Part 5:

MCMC: Metropolis and Gibbs Samplers

Full conditionals and Gibbs

- For many multi-parameter models, the joint posterior is non-standard and difficult to sample from directly
- However, it is often the case that we may easily sample from the posterior conditional distribution of each parameter
- In such cases we can construct an iterative algorithm that provides a dependent sequence of parameter values sampled from each conditional that converges to the target joint posterior distribution

This is the essence of the Gibbs sampler

From a tedious integral

In our normal model, recall how we found the posterior distribution of σ^2 by integrating over the unknown value of μ when $p(\mu, \sigma) = p(\mu|\sigma)p(\sigma)$

$$p(\sigma^2|y_1, \ldots, y_n) \propto p(y_1, \ldots, y_n|\sigma^2)p(\sigma^2)$$

$$= p(\sigma^2) \int p(y_1, \ldots, y_n|\mu, \sigma^2)p(\mu|\sigma^2)\, d\mu$$

giving $\{\sigma^2|y_1, \ldots, y_n\} \sim \mathrm{IG}(\nu_n/2, \nu_n\sigma_n^2/2)$

to a trivial sampler

This lead to the straightforward sampler

$$\begin{aligned}
\sigma^{2(1)} &\sim \mathrm{IG}(\nu_n/2, \sigma_n^2\nu/2), \quad & \mu^{(1)} &\sim \mathcal{N}(\mu_n, \sigma^{2(1)}/\kappa_n) \\
\sigma^{2(2)} &\sim \mathrm{IG}(\nu_n/2, \sigma_n^2\nu/2), \quad & \mu^{(2)} &\sim \mathcal{N}(\mu_n, \sigma^{2(2)}/\kappa_n) \\
&\vdots & &\vdots \\
\sigma^{2(S)} &\sim \mathrm{IG}(\nu_n/2, \sigma_n^2\nu/2), \quad & \mu^{(S)} &\sim \mathcal{N}(\mu_n, \sigma^{2(S)}/\kappa_n)
\end{aligned}$$

What can we do when we can't do this integral?

- e.g., when $p(\mu, \sigma^2) = p(\mu)p(\sigma^2)$

Quite a bit.

The conditional distribution of σ^2 given μ and $\{y_1, \ldots, y_n\}$ is

$$p(\sigma^2|\mu, y_1, \ldots, y_n)$$

$$\propto p(y_1, \ldots, y_n|\mu, \sigma^2)p(\sigma^2)$$

$$\propto (\sigma^2)^{-(\nu_0+n)+1} \times \exp\left\{-\frac{1}{2\sigma^2}\left[\nu_0\sigma_0^2 + \sum_{i=1}^{n}(y_i-\mu)^2\right]\right\}$$

This is a Gamma density for $\tilde{\sigma}^2 = 1/\sigma^2$, and so

$\{\sigma^2|\mu, y_1, \ldots, y_n\} \sim \mathrm{IG}(\nu_n/2, \nu_n\sigma_n^2(\mu)/2)$ where

$$\nu_n = \nu_0 + n, \quad \text{and} \quad \sigma_n^2(\mu) = \frac{1}{\nu_n}[\nu_o\sigma_0^2 + ns_n^2(\mu)]$$

Sampling from conditionals

So now we have the ability to sample from

$$\sigma^2|\mu, y_1, \ldots, y_n \quad \text{and} \quad \mu|\sigma^2, y_1, \ldots, y_n$$

Suppose we were given $\sigma^{2(1)}$, a single sample from the marginal posterior $p(\sigma^2|y_1, \ldots, y_n)$

Then we could take $\mu^{(1)} \sim p(\mu|\sigma^{2(1)}, y_1, \ldots, y_n)$ and $(\mu^{(1)}, \sigma^{2(1)})$ would be a sample from the joint posterior distribution of (μ, σ^2)

Additionally, $\mu^{(1)}$ can be considered as a sample from the marginal posterior $p(\mu|y_1, \ldots, y_n)$, from which we may obtain $\sigma^{2(2)} \sim p(\sigma^2|\mu^{(1)}, y_1, \ldots, y_n)$

Sampling from conditionals

Now, since $\mu^{(1)}$ is a sample from the marginal posterior for μ, and $\sigma^{2(2)}$ is a sample from the posterior conditional of $\sigma^2|\mu^{(1)}$, then $\{\mu^{(1)}, \sigma^{2(2)}\}$ is also a sample from the joint distribution of $\{\mu, \sigma^2\}$

This, in turn, means that $\sigma^{2(2)}$ is a sample from the marginal posterior $p(\sigma^2|y_1, \ldots, y_n)$

which can then be used to generate a new sample $\mu^{(2)}$ and so on

So the conditionals can be used to generate samples from the joint if we have $\sigma^{2(1)}$ from which to start

Full conditionals

The distributions

$$p(\mu|\sigma^2, y_1, \ldots, y_n)$$

$$p(\sigma^2|\mu, y_1, \ldots, y_n)$$

are called the full (posterior) conditional distributions, or "full conditionals" of μ and σ^2, respectively, as they are each a conditional distribution of a parameter given everything else (including the data)

Full conditionals

To make this sampling idea more precise, suppose the current state of the p-dimensional parameter is

$$\theta^{(s)} = \{\theta_1^{(s)}, \ldots, \theta_p^{(s)}\}$$

where $p = 2$ and $(\theta_1^{(s)}, \theta_2^{(s)}) = (\mu^{(s)}, \sigma^{2(s)})$ in our motivating normal example

And suppose that we can sample from each conditional

$$\pi(\theta_i | \theta_{(-i)}, y) \equiv p(\theta_i | \theta_{(-i)}, y_1, \ldots, y_n)$$

for $i = 1, \ldots, p$, where

$$\theta_{(-i)} = \{\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_p\}$$

Gibbs sampling

Gibbs Sampling (GS) proceeds as follows from some arbitrary state $\theta^{(0)} = (\theta_1^{(0)}, \ldots, \theta_p^{(0)})$. At iteration s in state $\theta^{(s)}$, take

$$\theta_1^{(s+1)} \sim \pi(\theta_1 | \theta_{(-1)}^{(s)}, y)$$

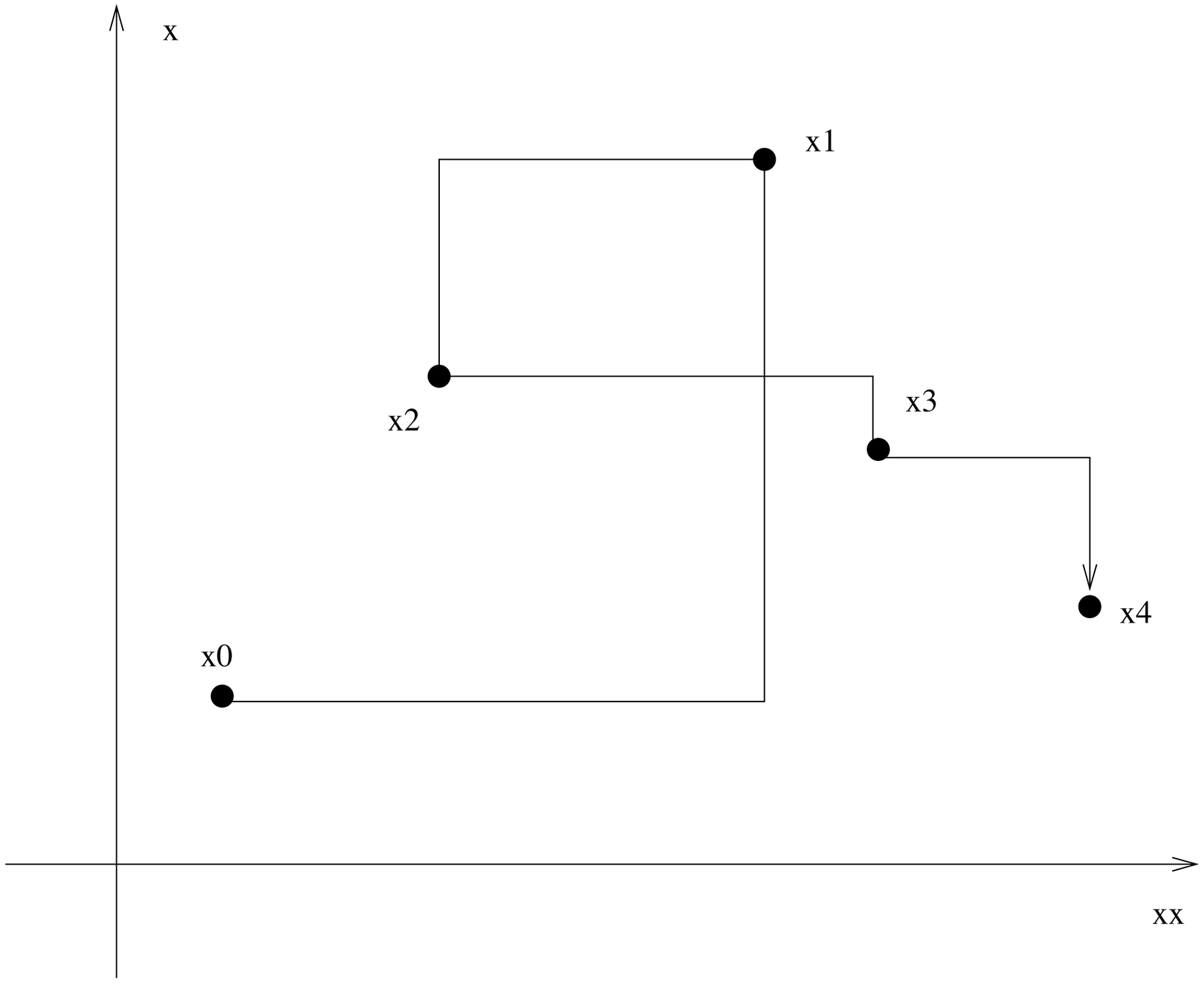
$$\vdots$$

$$\theta_i^{(s+1)} \sim \pi(\theta_i | \theta_1^{(s+1)}, \ldots, \theta_{i-1}^{(s+1)}, \theta_{i+1}^{(s)}, \ldots, \theta_p^{(s)}, y)$$

$$\vdots$$

$$\theta_p^{(s+1)} \sim \pi(\theta_p | \theta_{(-p)}^{(s+1)}, y)$$

Moving coordinate-wise



Gibbs sampling

In this way, the algorithm generates a *dependent* sequence of vectors:

$$\theta^{(1)} = \{\theta_1^{(1)}, \ldots, \theta_p^{(1)}\}$$

$$\theta^{(2)} = \{\theta_1^{(2)}, \ldots, \theta_p^{(2)}\}$$

$$\vdots$$

$$\theta^{(S)} = \{\theta_1^{(S)}, \ldots, \theta_p^{(S)}\}$$

In this sequence, $\theta^{(s)}$ depends upon $\theta^{(0)}, \ldots, \theta^{(s-1)}$ only through $\theta^{(s-1)}$

Markov property

In other words, $\theta^{(s)}$ is conditionally independent of $\theta^{(0)}, \ldots, \theta^{(s-2)}$ given $\theta^{(s-1)}$

This is called the Markov property, and so the sequence is called a Markov chain

The ergodic theorem insures that

$$P(\theta^{(s)} \in A) \rightarrow \int_A \pi(\theta|y)\, d\theta \quad \text{as} \quad s \rightarrow \infty$$

In other words, the sampling distribution of $\theta^{(s)}$ approaches the target distribution π as $s \rightarrow \infty$, no matter the starting value $\theta^{(0)}$

MCMC

More importantly, for most functions g of interest,

$$\frac{1}{S}\sum_{s=1}^{S} g(\theta^{(s)}) \rightarrow \mathbb{E}\{g(\theta)\} = \int g(\theta)\pi(\theta)\, d\theta$$

as $S \rightarrow \infty$

This means that we can approximate $\mathbb{E}\{g(\theta)\}$ with the sample average of $\{g(\theta^{(1)}), \ldots, g(\theta^{(S)})\}$ just as in MC approximation

For this reason we call such approximations Markov chain Monte Carlo (MCMC) approximations, and the procedure an MCMC algorithm

Example: bivariate normal

Suppose we wish to sample from a bivariate normal distribution $\theta = (x, y)$, where

$$\pi(\theta) \equiv f(x, y) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left\{-\frac{x^2 + y^2 - 2\rho xy}{2(1-\rho^2)}\right\}$$

and $\rho \in (0, 1)$

The full conditionals are

$$f(x|y) \propto \exp\left\{-\frac{(x-\rho y)^2}{2(1-\rho^2)}\right\} \Rightarrow X|y \sim \mathcal{N}(\rho y, 1-\rho^2)$$

(and similarly) $$\Rightarrow Y|x \sim \mathcal{N}(\rho x, 1-\rho^2)$$

Example: bivariate normal

This gives the following Gibbs Sampling algorithm

Take $\theta^{(0)} = (x^{(0)}, y^{(0)}) = (0, 0)$, say

Then, conditional on $\theta^{(s)} = (x^{(s)}, y^{(s)})$, sample

$$x^{(s+1)} \sim \mathcal{N}(\rho y^{(s)}, 1 - \rho^2)$$
$$y^{(s+1)} \sim \mathcal{N}(\rho x^{(s+1)}, 1 - \rho^2)$$

for $s = 1, \ldots, S$

Example: Midge data

Consider constructing a Gibbs Sampler for the midge data example and the prior decomposition

$$p(\mu, \sigma^2) = p(\mu)p(\sigma^2)$$

leading to the posterior conditional

$\{\sigma^2|\mu, y_1, \ldots, y_n\} \sim \text{IG}(\nu_n/2, \nu_n\sigma_n^2(\mu)/2)$ where

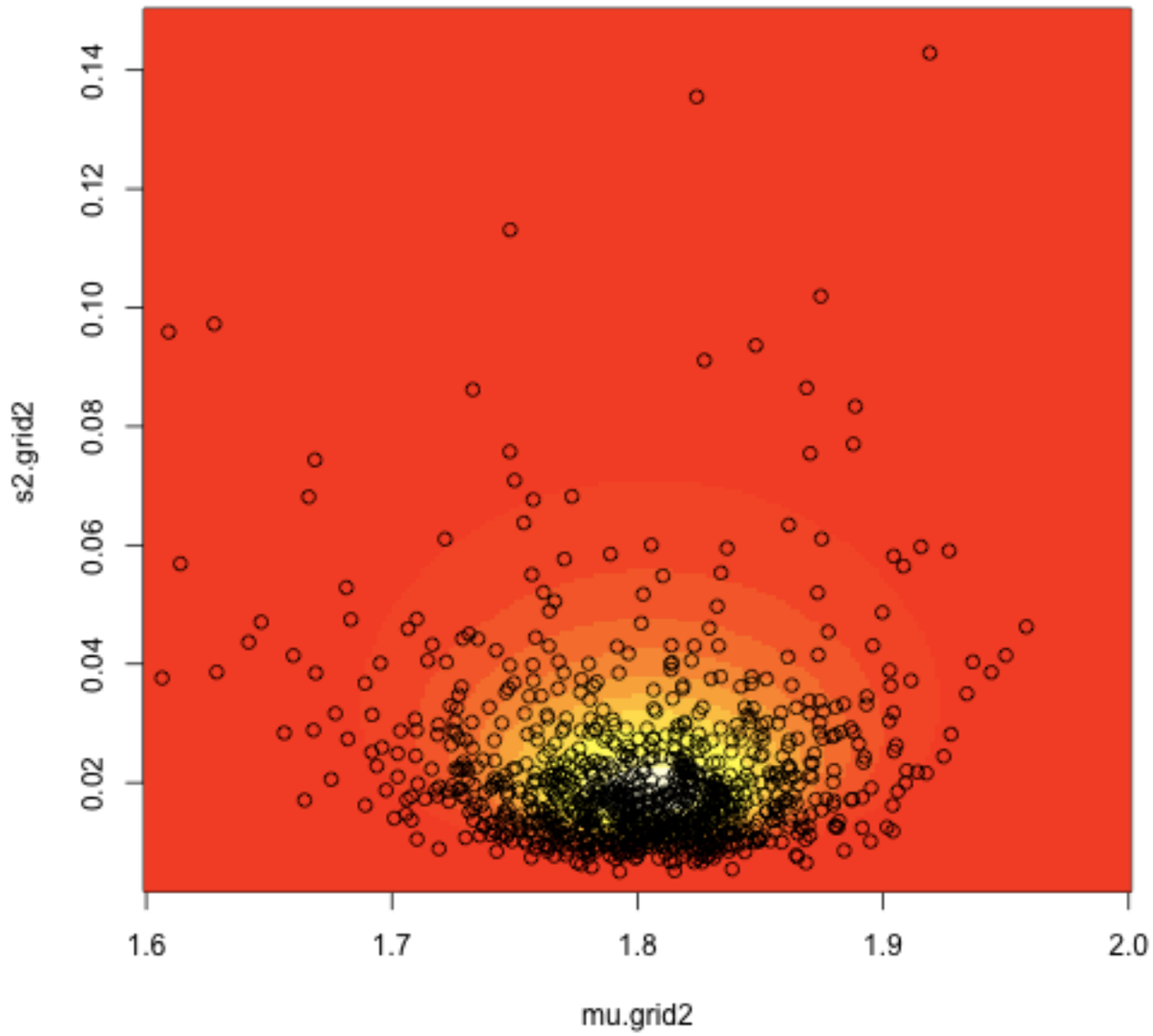
$$\nu_n = \nu_0 + n, \quad \text{and} \quad \sigma_n^2(\mu) = \frac{1}{\nu_n}[\nu_o\sigma_0^2 + ns_n^2(\mu)]$$

Our code uses the fact that

$$ns_n^2(\mu) = \sum_{i=1}^{n}(y_i - \mu)^2$$

which is more efficient in the GS loop

Example: Midge data



Quantile-based CIs

	2.5%	97.5%
μ	1.7	1.9
σ^2	0.008	0.06

$$\mathbb{E}\{\mu|y_1,\ldots,y_n\} \approx \frac{1}{1000}\sum_{s=1}^{1000}\mu^{(s)} = 1.80$$

$$\mathbb{E}\{\sigma^2|y_1,\ldots,y_n\} \approx \frac{1}{1000}\sum_{s=1}^{1000}\sigma^{2(s)} = 0.021$$

Non-conjugate priors

In situations where a conjugate prior distribution is unavailable, or undesirable, the full conditional (posterior) distributions of the parameters do not have a standard form and GS cannot easily be used

In these situations we can use the Metropolis-Hastings algorithm, which is a generic method for sampling from distributions that requires only knowledge of the density function

- which may be a posterior: product of prior and sampling model (likelihood)
- but it applies much more generally

Metropolis-Hastings

The Metropolis-Hastings (MH) algorithm is a form of generalized rejection sampling

- draw candidate samples from a proposal distribution, possibly conditional on (only) the last sample
- the samples are accepted or rejected according to the relative densities of the next sample and the last sample, and the proposal probabilities
- upon rejection, we take the next sample to be (a copy of) the last sample

thereby inducing a Markov chain

MH rejection rule

Suppose the "chain" is currently in state $\theta^{(s)}$, then the MH algorithm proposes a new state

$$\phi \sim q(\theta^{(s)}, \phi)$$

The new value is accepted with probability

$$\alpha(\theta^{(s)}, \phi) = \min\{1, A\}$$

where

$$A = \frac{\pi(\phi|y)q(\phi, \theta^{(s)})}{\pi(\theta^{(s)}|y)q(\theta^{(s)}, \phi)}$$

The MH algorithm

.... goes as follows

1. Begin in some arbitrary state $\theta^{(0)}$

2. Simulate $\phi \sim q(\theta^{(s)}, \phi)$

3. Set $\theta^{(s+1)} = \phi$ with probability $\alpha(\theta^{(s)}, \phi)$ or else reject and set $\theta^{(s+1)} = \theta^{(s)}$

4. Set $s \leftarrow s + 1$ and repeat from Step 2.

The MH algorithm

Important notes:

- Upon "rejection" the chain stays in the same place (different from rejection sampling)

- we only need to know π up to a normalizing constant

- MH can be used for each parameter individually via the full conditionals as in Gibbs Sampling

  - called Metropolis-within-Gibbs

Flavors of MH: Metropolis

There are many common choices for simulating proposals $q(\theta, \phi)$, e.g.,

Symmetric (Metropolis) proposal:

$$q(\theta, \phi) = q(\phi, \theta)$$

The acceptance probability reduces to

$$A = \frac{\pi(\phi|y)}{\pi(\theta|y)}$$

Random-Walk Metropolis

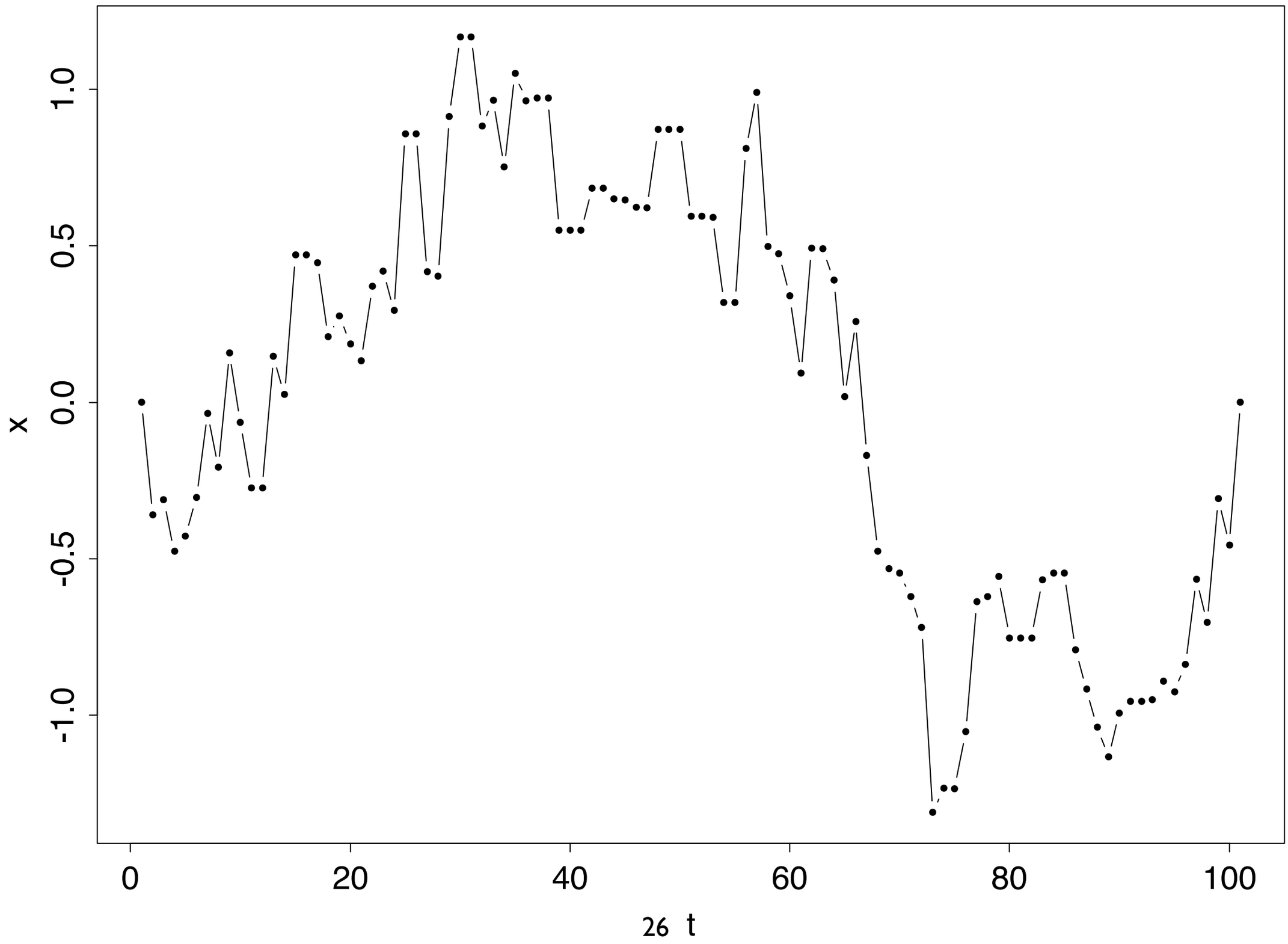
The Random Walk (RW) Metropolis algorithm involves proposals of the form

$$\phi = \theta + Z, \quad \text{where} \quad Z \sim f$$

Common choices for f include the uniform, normal, or multivariate normal

- often symmetric, but not necessarily so

Random-Walk Metropolis



Example: Normal distribution with known variance

Recall that with $\mu \sim \mathcal{N}(\mu_0, \tau_0^2)$

$$\{Y_1, \ldots, Y_n\} \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$$

the posterior is $\mu|y \sim \mathcal{N}(\mu_n, \tau_n^2)$ where

$$\mu_n = \bar{y}\frac{n/\sigma^2}{n/\sigma^2 + 1/\tau_0^2} + \mu_0\frac{1/\tau_0^2}{n/\sigma^2 + 1/\tau_0^2}$$

$$\tau_n^2 = \frac{1}{n/\sigma^2 + 1/\tau_0^2}$$

If we take $\sigma^2 = 1, \tau_0^2 = 10, \mu_0 = 5$ and observe $y = (9.37, 10.18, 9.16, 11.60, 10.33)$, then

$$\mu|y \sim \mathcal{N}(10.03, 0.196)$$

Example: A RWM approximation

Now suppose for some reason we were unable to obtain the formula for this posterior distribution and needed to use RWM with proposals

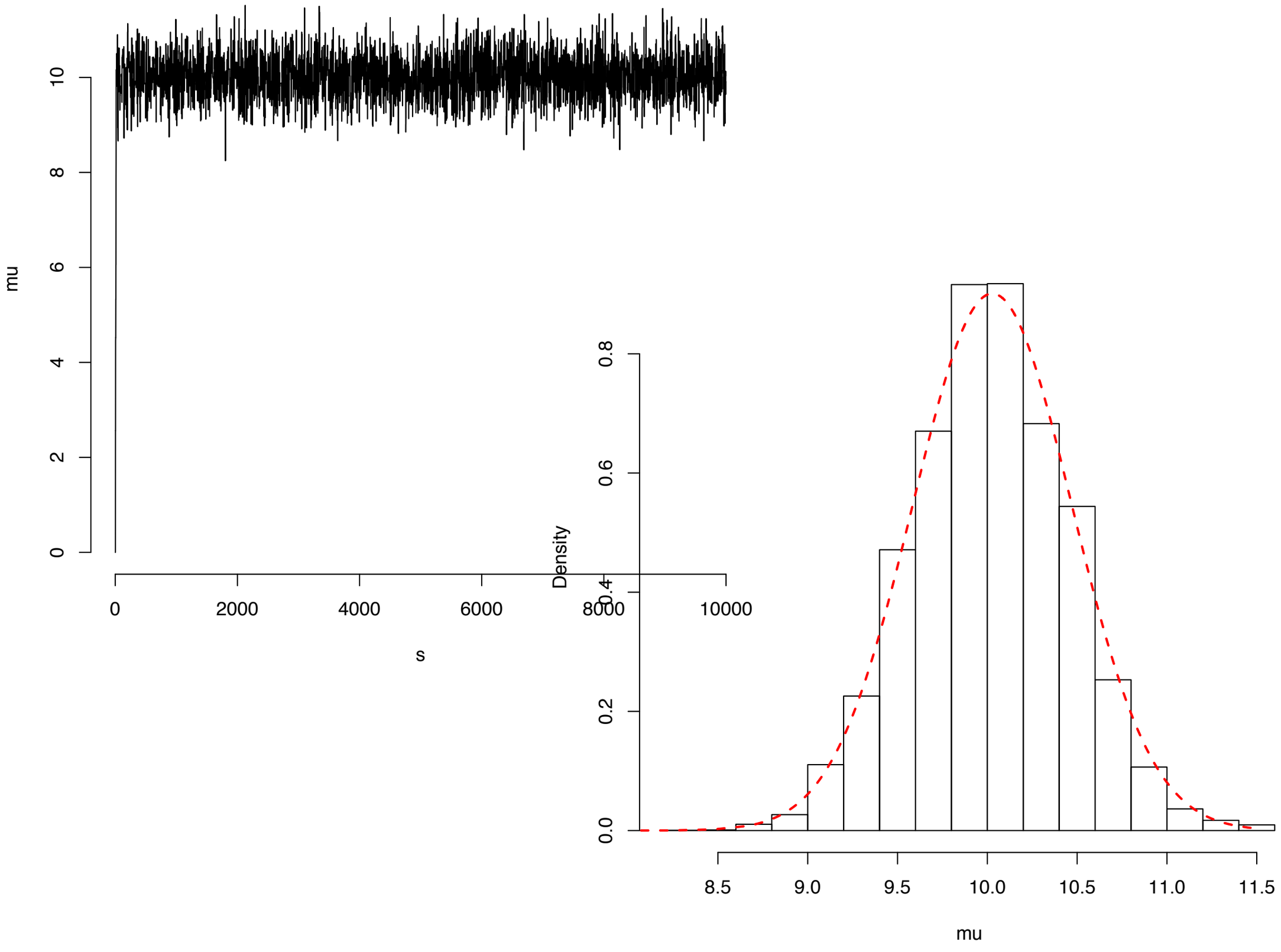
$$\mu^* \sim \mathcal{N}(\mu^{(s)}, \sigma_q^2)$$

This is symmetric, so the acceptance probability reduces to $\alpha(\mu, \mu^*) = \min\{1, A\}$ where

$$A = \frac{\pi(\mu^*|y)}{\pi(\mu^{(s)}|y)} = \frac{\prod_{i=1}^n \mathcal{N}(y_i; \mu^*, \sigma^2)}{\prod_{i=1}^n \mathcal{N}(y_i; \mu^{(s)}, \sigma^2)} \times \frac{\mathcal{N}(\mu^*; \mu_0, \tau_0^2)}{\mathcal{N}(\mu^{(s)}; \mu_0, \tau_0^2)}$$

(likelihood ratio) (prior ratio)

Example: Comparing RWM to the truth



Independence sampler

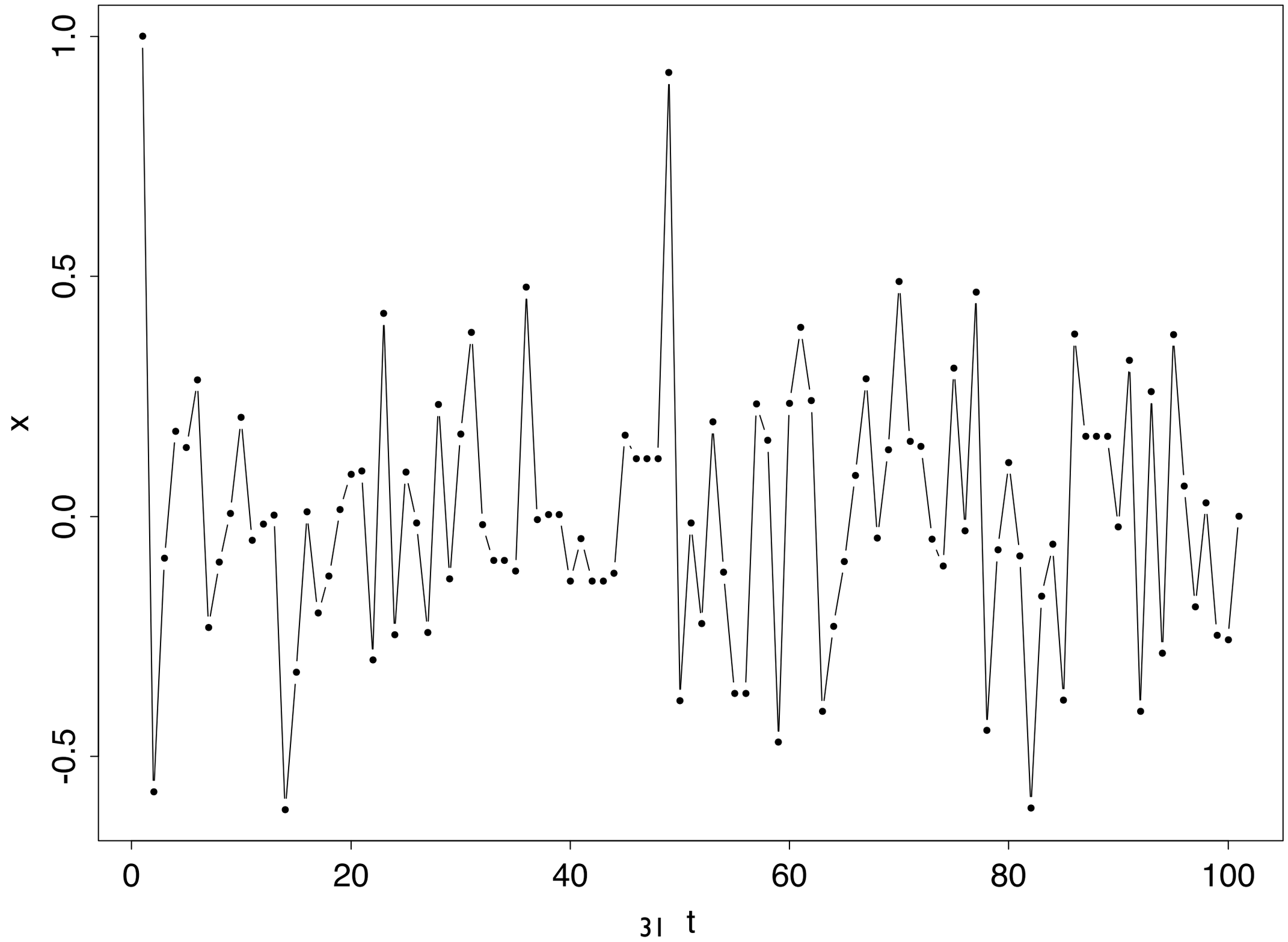
Here the proposed ϕ is drawn independently of the current state θ, so that $q(\theta, \phi) = f(\phi)$

In this case, the corresponding acceptance probability can be written as

$$\alpha(\theta, \phi) = \min\left\{1, \frac{\omega(\phi)}{\omega(\theta)}\right\} \text{ where } \quad \omega(\theta) = \frac{\pi(\theta|y)}{f(\theta)}$$

This is the importance weight function that would be used in importance sampling with draws from f

Independence sampler



Gibbs sampling

GS is a special case of the MH algorithm

We generate new values of the parameters from their corresponding (posterior) conditional distribution, which are accepted with probability one

Proof: Suppose that we have parameters

$$\theta = (\theta_1, \ldots, \theta_p)$$

We need to break each iteration of the MH algorithm into steps, and propose to update each θ_j in turn with proposal density q_j

Gibbs sampler

Consider $q_j(\theta, \phi) = \pi(\phi_j | \theta_{(-j)}, y)$ the j^{th} full conditional, with $\phi_{(-j)} = \theta_{(-j)}$ otherwise

The acceptance probability is $\min\{1, A_j\}$ where

$$A_j = \frac{\pi(\phi|y) q_j(\phi, \theta)}{\pi(\theta|y) q_j(\theta, \phi)} = \frac{\pi(\phi_{(-j)}|y)}{\pi(\theta_{(-j)}|y)} = 1$$

So using the conditional distribution for the proposed update results in a MH acceptance ratio of unity, i.e., always accept

MCMC issues

.... include, how to know

- when the Markov chain has converged to the stationary distribution π
- where to start the Markov chain, i.e., choosing $\theta^{(0)}$
- how many samples S to use to summarize empirically

The search for *good* answers to these questions is ongoing

MCMC practice

In the limit as $S \to \infty$, our approximations based on samples $\{\theta^{(1)}, \ldots, \theta^{(S)}\}$ will be exact, but in practice we cannot run the Markov chain forever

Instead, standard practice for MH or GS, is as follows:

- run the algorithm until some iteration B for which it looks like the chain has achieved stationarity
- run the algorithm S more times, generating $\{\theta^{(B+1)}, \ldots, \theta^{(B+S)}\}$
- discard $\{\theta^{(1)}, \ldots, \theta^{(B)}\}$, and use the empirical distribution of $\{\theta^{(B+1)}, \ldots, \theta^{(B+S)}\}$ to approximate $\pi(\theta|y)$

Burn-in and initialization

The iterations up to and including B are called the burn-in period, in which the Markov chain moves from its initial value to a region of the parameter space that has high posterior probability

If we have a good idea where the high probability region is, we can reduce the burn-in period by starting the Markov chain there

E.g., starting at $\mu^{(0)} = \bar{y}$ in our Normal example is sensible

However, starting with $\mu^{(0)} = 0$ illustrated that the MH algorithm was able to move from a low posterior probability region to one of high probability

MCMC diagnostics

How can we tell when our MCMC sampler has "reached stationarity"?

It turns out that this is a very difficult question to answer in any concrete way

However, it is easy to identify several "desirable properties" that can be used to make qualitative statements about MCMC samplers, and thereby make informed comparisons between competing choices

This will help us choose amongst proposal mechanisms, and pick the burn-in size B

Autocorrelation

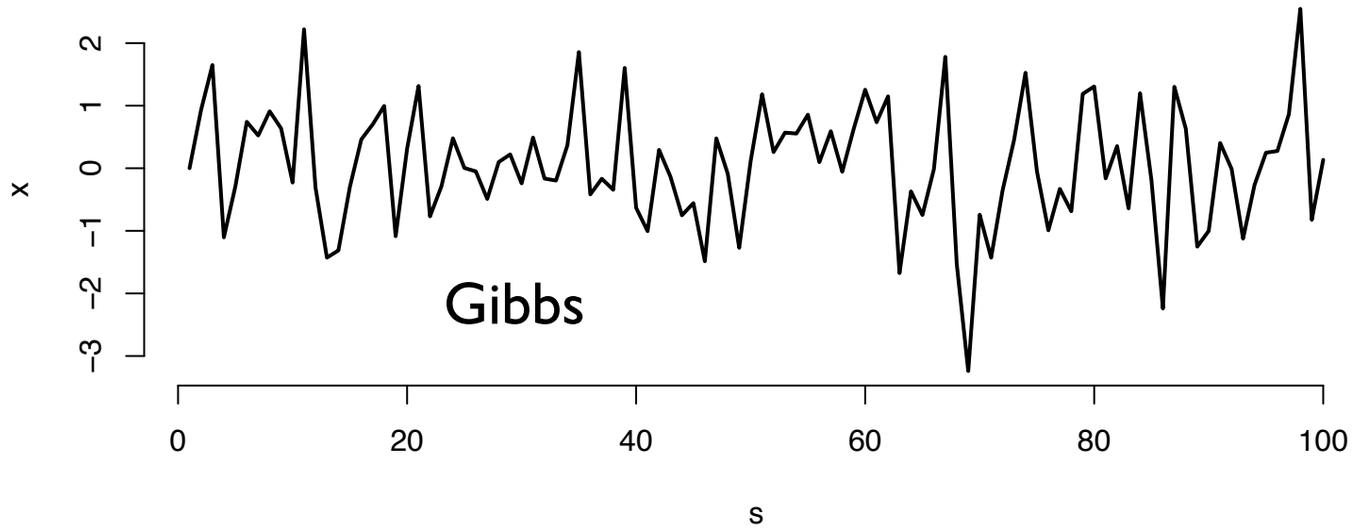
The main culprit in poor MCMC performance is high autocorrelation, or "stickiness" in the chain

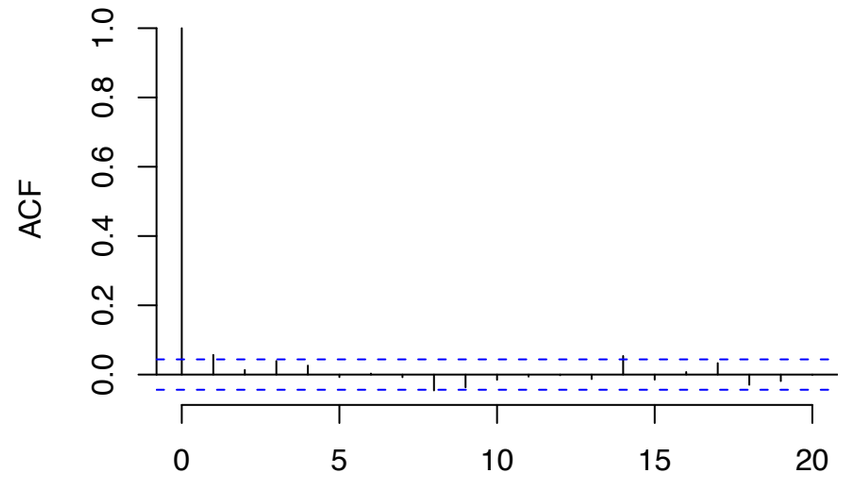
Monte Carlo simulation, in which we generate independent samples directly from the target (posterior) distribution, is in some sense the "gold standard"

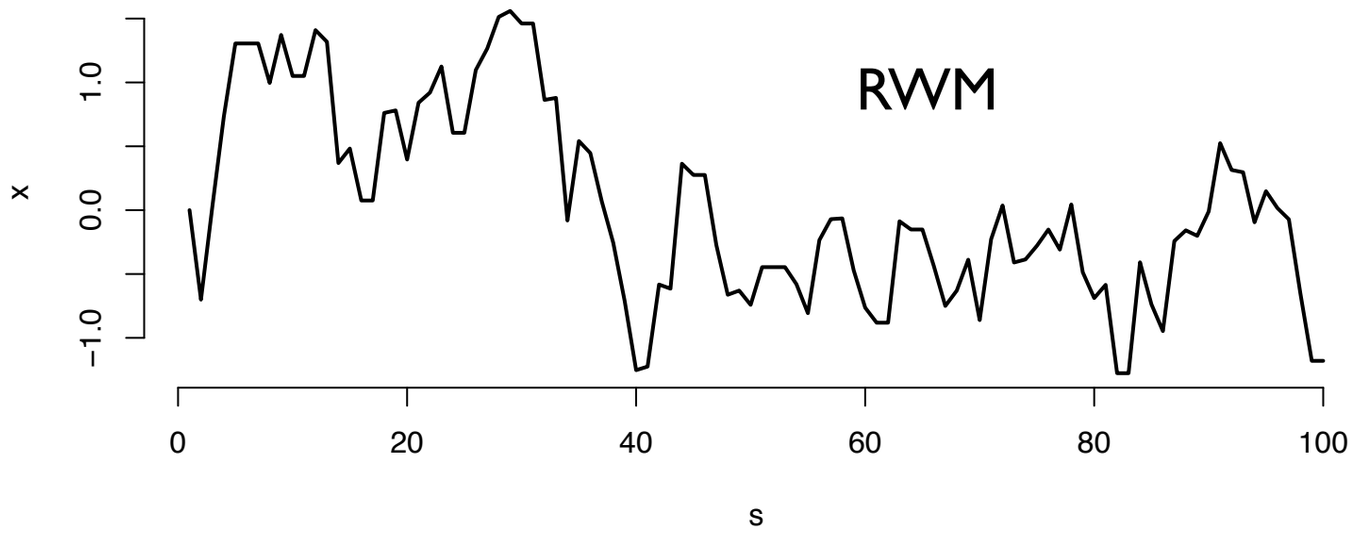
- since MC samples are independent they are uncorrelated

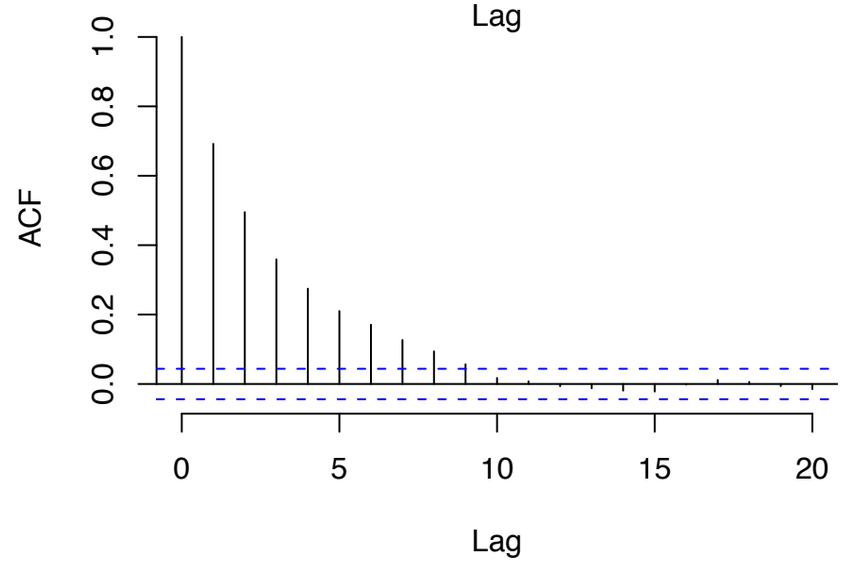
So estimators based upon MC samples perform better than MCMC based ones since Markov chains produce inherently correlated samples

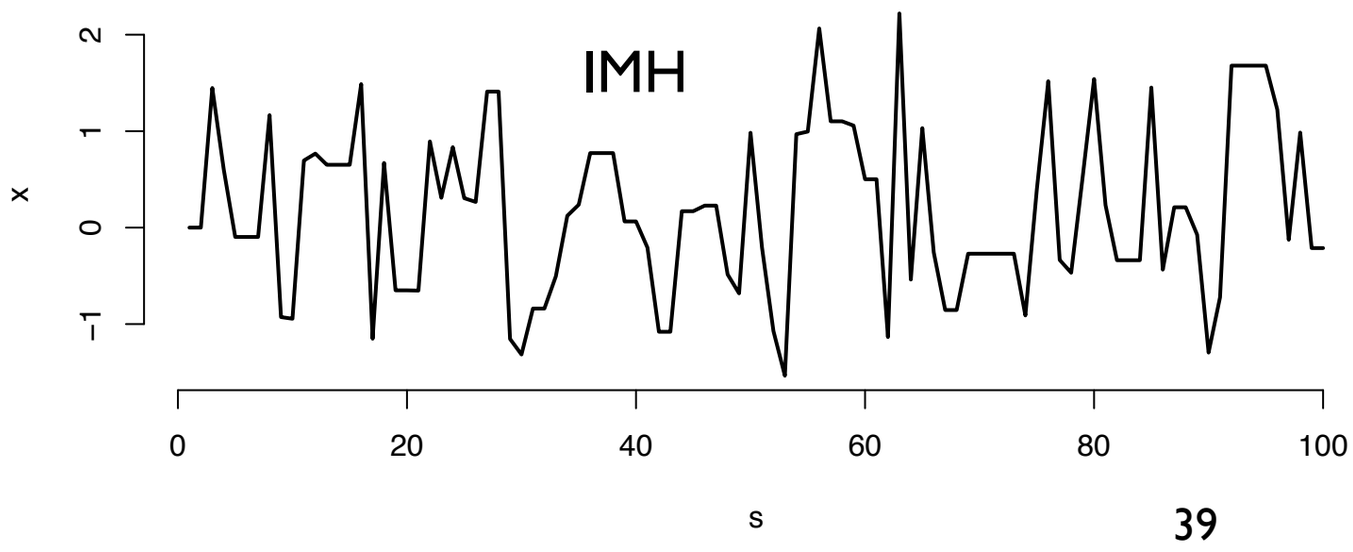
Example: Autocorrelation

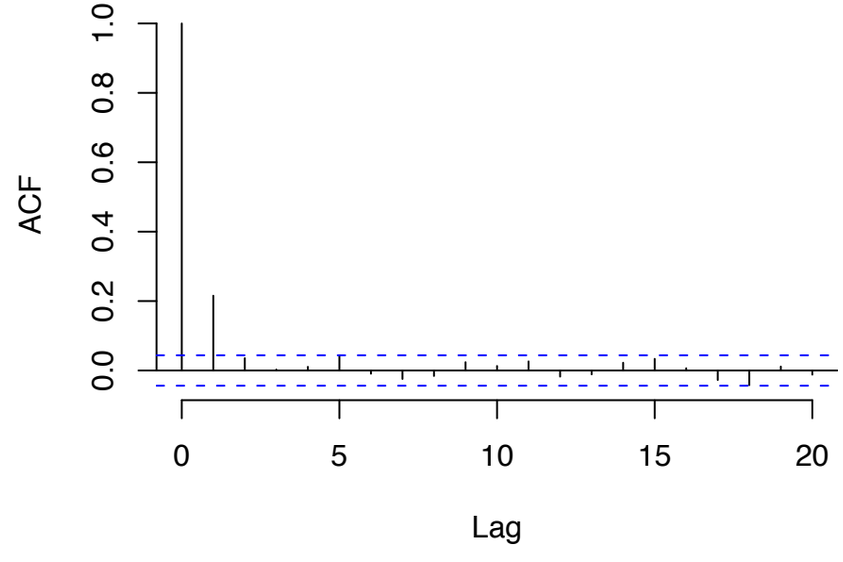












High autocorrelation

A Markov chain with high autocorrelation, such as our RWM example, moves around the parameter space slowly, taking a long time to achieve the correct balance of samples according to the (posterior) density

The higher the autocorrelation, the more MCMC samples will be needed to attain a given level of precision for the approximation

effectiveSize

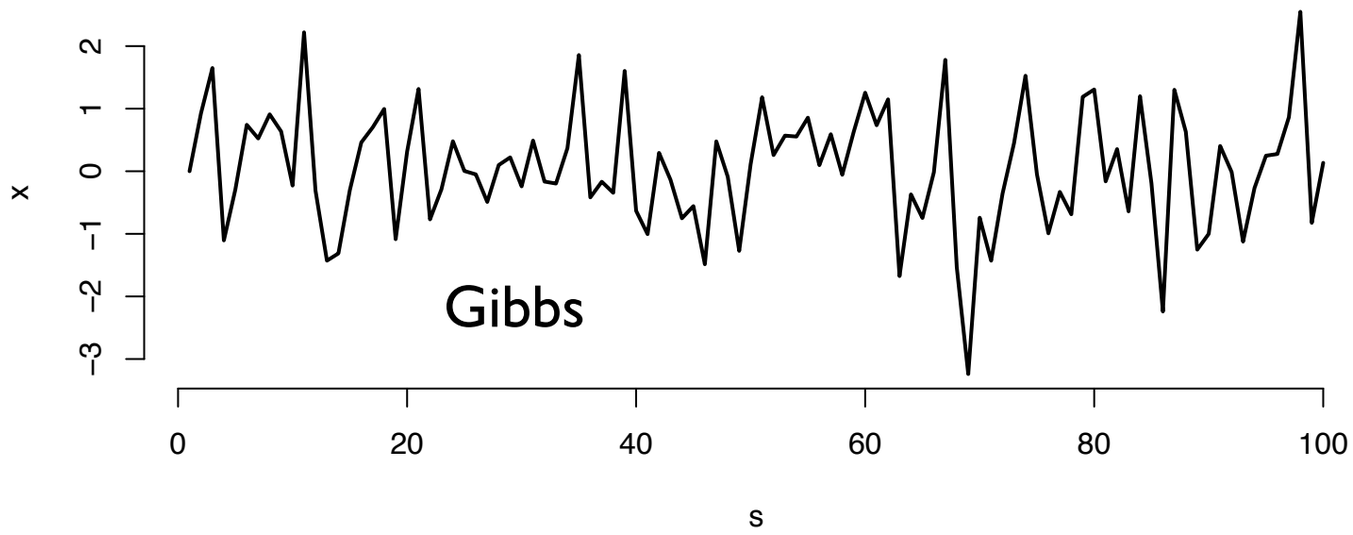
One (single number) commonly used to measure the efficiency in a collection of samples $\theta = \{\theta_1, \ldots, \theta_S\}$ is

$$S_{\text{eff}} = \text{ESS}(\theta) = \frac{S}{1 + 2\sum_{t=1}^{S-1} \hat{\rho}_t(\theta)}$$

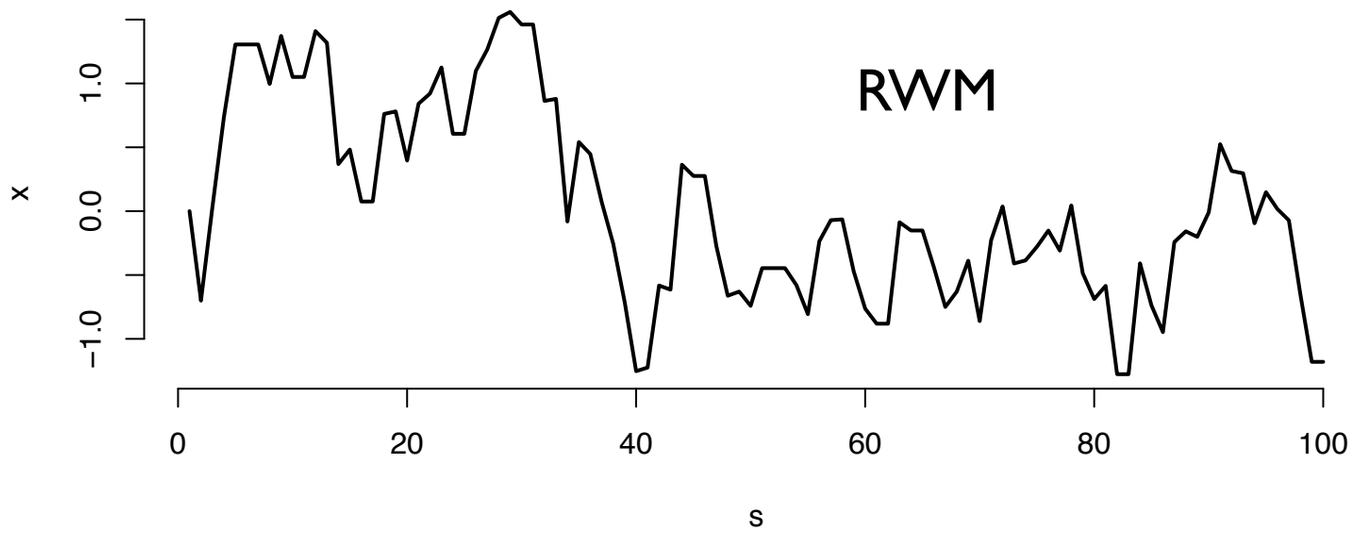
where $\hat{\rho}_t(\theta)$ may be any estimator of the autocorrelation in the collection of samples θ at lag t

- $\hat{\rho}_t(\theta) = \text{acf}_t(\theta)$ is one choice
- the function `effectiveSize` in the R library `coda` models θ with an autoregressive (AR) model to obtain a more robust estimator

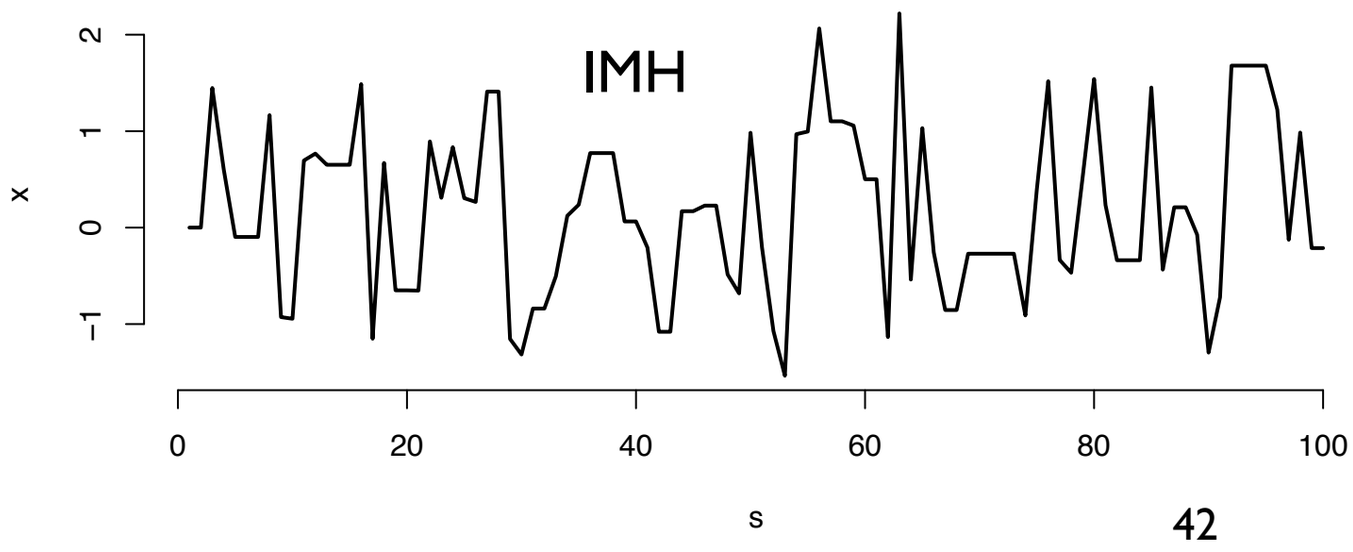
Example: Autocorrelation $S = 2000$



$$S_{\text{eff}} = 1764$$



$$S_{\text{eff}} = 86$$

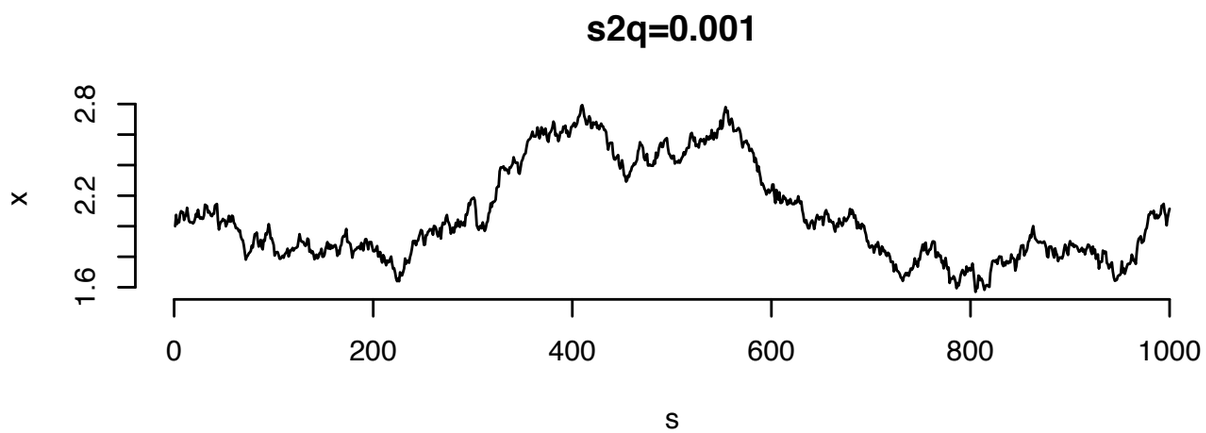


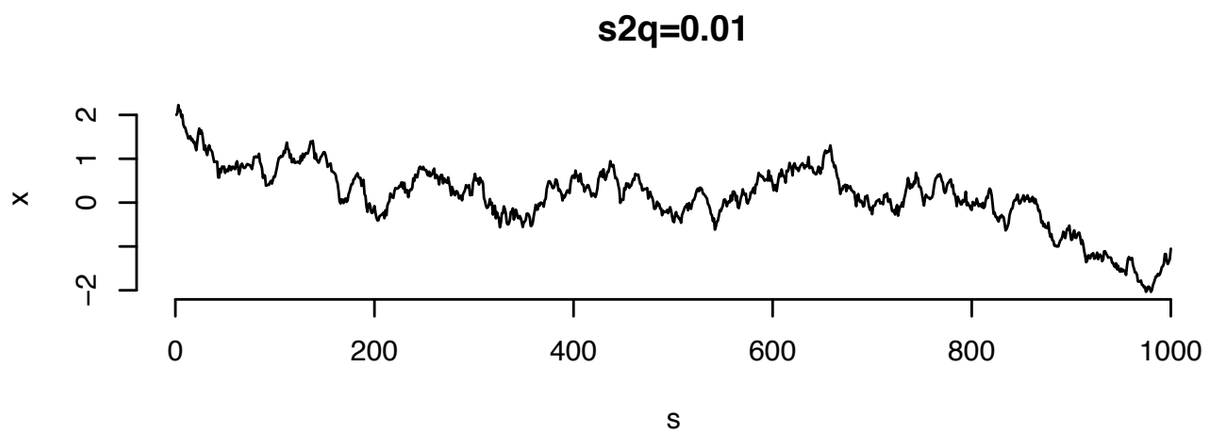
$$S_{\text{eff}} = 1191$$

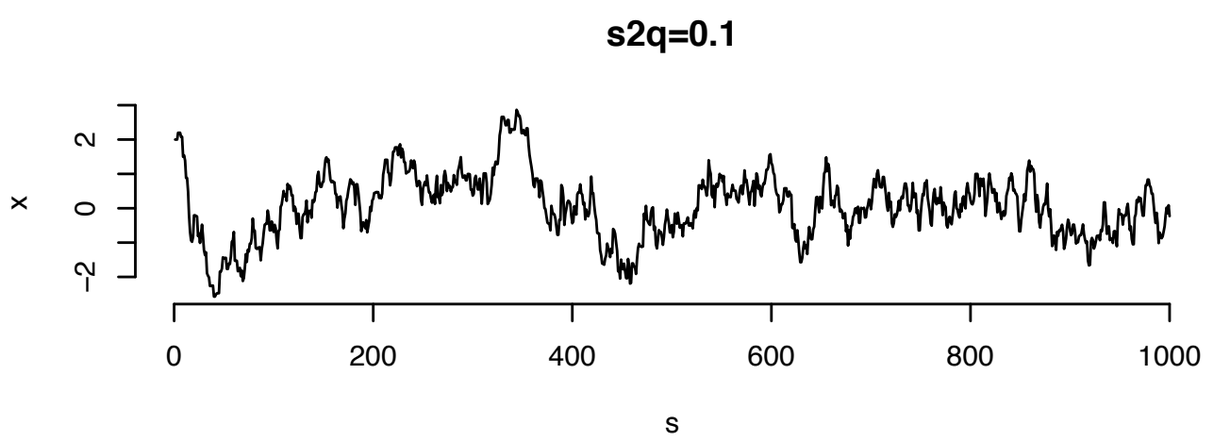
Controlling autocorrelation

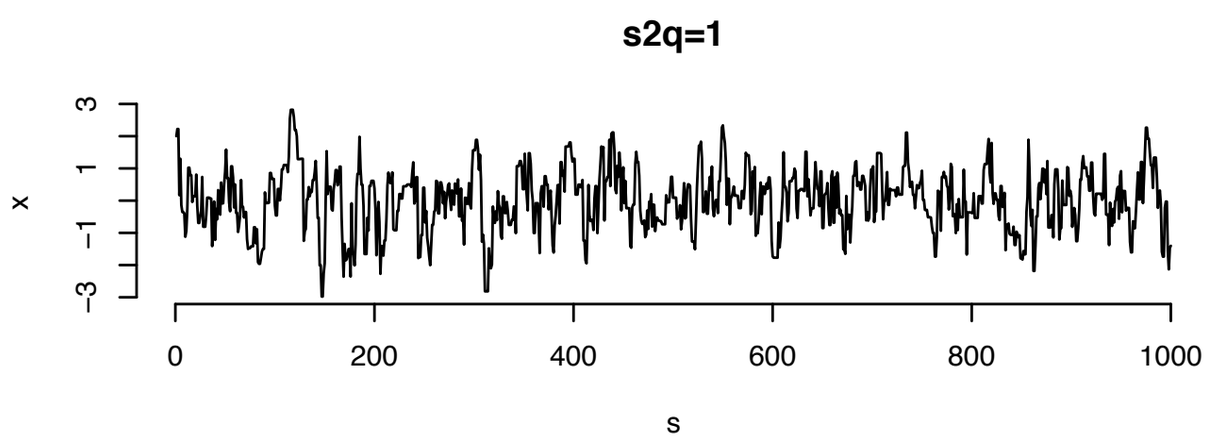
- So the more correlated our Markov chain is, the less information we get per iteration
- It would seem that GS offers lower correlation than MH since in GS the "proposals" are never rejected
- In MH we may adjust the level of correlation by adjusting the proposal mechanism $q(\theta, \phi)$
  - usually by adjusting the proposal variance, e.g., σ_q^2

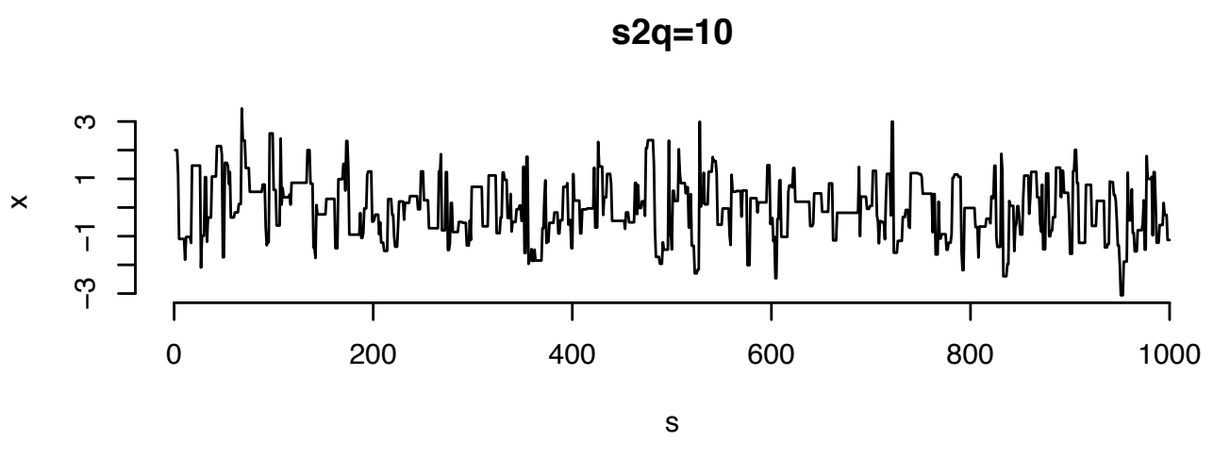
Example: RWM proposal choice

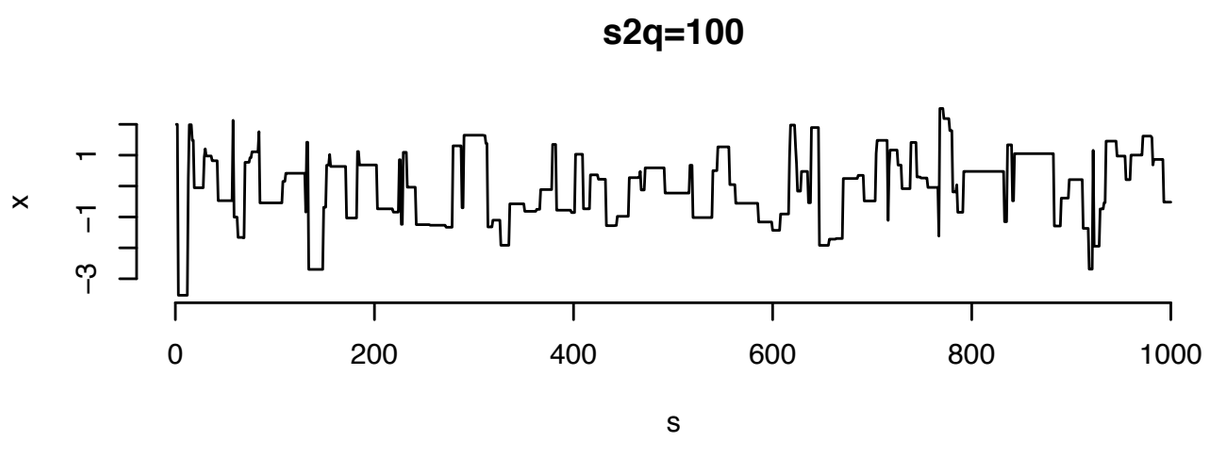












Moderate proposals

In order to construct a Markov chain with MH that has a low autocorrelation we need a proposal variance that is

- large enough so that the chain can quickly move throughout the parameter space
- but not so large that proposals end up being rejected most of the time