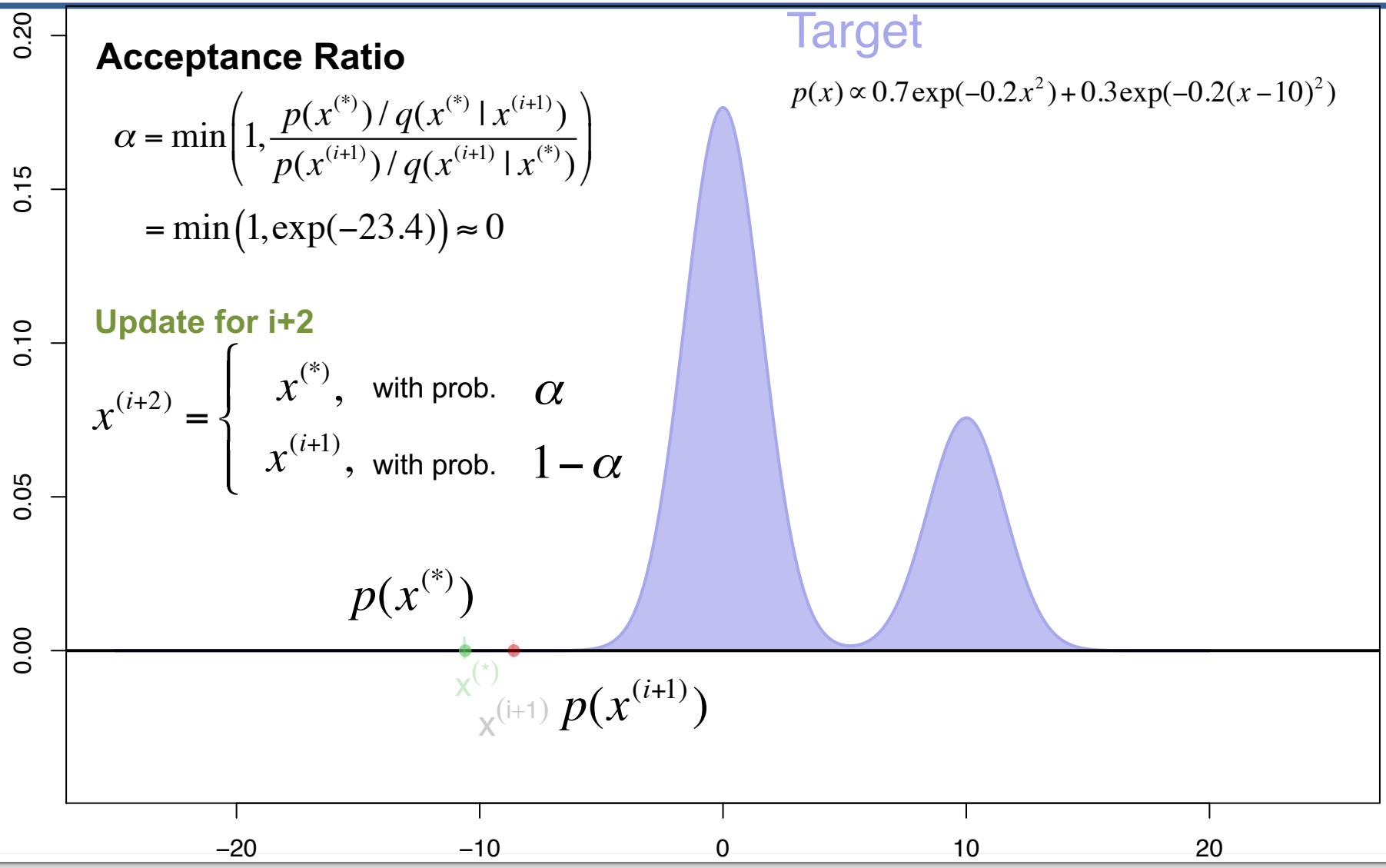
# Metropolis-Hastings Algorithm



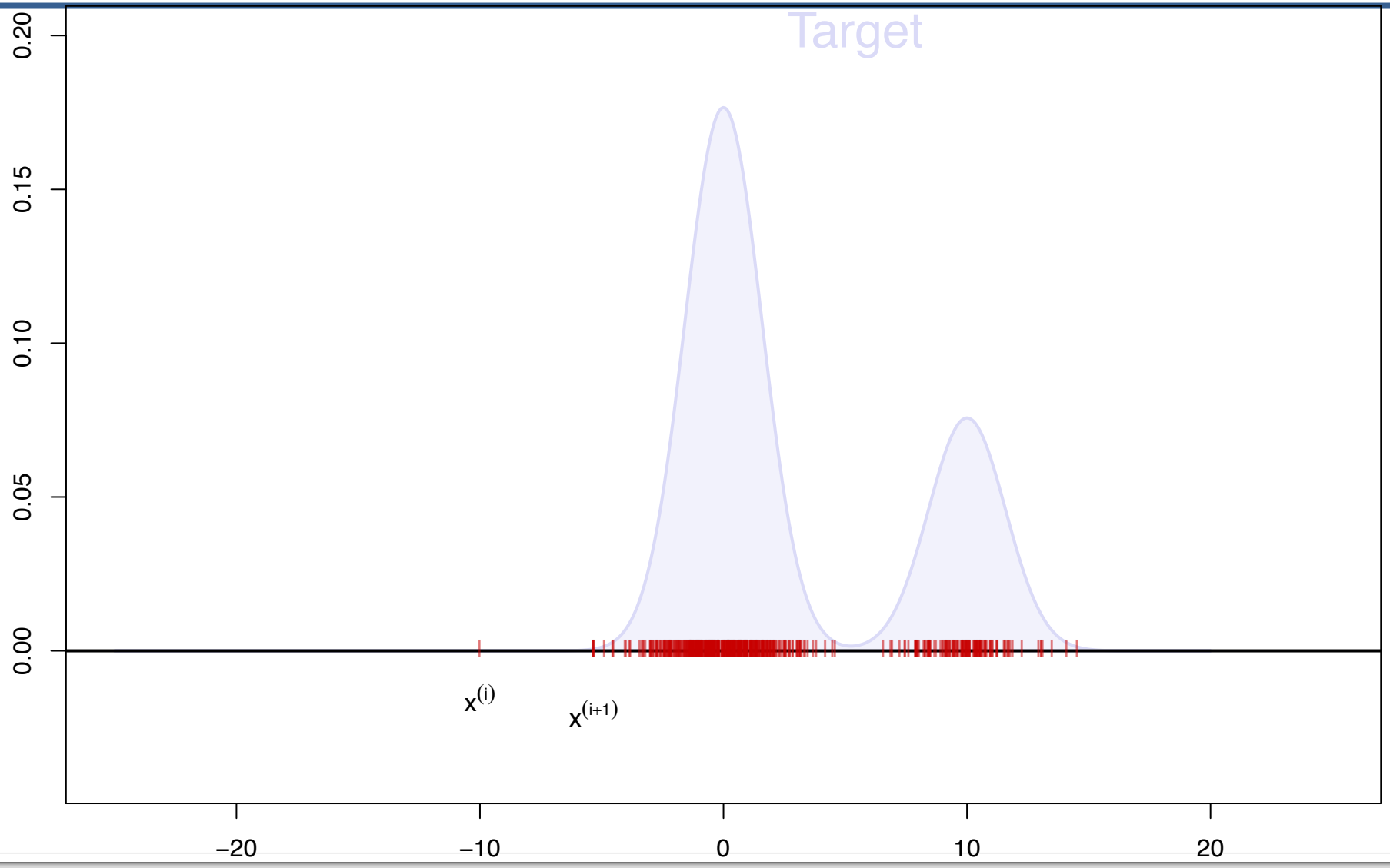
# Metropolis-Hastings Algorithm



# Metropolis-Hastings Algorithm

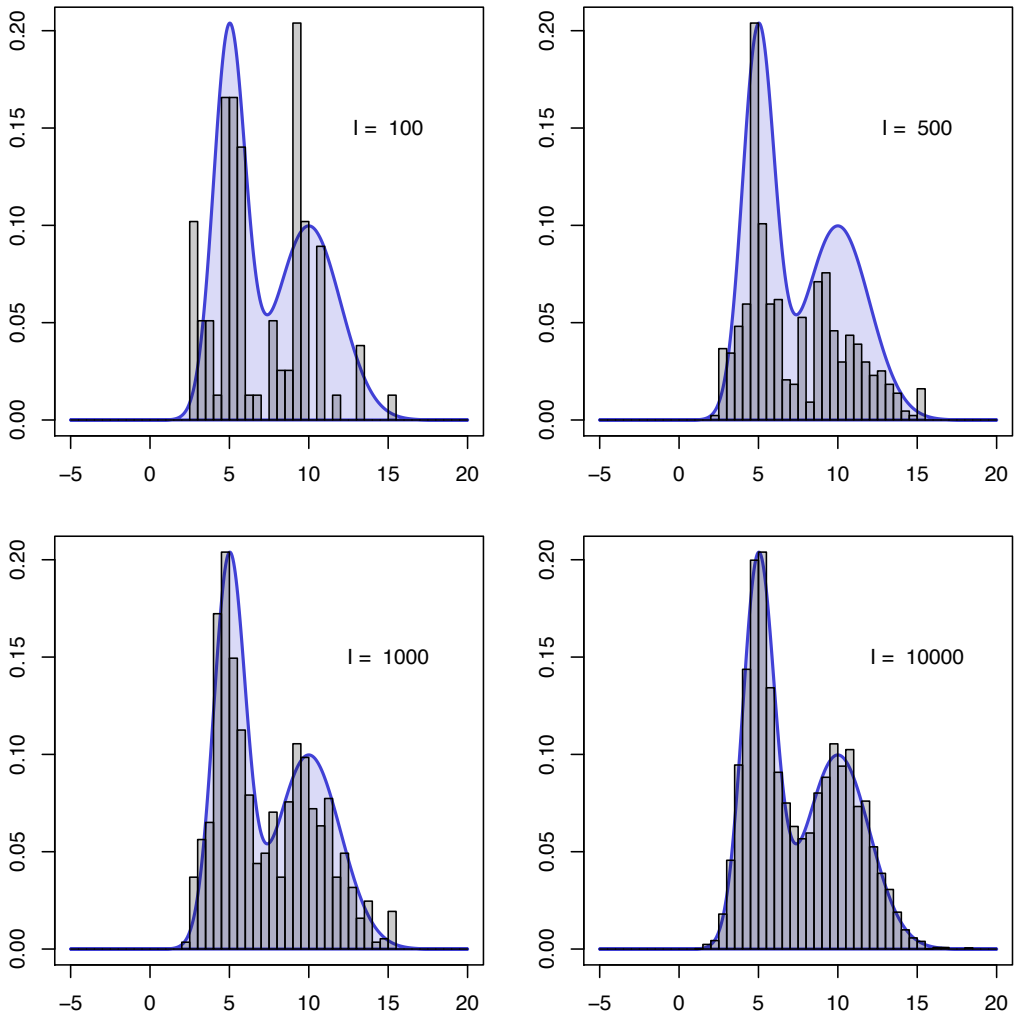


# Metropolis-Hastings Algorithm



# Markov Chain Monte Carlo

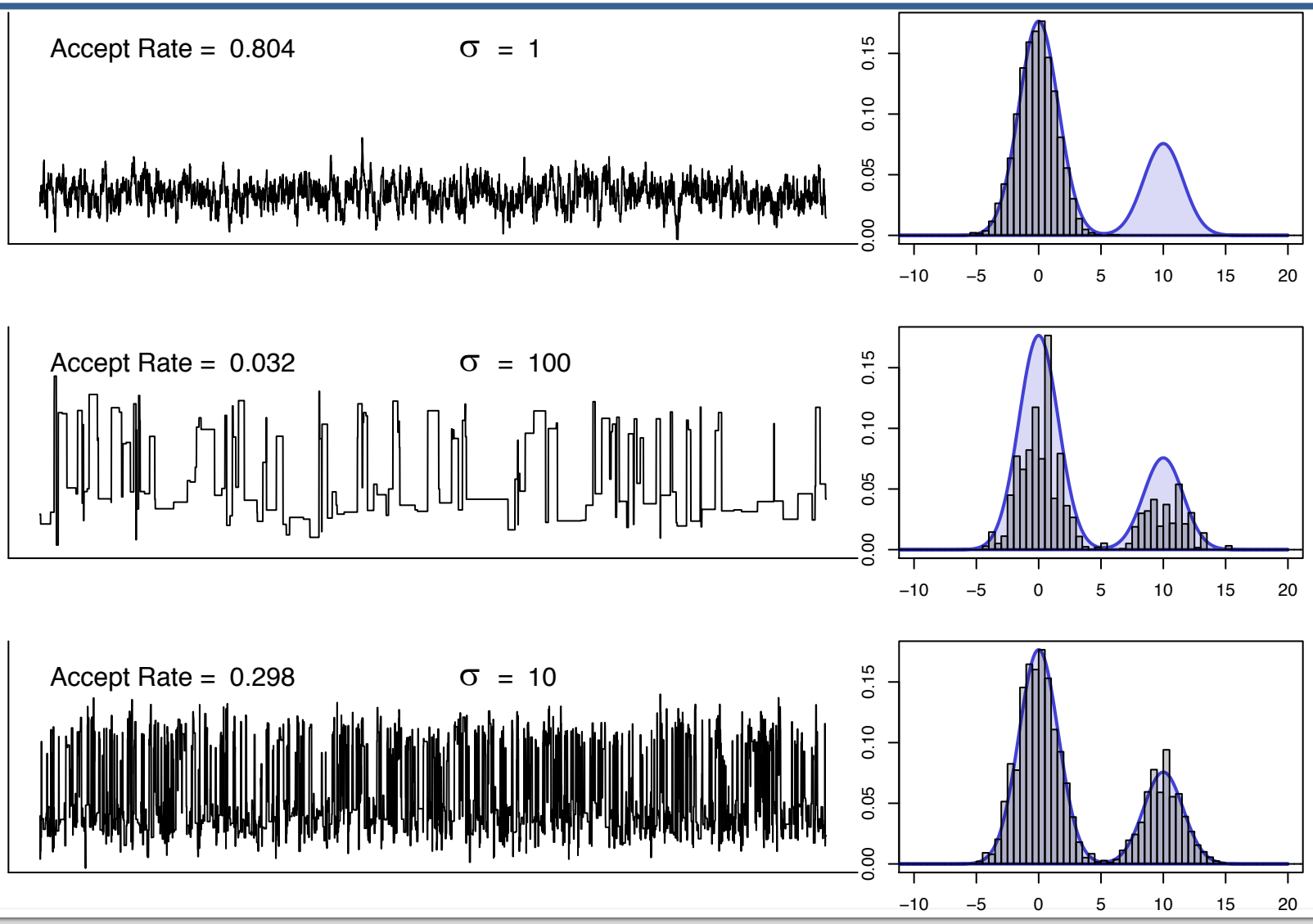
- Main Idea:
  - Create a Markov Chain that has the desired limiting distribution.
- In many cases you can sample 1 million sample chain in a relatively fast time.
- In some cases, estimating the target distribution is too computational expensive.



# Diagnostics: Keep an eye on these

- Proposal Distribution
  - Usually center the proposal around the current state and make ‘local’ moves.
  - Convergence rates will vary depending on the density.
  - Density must explore around the state space.
  - Easy to simulate: don’t make life hard for yourself.
- Convergence
  - Standard Tests: Gelman and Rubin (1992) and Raftery and Lewis (1992)
- Burn-In
  - When do you begin collecting samples?
- Subsampling

# Importance of Proposal Selection

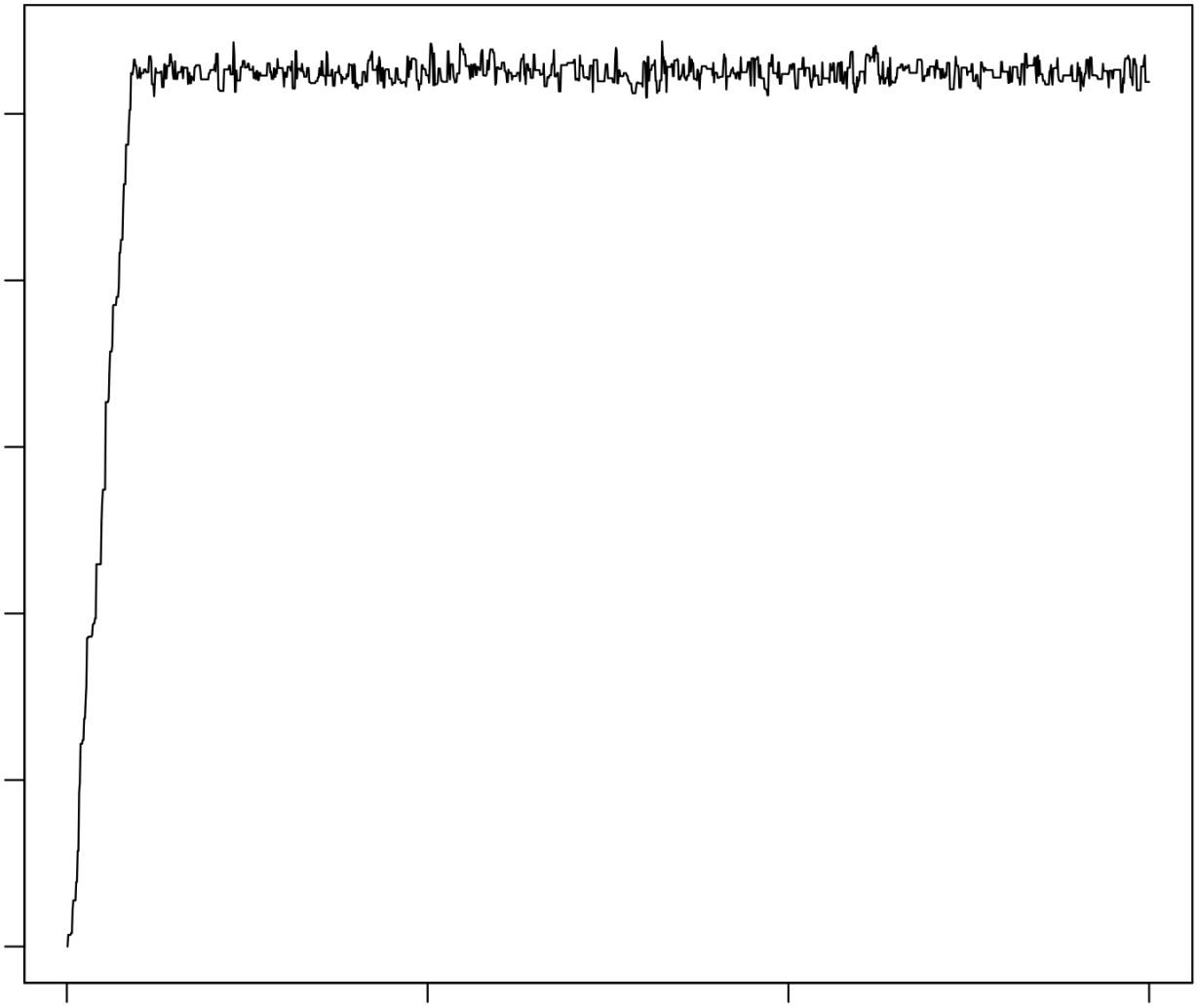


# Burn-In

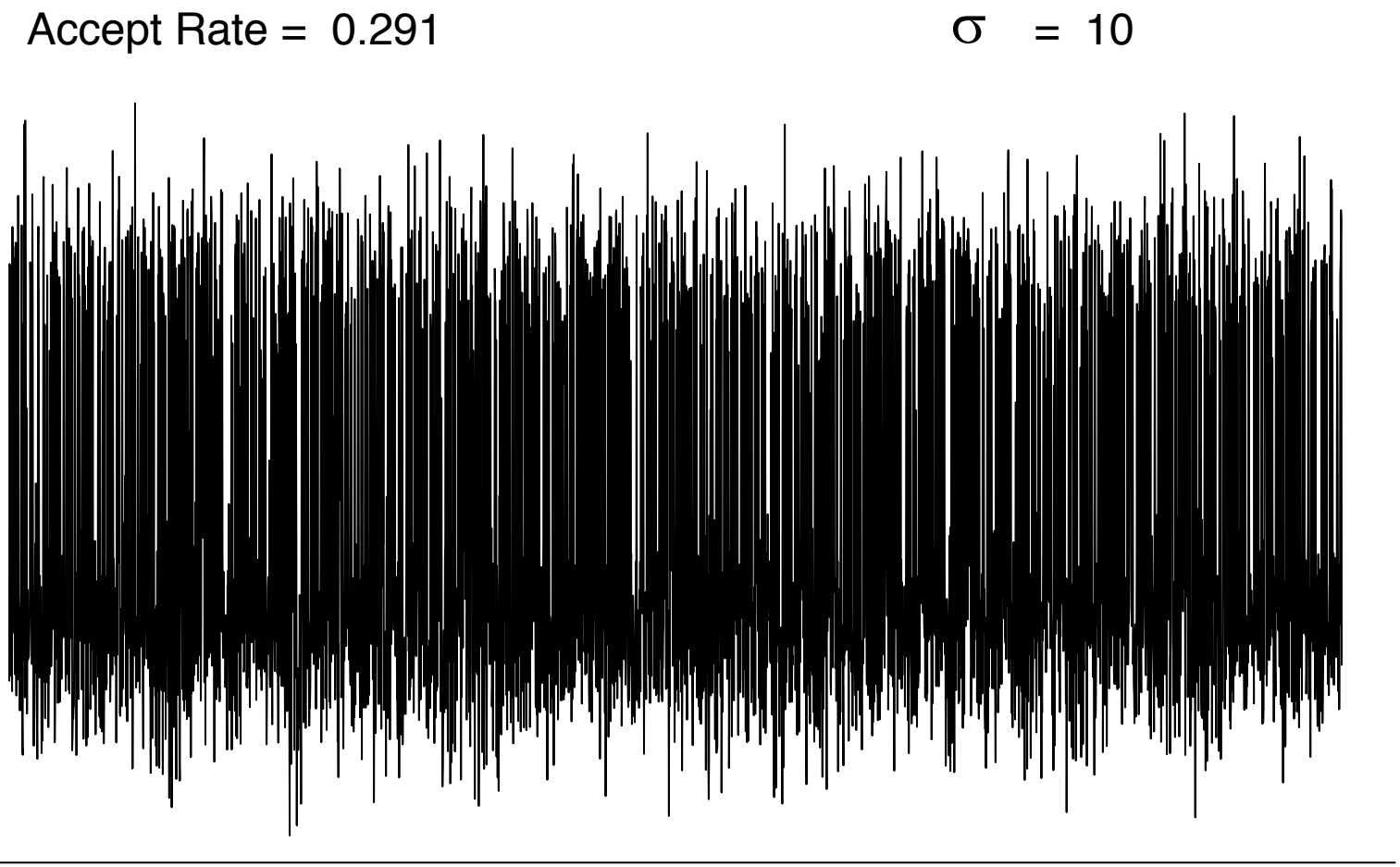
Burn-in refers to the initial n steps of the chain needed for the chain to reach the stationary state.

Burn-in needs to be removed from the final chain.

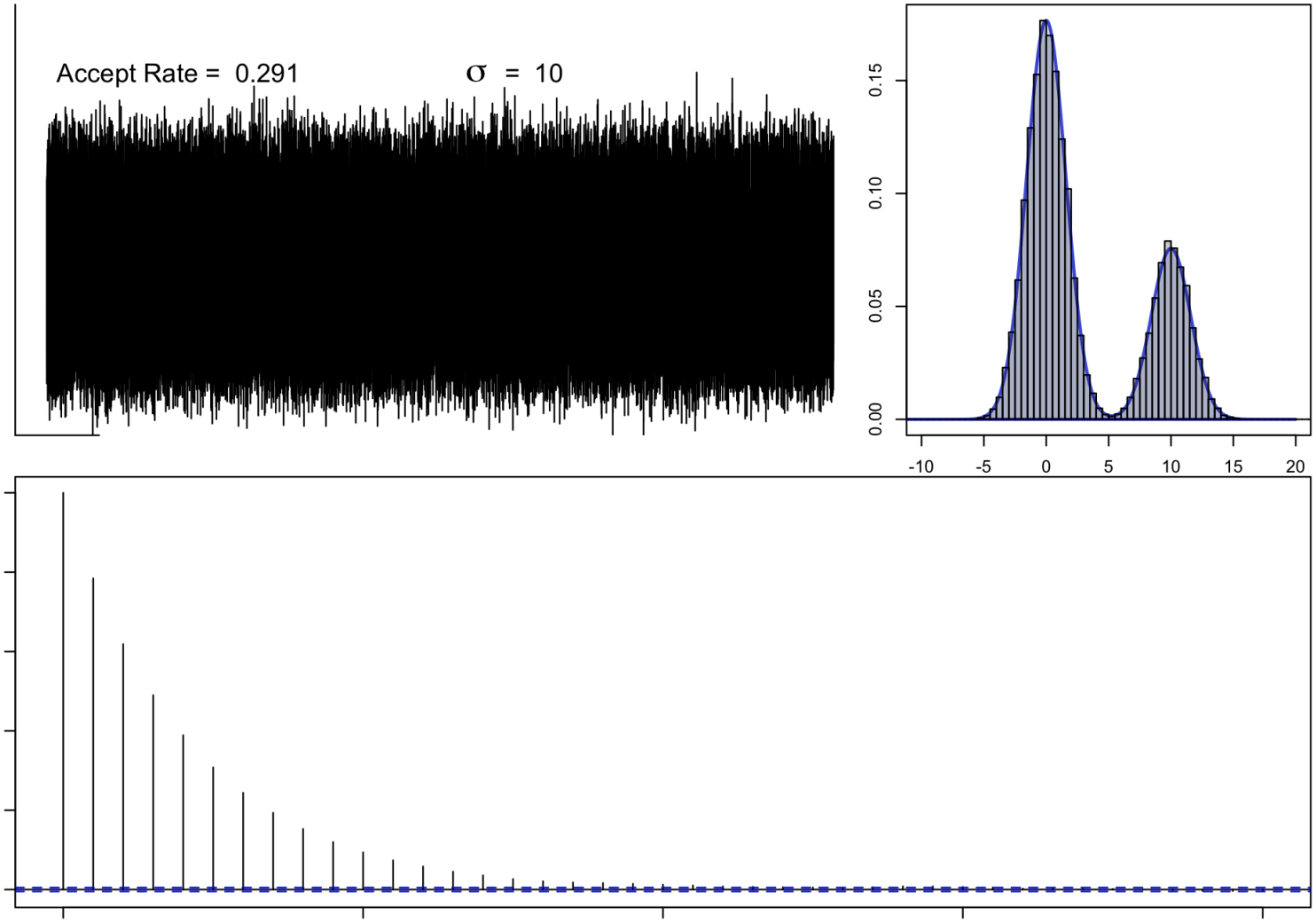
After burn-in you run normally using each iteration.



# Convergence



# Subsampling (Thinning)



# Diagnostics (revisited)

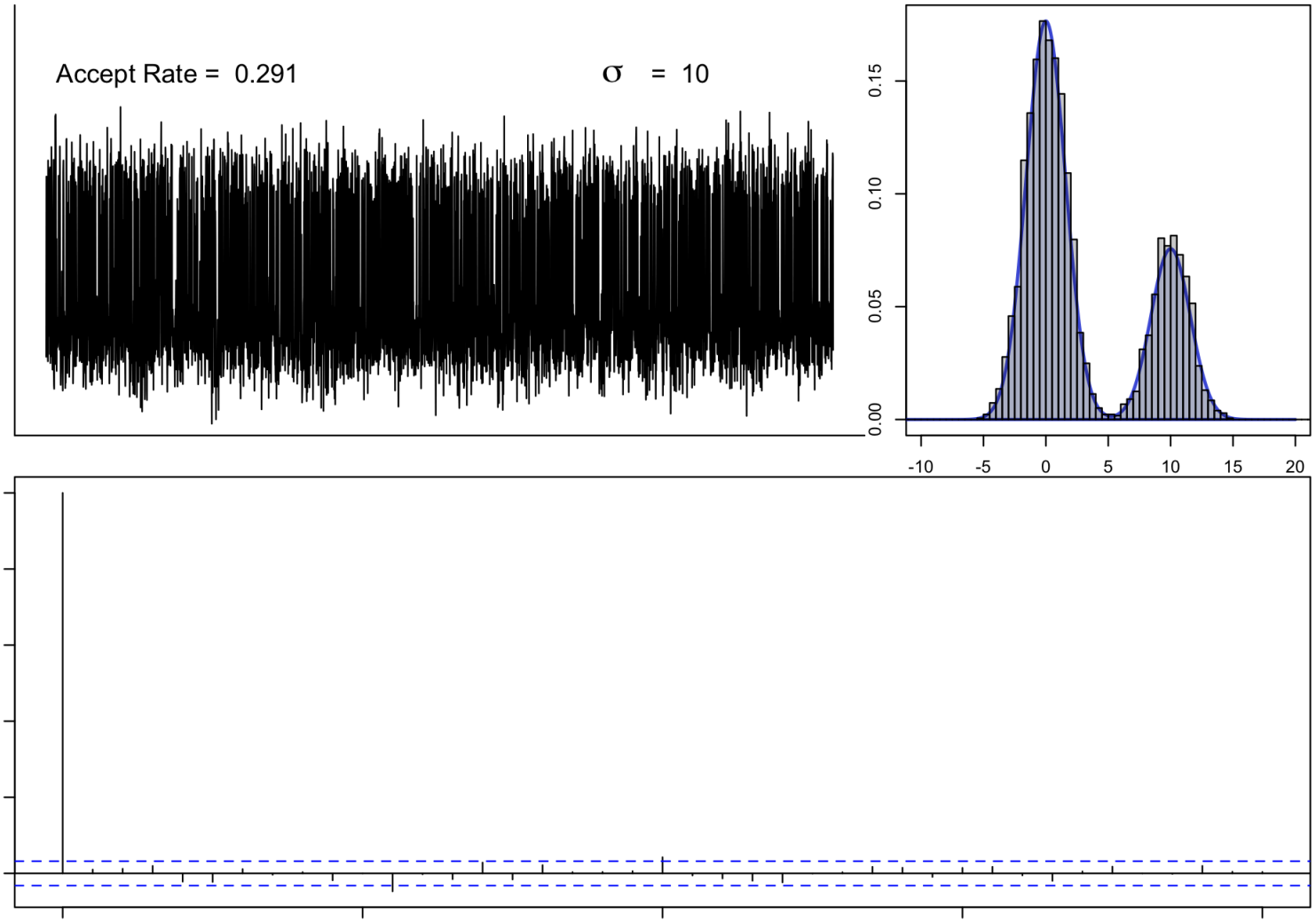
- Proposal Distribution
- Convergence
- Burn-In
- Subsampling

The Metropolis-Hastings algorithm is a general approach to sample from the target density.

REQUIRES user specified proposal density and acceptance rates must monitored.

Not very good for automated models.

# Subsampling (Thinning)



# Gibbs: Special case of Metropolis-Hastings

$$r(x,y) = \frac{h(u^*,v)q(u,v)}{h(u,v)q(v,u^*)} = \frac{g(v)q(v,u^*)q(u,v)}{g(v)q(v,u)q(v,u^*)} = 1$$

- h(u,v) = g(v)q(v,u)
- g(v) is an unnormalized marginal of v
- q(u,v) is the conditional of u|v and is the proposal distribution

- The Proposal is always accepted.

# Bayesian Example: Model

- Let $y = (y_1, \ldots, y_n)'$ be a set of measurements.
- Consider a normal linear regression:

$$y_i = \theta + \epsilon_i$$

  - where $\epsilon_i \sim N(0, \tau^2)$
  - parameter $\tau^2$ and $\mu$ are unknown
- We can write the likelihood as :

$$y = X'\theta + \varepsilon$$

  - where $\varepsilon \sim N_n(0, \tau^2 I_n)$
  - X' is the design matrix

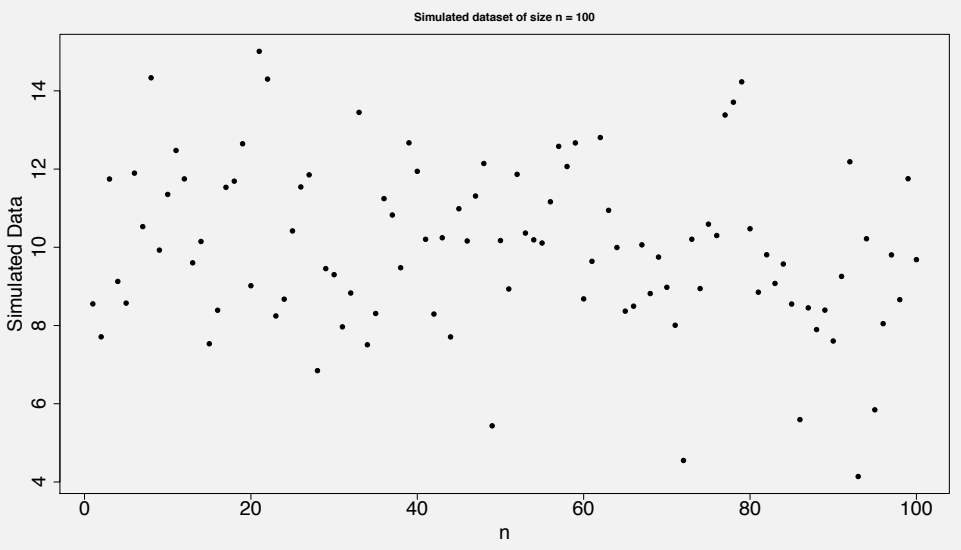
- Priors:

$$\theta \sim N(\mu_\theta, \tau^2 \Omega_\theta)$$

$$\mu_\theta = 0\,,\ \Omega_\theta = 100$$

$$\tau^2 \sim InvGam(\alpha_\tau, \beta_\tau)$$

$$\alpha_\tau = 1\,, \beta_\tau = 1$$



# Bayesian Example: Model

- Bayesian model proceeds by calculating the Joint Posterior distribution of $(\mu, \tau^2)$ conditioned on the data.

$$p(\theta, \tau^2 | \mathrm{y}, \mathrm{X}) \propto \underbrace{p(y|X, \theta, \tau^2)}_{\text{Likelihood}} * \underbrace{\mathrm{p}(\theta|\mu_\theta, \Omega_\theta, \tau^2)}_{\text{Prior}} * \underbrace{p(\tau^2|\alpha_\tau, \beta_\tau)}_{\text{Prior}}$$

- Conditional on the variance parameter:

$$\begin{aligned}
p(\theta|Y, X, \tau^2) &\propto p(Y|\theta, X, \tau^2) p(\theta|\mu_\theta, \tau^2, \Omega_\theta) \\
&\propto (\tau^2)^{-\frac{n}{2}-\frac{k}{2}} |\Omega_\theta|^{-\frac{1}{2}} \exp\left(-\tfrac{1}{\tau^2}\tfrac{1}{2}\left((Y - X'\hat{\theta})'(Y - X'\hat{\theta}) + (\hat{\theta} - \mu_\theta)'\left((XX')^{-1} + \Omega_\theta\right)^{-1}(\hat{\theta} - \mu_\theta)\right)\right) \\
&\times \exp\left(-\tfrac{1}{2}\left(\theta - (XX' + \Omega_\theta^{-1})^{-1}(XX'\hat{\theta} + \Omega_\theta^{-1}\mu_\theta)\right)' \left(\tfrac{1}{\tau^2}(XX' + \Omega_\theta^{-1})\right)\left(\theta - (XX' + \Omega_\theta^{-1})^{-1}(XX'\hat{\theta} + \Omega_\theta^{-1}\mu_\theta)\right)\right) \\
&\sim N\left((XX' + \Omega_\theta^{-1})^{-1}(XX'\hat{\theta} + \Omega_\theta^{-1}\mu_\theta), \tau^2(XX' + \Omega_\theta^{-1})^{-1}\right)
\end{aligned}$$

– where $\hat{\theta} = (XX')^{-1}Xy$

# Bayesian Example: Model

- Bayesian model proceeds by calculating the Joint Posterior distribution of $(\mu, \tau^2)$ conditioned on the data.

$$p(\theta, \tau^2|\mathrm{y}, \mathrm{X}) \propto \underbrace{p(y|X, \theta, \tau^2)}_{\text{Likelihood}} * \underbrace{\mathrm{p}(\theta|\mu_\theta, \Omega_\theta, \tau^2)}_{\text{Prior}} * \underbrace{p(\tau^2|\alpha_\tau, \beta_\tau)}_{\text{Prior}}$$

- Marginalize $\theta$ :

$$p(\tau^2|\mathrm{y}, \mathrm{X}) \propto \int p(y|X, \theta, \tau^2) * \mathrm{p}(\theta|\mu_\theta, \Omega_\theta, \tau^2) * p(\tau^2|\alpha_\tau, \beta_\tau)\, d\theta$$

- Marginal Posterior Distribution:

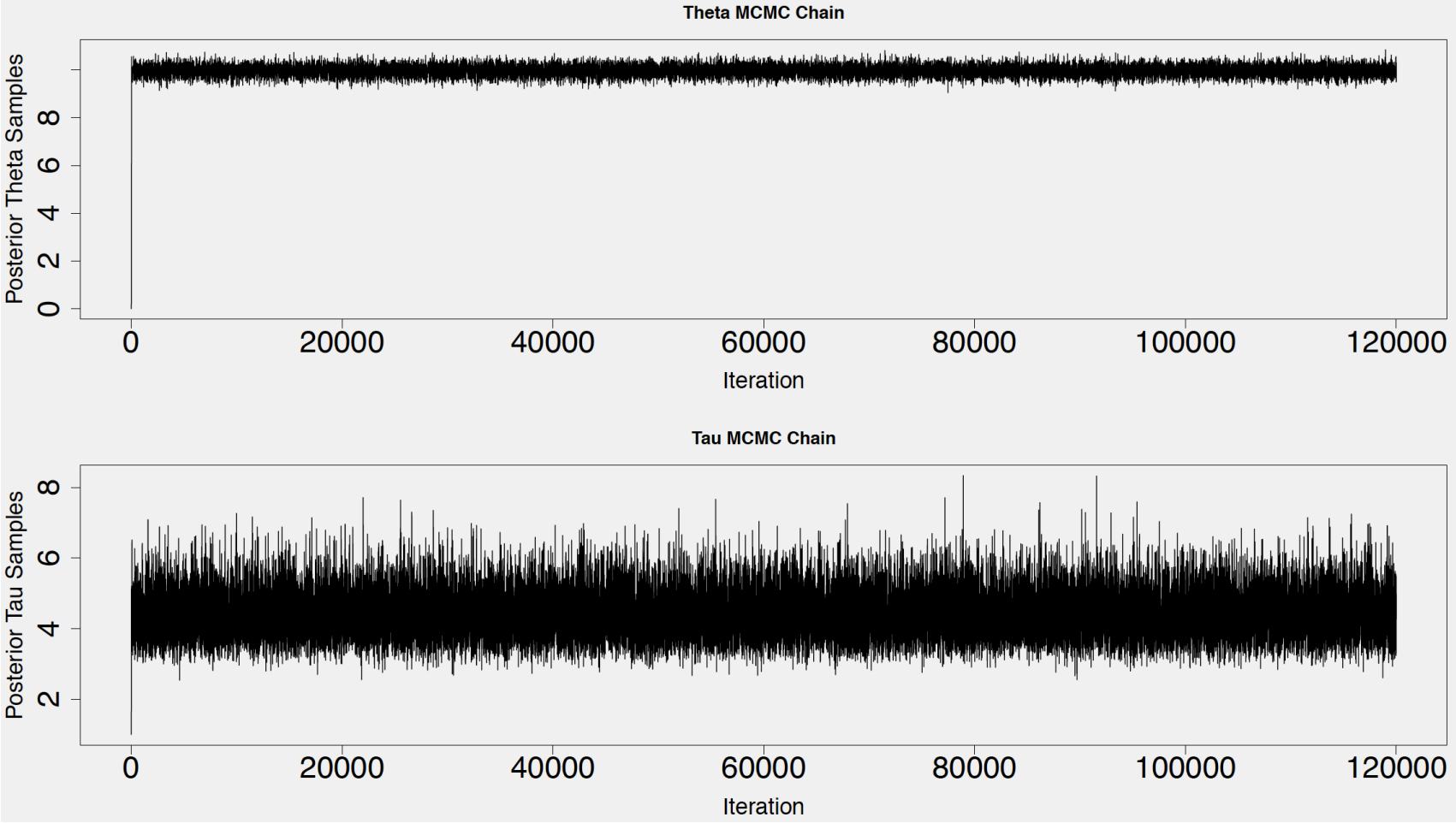
$$\begin{aligned}
p(\tau^2|Y, X) &\propto p(Y|X, \tau^2)p(\tau^2|\alpha_\tau, \beta_\tau) \\
&\propto |\Omega_\theta|^{-\frac{1}{2}}|XX' + \Omega_\theta|^{\frac{1}{2}}(\tau^2)^{-\frac{n}{2}-\alpha_\tau-1} \\
&\times \exp\left(-\tfrac{1}{\tau^2}\tfrac{1}{2}\left((Y - X'\hat{\theta})'(Y - X'\hat{\theta}) + (\hat{\theta} - \mu_\theta)'\left((XX')^{-1} + \Omega_\theta\right)^{-1}(\hat{\theta} - \mu_\theta) + 2\beta_\tau\right)\right) \\
&\sim \Gamma^{-1}\left(\tfrac{n}{2} + \alpha_\tau, \tfrac{1}{2}\left((Y - X'\hat{\theta})'(Y - X'\hat{\theta}) + (\hat{\theta} - \mu_\theta)'\left((XX')^{-1} + \Omega_\theta\right)^{-1}(\hat{\theta} - \mu_\theta) + 2\beta_\tau\right)\right)
\end{aligned}$$

# MCMC Algorithm

- We ran the algorithm for 120K iterations.
- Run time: 6 seconds
- Burn-in size: 20K
- Normal Proposal distribution for $\theta$
  - $\theta^* \sim N\left(\theta^{(i)}, 1\right)$
- Normal Proposal distribution for $\tau^2$
  - $\tau^{2^*} \sim N(\tau^{2^{(i)}}, 2)$
- $\theta$ Acceptance rate = 0.252
- $\tau^2$ Acceptance rate = 0.347

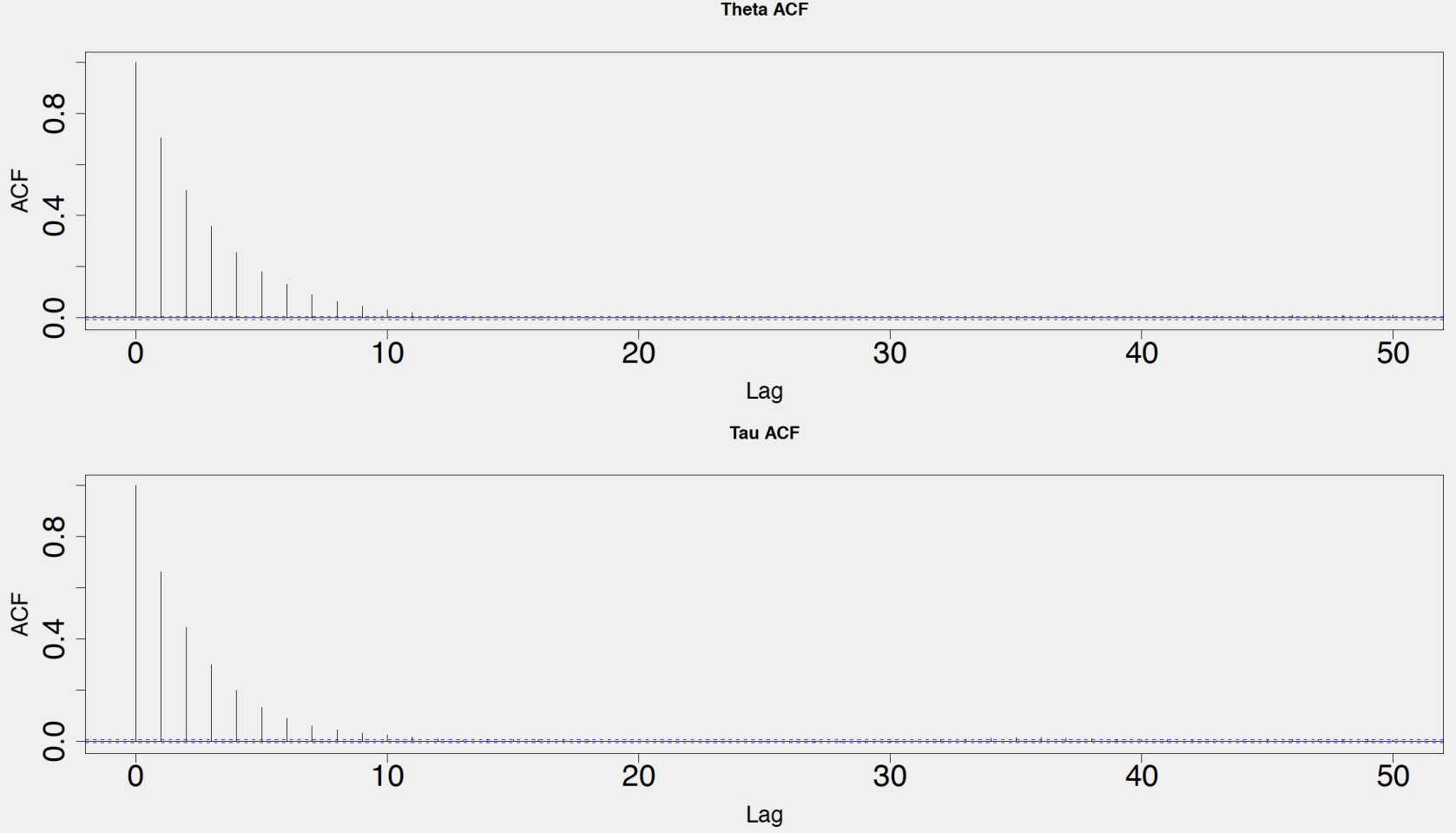
# MCMC Chain: Raw

Initialization: $\theta^{(1)} = 0\,, \tau^{2^{(1)}} = 1$



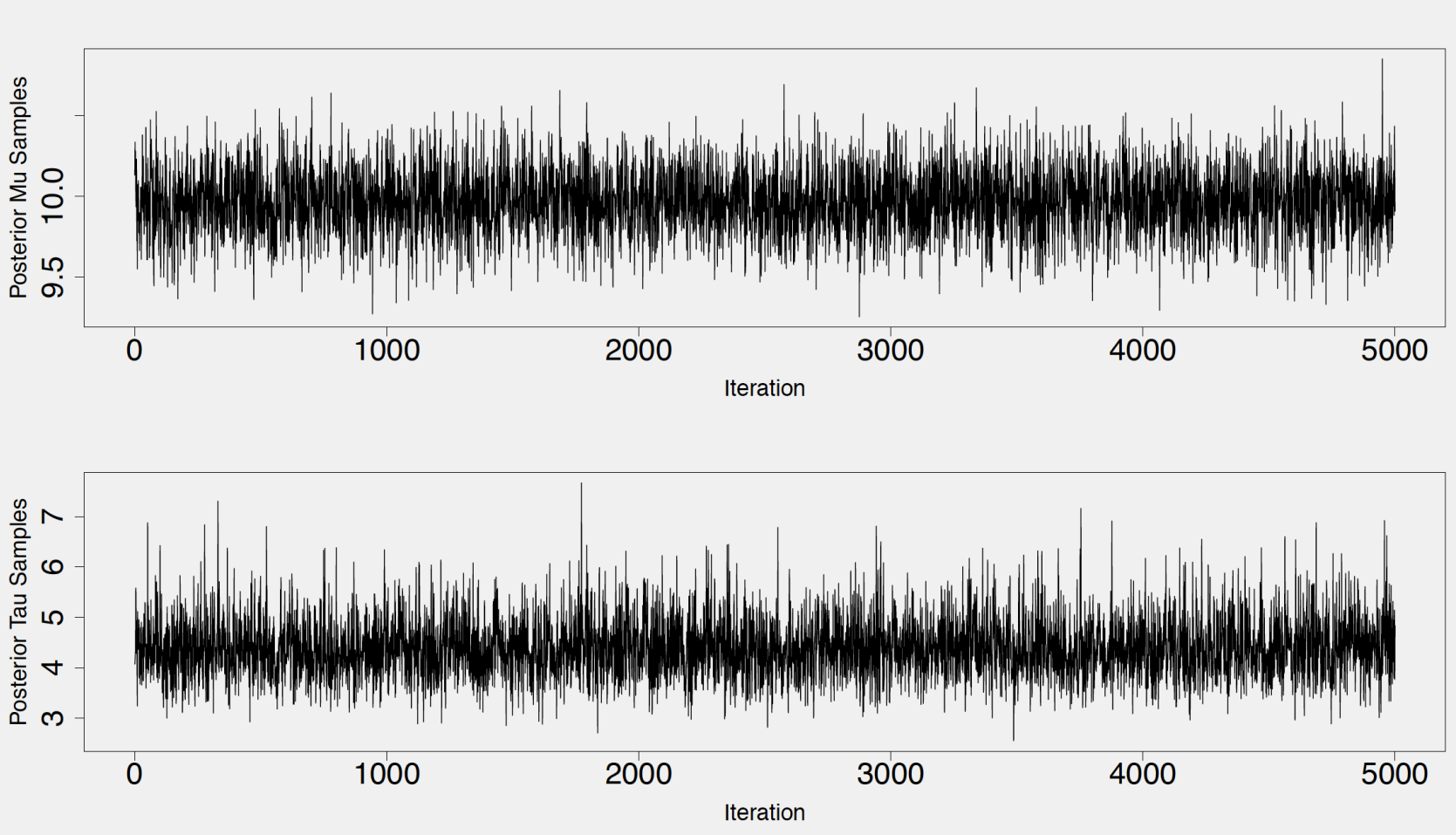
# MCMC Chain: ACF

The Auto-correlation shows a lag of about 12. For this example we take every 20.

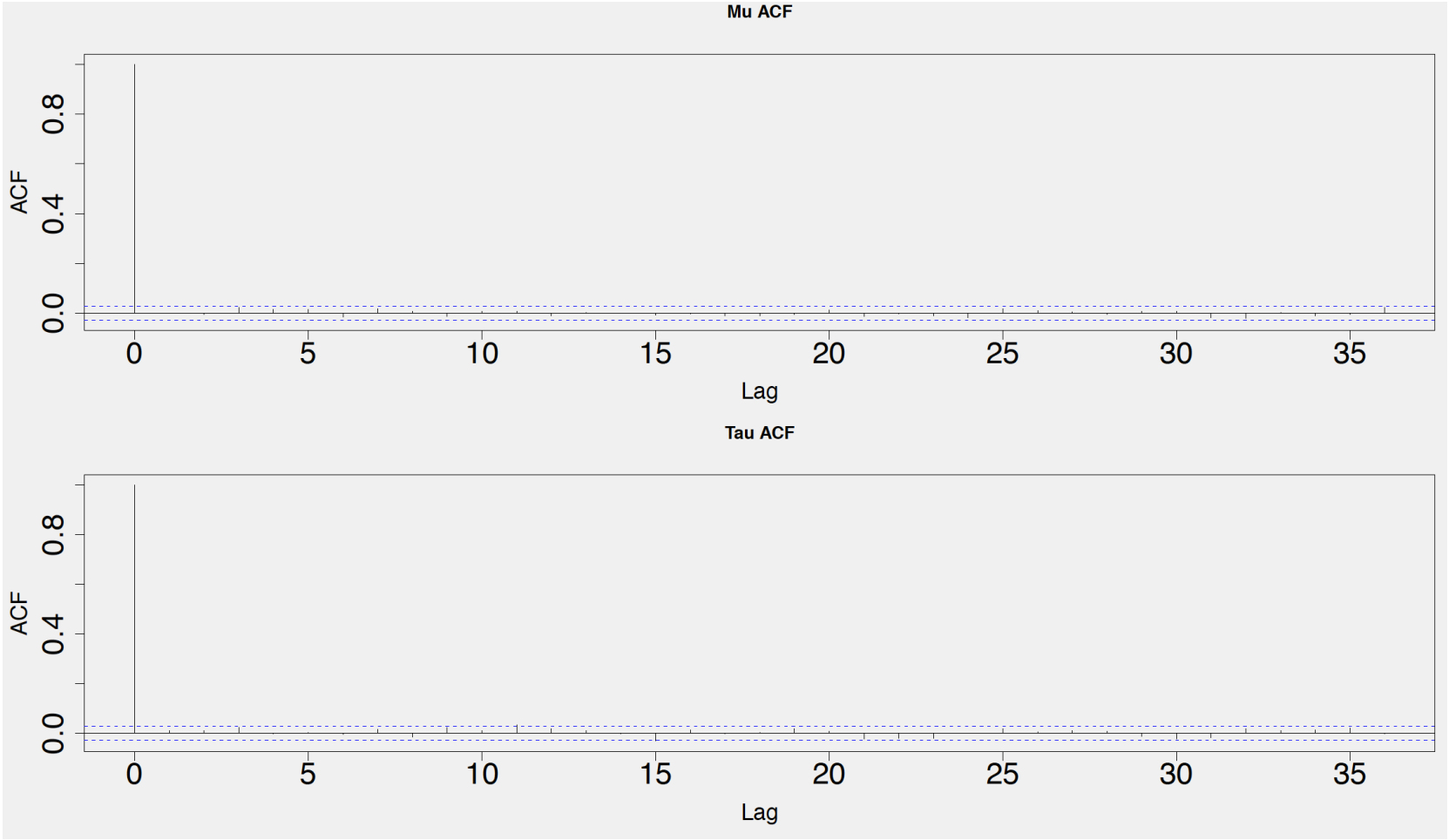


# MCMC Chain: Thinned

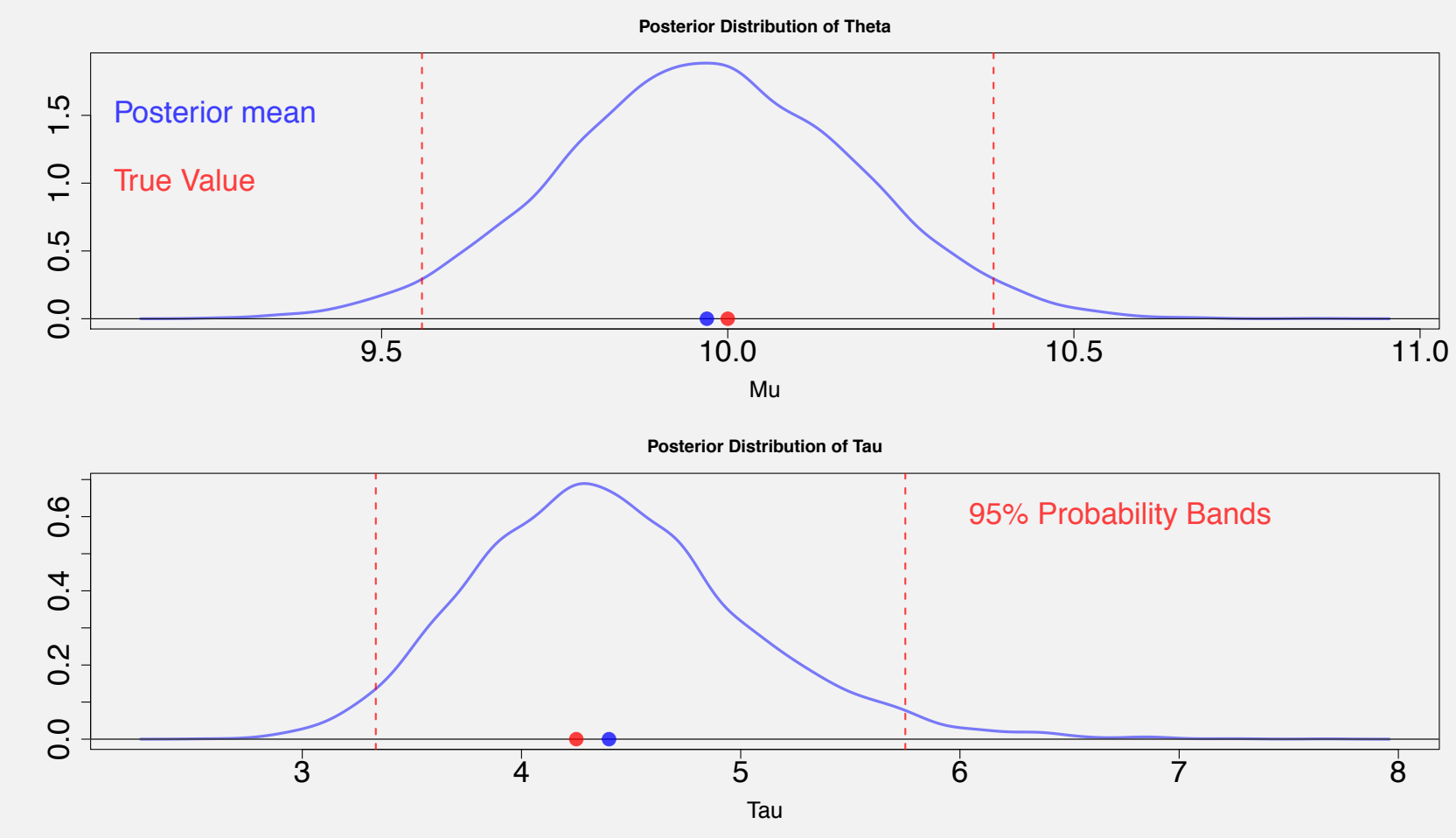
We have a sample of 5000 posterior estimates.



# MCMC Chain: Thinned ACF



# Estimates v Truth



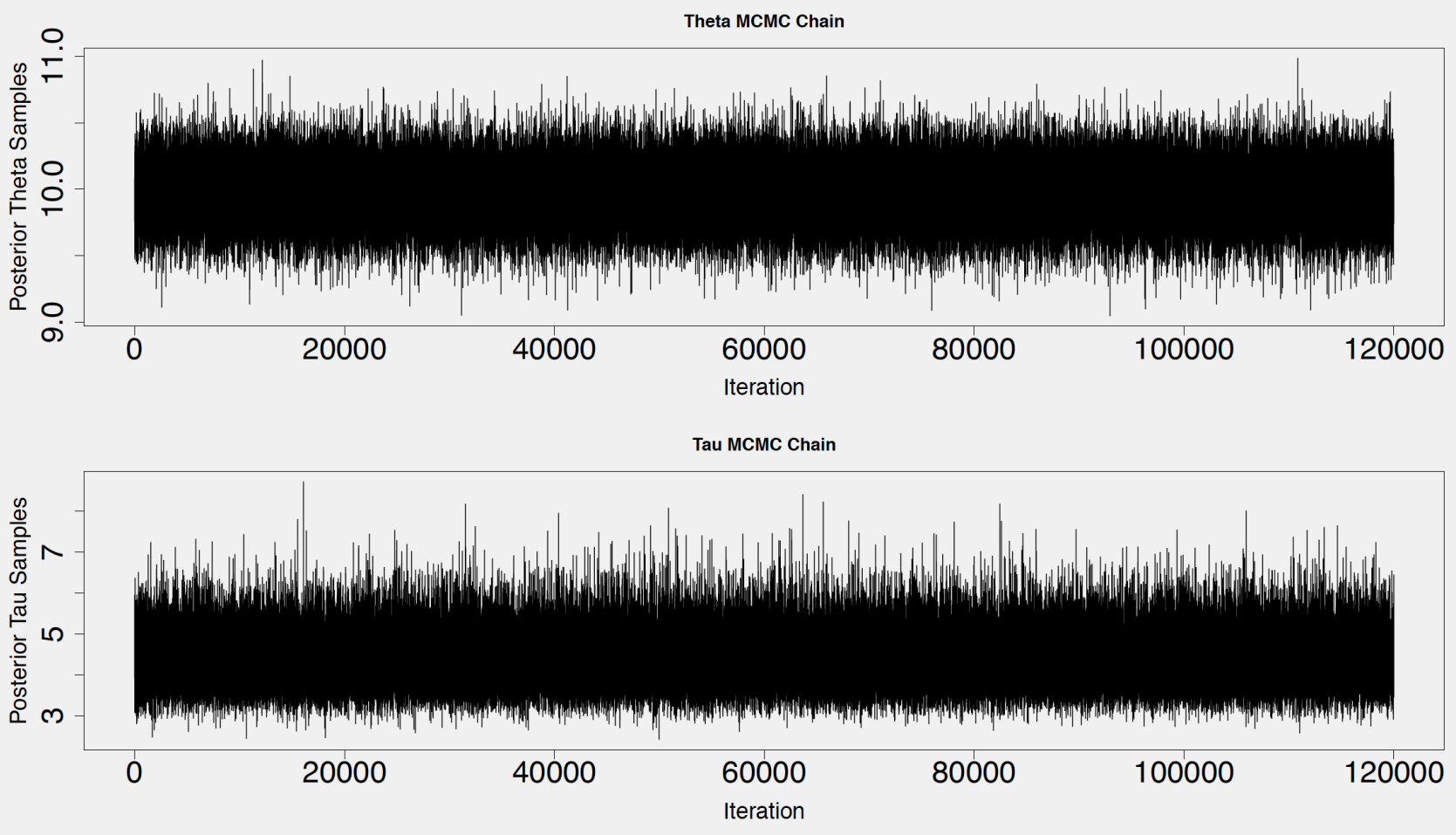
# Same Model via Gibbs Step

- Sample straight from the posterior distribution.
- Recall that:

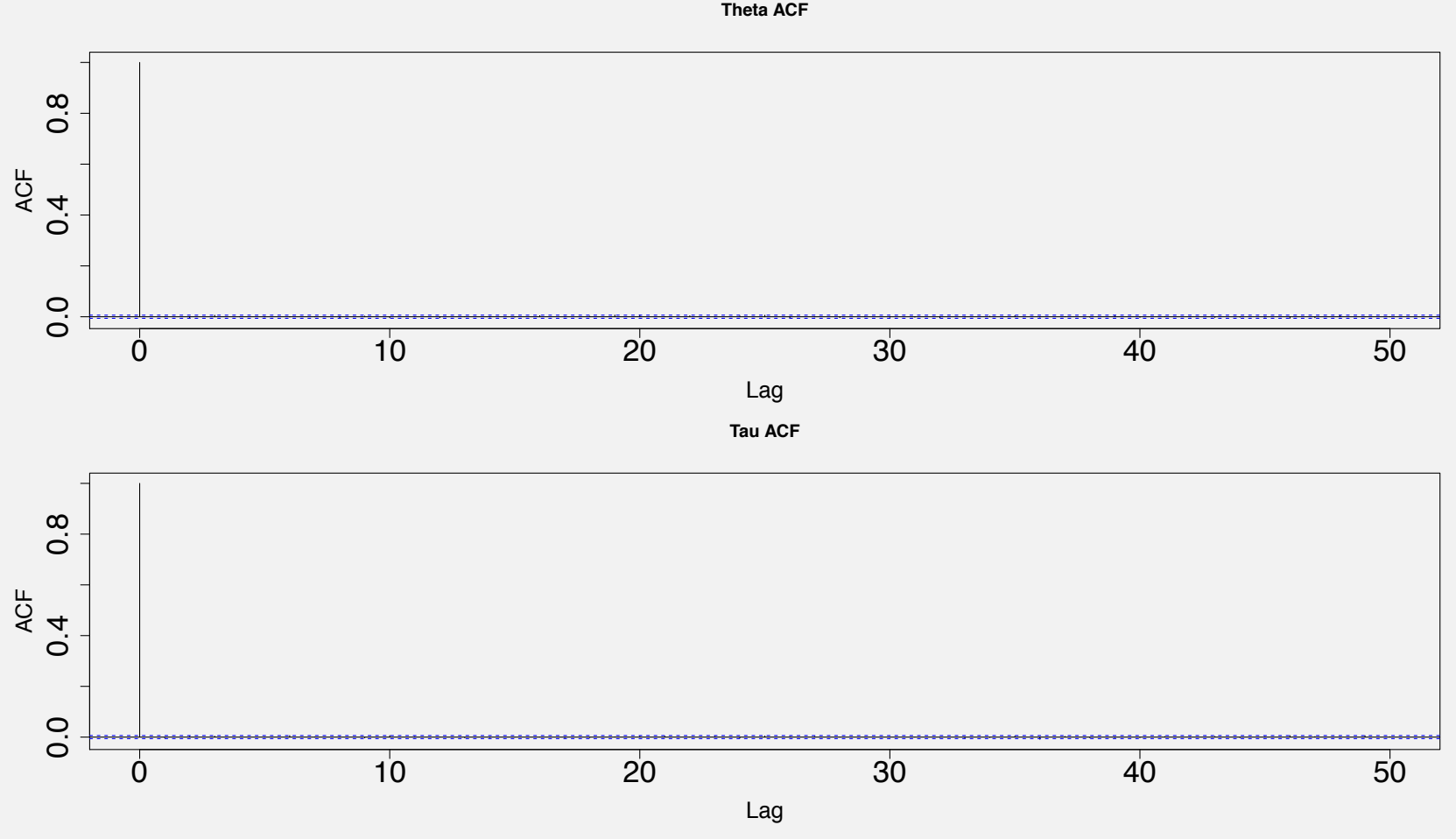
$$
\begin{aligned}
p(\tau^2|Y,X) &\propto p(Y|X,\tau^2)p(\tau^2|\alpha_\tau,\beta_\tau) \\
&\propto |\Omega_\theta|^{-\frac{1}{2}}|XX'+\Omega_\theta|^{\frac{1}{2}}(\tau^2)^{-\frac{n}{2}-\alpha_\tau-1} \\
&\times \exp\left(-\tfrac{1}{\tau^2}\tfrac{1}{2}\left((Y-X'\hat{\theta})'(Y-X'\hat{\theta})+(\hat{\theta}-\mu_\theta)'\left((XX')^{-1}+\Omega_\theta\right)^{-1}(\hat{\theta}-\mu_\theta)+2\beta_\tau\right)\right) \\
&\sim \Gamma^{-1}\left(\tfrac{n}{2}+\alpha_\tau,\tfrac{1}{2}\left((Y-X'\hat{\theta})'(Y-X'\hat{\theta})+(\hat{\theta}-\mu_\theta)'\left((XX')^{-1}+\Omega_\theta\right)^{-1}(\hat{\theta}-\mu_\theta)+2\beta_\tau\right)\right)
\end{aligned}
$$

$$
\begin{aligned}
p(\theta|Y,X,\tau^2) &\propto p(Y|\theta,X,\tau^2)p(\theta|\mu_\theta,\tau^2,\Omega_\theta) \\
&\propto (\tau^2)^{-\frac{n}{2}-\frac{k}{2}}|\Omega_\theta|^{-\frac{1}{2}}\exp\left(-\tfrac{1}{\tau^2}\tfrac{1}{2}\left((Y-X'\hat{\theta})'(Y-X'\hat{\theta})+(\hat{\theta}-\mu_\theta)'\left((XX')^{-1}+\Omega_\theta\right)^{-1}(\hat{\theta}-\mu_\theta)\right)\right) \\
&\times \exp\left(-\tfrac{1}{2}\left(\theta-(XX'+\Omega_\theta^{-1})^{-1}(XX'\hat{\theta}+\Omega_\theta^{-1}\mu_\theta)\right)'(\tfrac{1}{\tau^2}(XX'+\Omega_\theta^{-1}))\left(\theta-(XX'+\Omega_\theta^{-1})^{-1}(XX'\hat{\theta}+\Omega_\theta^{-1}\mu_\theta)\right)\right) \\
&\sim N\left((XX'+\Omega_\theta^{-1})^{-1}(XX'\hat{\theta}+\Omega_\theta^{-1}\mu_\theta),\tau^2(XX'+\Omega_\theta^{-1})^{-1}\right)
\end{aligned}
$$

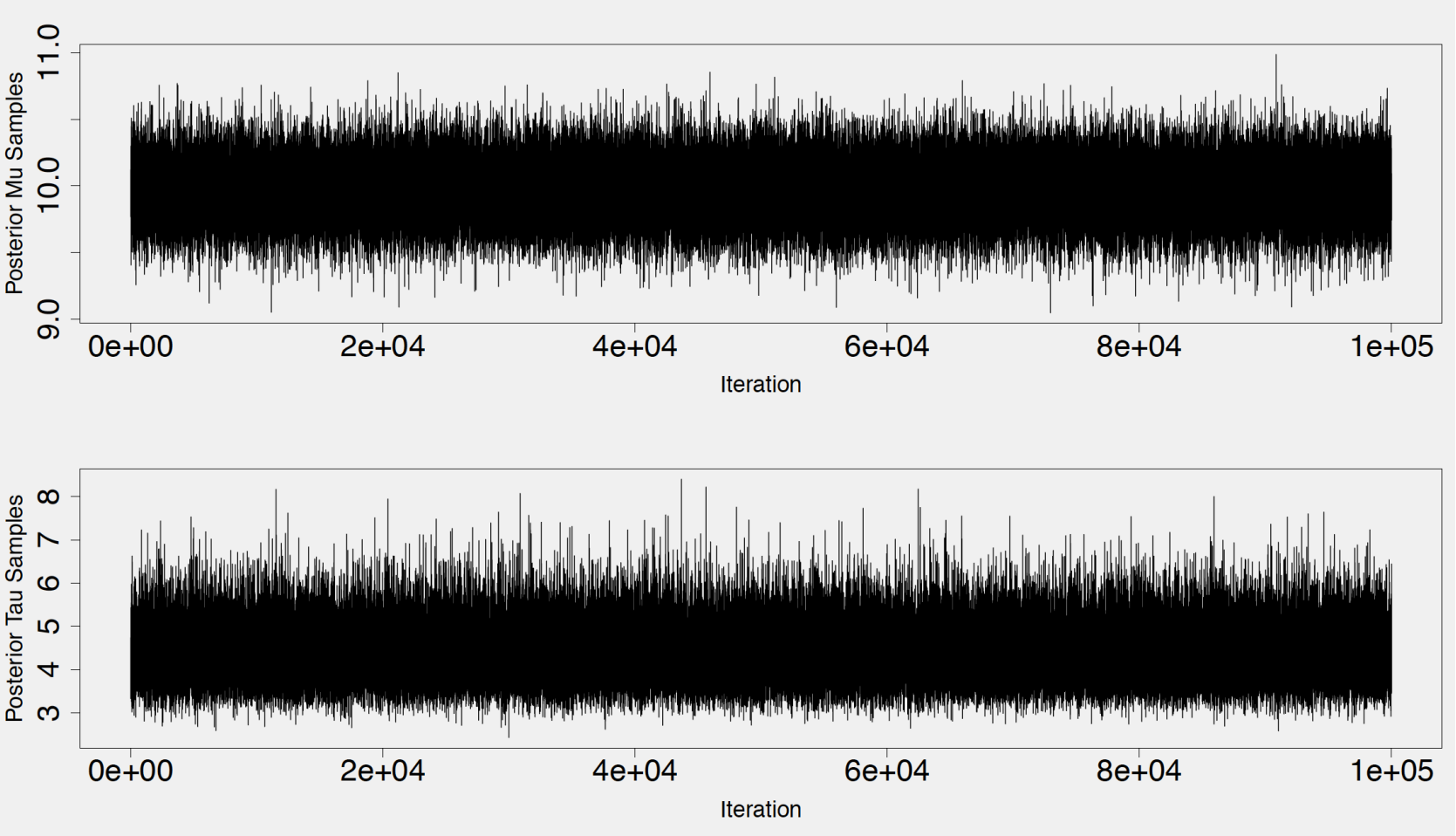
# Gibbs: Raw


Theta MCMC Chain
Posterior Theta Samples
9.0
10.0
11.0
0
20000
40000
60000
80000
100000
120000
Iteration
Tau MCMC Chain
Posterior Tau Samples
3
5
7
0
20000
40000
60000
80000
100000
120000
Iteration

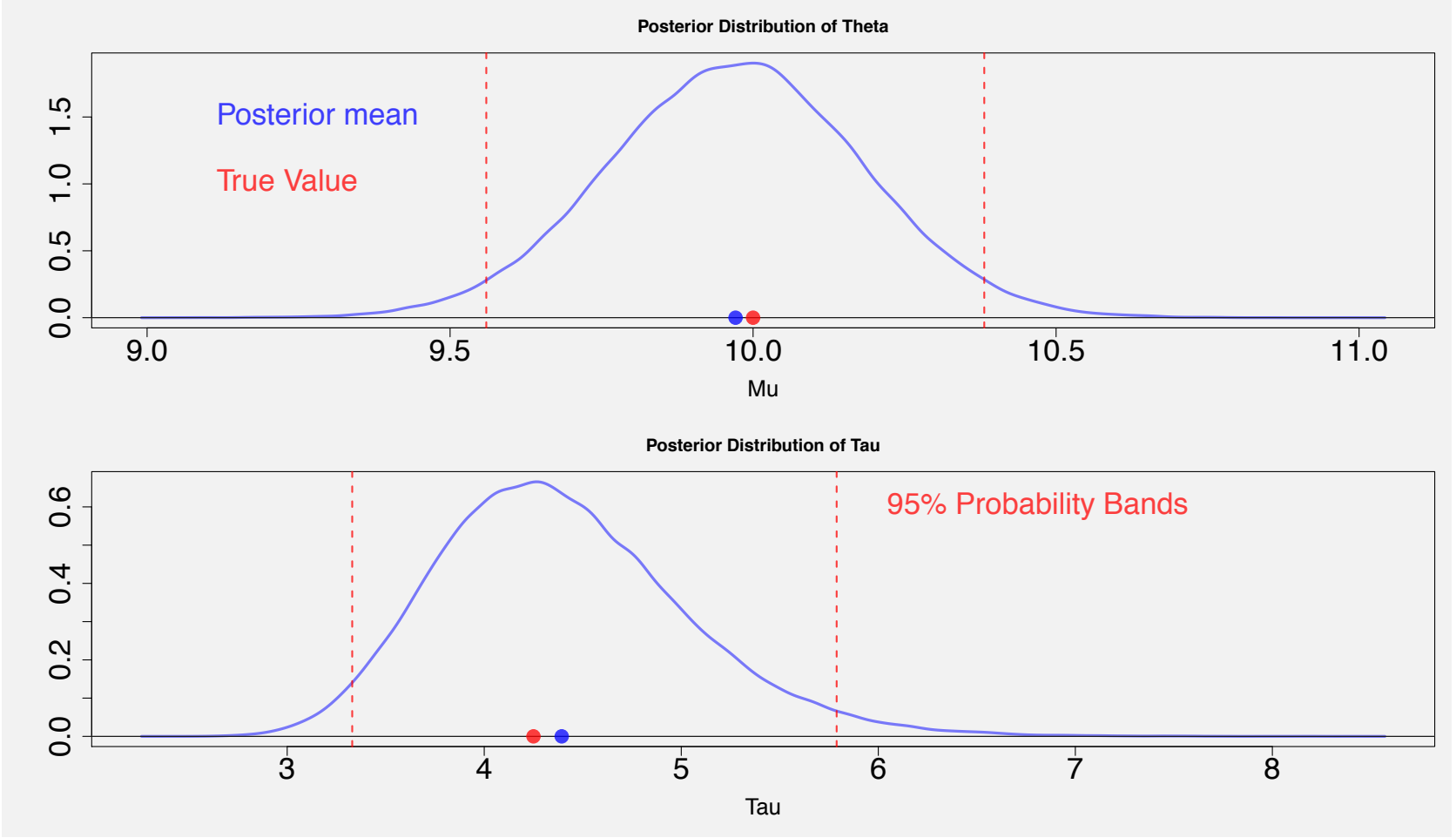
# Gibbs: ACF


Theta ACF
ACF
0.0
0.4
0.8
0
10
20
30
40
50
Lag
Tau ACF
ACF
0.0
0.4
0.8
0
10
20
30
40
50
Lag

# Gibbs: No Thinning Needed



# Estimates v Truth



# Discussion

- There are many extensions to optimize the MCMC algorithms.
- MCMC sampling needs to be tuned and monitored.
- Improvements to proposal distributions and sampling schemes will improve the algorithms run time.
- With high powered computers, MCMC can be run with little to no problems for most applications.
- Offers a powerful tool for inference.
- Should not be used blindly.

# Key References


- Gelman, A. et al. Bayesian Data Analysis. 2nd Ed. (2004). Chapman & Hall
- Robert, C. P. and Casella, G. Monte Carlo Statistical Methods. (2004/1999). Springer
- Gilks, W. R. et al. eds. Markov chain Monte Carlo in Practice. (1996). Chapman & Hall.


Lawrence Livermore
National Laboratory