Post-Processing of MCMC

Leah South

Queensland University of Technology

15th July, 2021

Bayesian Statistics

In Bayesian statistics, we are interested in the posterior distribution of the model parameters θ given the data y:

$$\pi(\theta|y) = \frac{\ell(y|\theta)p(\theta)}{Z},$$

where $\ell(y|\theta)$ is the likelihood, $p(\theta)$ is the prior and $Z = \int_\Theta \ell(y|\theta)p(\theta)\mathrm{d}\theta$ is referred to as the evidence.

Monte Carlo

Most quantities of interest can be written as posterior expectations

$$I = \mathbb{E}_\pi[g(\theta)] = \int_\Theta g(\theta)\pi(\theta|y)\mathrm{d}\theta.$$

These are analytically intractable in many cases, but we can estimate them using *Monte Carlo* integration:

$$\hat{I}_{\mathrm{MC}} = \frac{1}{T}\sum_{t=1}^{T} g(\theta^{(t)}) \text{ where } \{\theta^{(t)}\}_{t=1}^{T} \overset{iid}{\sim} \pi.$$

This is an unbiased estimator and the variance of $\hat{I}_{\mathrm{MC}}$ is $\mathrm{Var}_\pi[g(\theta)]/T$.

Why use MCMC?

We can rarely simulate directly $\{\theta^{(t)}\}_{t=1}^{T} \overset{iid}{\sim} \pi$.

Markov chain Monte Carlo (MCMC) is a method that produces correlated draws from the posterior after the Markov chain has *converged*.

This lecture is about how to use our samples from MCMC.

Random Walk MH-MCMC (RWM)

The most basic and common MH-MCMC algorithm is:

Set $\theta^{(0)}$.

For $t = 1, ..., T$

- Propose $\theta^* \sim \mathcal{N}(\theta^{(t-1)}, h^2\Sigma)$
- Compute MH acceptance probability

$$\alpha_{\text{HM}} = \min\left(1, \frac{\ell(\theta^*|y)p(\theta^*)}{\ell(\theta^{(t-1)}|y)p(\theta^{(t-1)})}\right)$$

- Draw $u \sim \text{Unif}[0, 1]$
  If $u < \alpha_{\text{MH}}$ set $\theta^{(t)} = \theta^*$
  otherwise set $\theta^{(t)} = \theta^{(t-1)}$

MH-MCMC

The MH-MCMC algorithm is:

Set $\theta^{(0)}$.

For $t = 1, ..., T$

- Propose $\theta^* \sim q(\cdot|\theta^{(t-1)})$
- Compute MH acceptance probability

$$\alpha_{\text{HM}} = \min\left(1, \frac{\ell(\theta^*|y)p(\theta^*)q(\theta^{(t-1)}|\theta^*)}{\ell(\theta^{(t-1)}|y)p(\theta^{(t-1)})q(\theta^*|\theta^{(t-1)})}\right)$$

- Draw $u \sim \text{Unif}[0, 1]$
  If $u < \alpha_{\text{MH}}$ set $\theta^{(t)} = \theta^*$
  otherwise set $\theta^{(t)} = \theta^{(t-1)}$

Demo

Dr Chi Feng (MIT) produced this nice interactive demo:

`https://chi-feng.github.io/mcmc-demo/app.html`

Running Example: Radiata Pines

Our running example is a simple linear regression model about strength of radiata pine trees.

The goal is to estimate the posterior for $\theta = (\alpha, \beta, \sigma^2)$ given

$$y^{(i)} = \alpha + \beta(x^{(i)} - \bar{x}) + \epsilon^{(i)} \text{ where } \epsilon^{(i)} \sim \mathcal{N}(0, \sigma^2),$$

y is the maximum compression strength parallel to the grain and x is the density. Our priors are $\alpha \sim \mathcal{N}(3000, 10^6)$, $\beta \sim \mathcal{N}(185, 10^4)$ and $\sigma^2 \sim \text{InvGamma}(3, (2 \times 300^2)^{-1})$.

Checking the Output

What can go wrong in MCMC?

Checking the Output

What can go wrong in MCMC?

1. The MCMC algorithm as implemented is invalid
2. MCMC hasn't *converged*
3. MCMC not exploring the space well: high autocorrelation
4. Not run for long enough

Point 1 can be addressed through correct implementation of an ergodic, π-invariant MCMC algorithm.

Outline

This lecture series is about tools for answering the questions:

- Which samples should we use?
- Do we have good quality samples?
- How can we improve estimates of expectations?

This lecture covers the first two questions in the context of standard MCMC. Lecture 2 covers approaches that apply to standard MCMC, *biased MCMC* and beyond.

Standard MCMC

Outline for This Lecture

Standard MCMC

- *Theory:* Theory for standard MCMC.
- *Which samples:* Which samples should we use in standard MCMC?
- *Sample quality:* Do we have good enough samples for estimation (standard MCMC)?

Motivation for tomorrow

- What is biased MCMC and why do we need new tools for it?

Theory

Markov chains

A *Markov chain* is a stochastic process, where a system transitions from one state to another, based only on information about the current state of the chain.

Mathematically, $P(\theta^{(t)}|\theta^{(0)}, \theta^{(1)}, \ldots, \theta^{(t-1)}) = P(\theta^{(t)}|\theta^{(t-1)})$ so we have conditional independence.

However $\theta^{(t)}$ is still correlated with all earlier states so the samples from our Markov chain suffer from *autocorrelation* ('auto' means 'self').

Markov chains

We say that π is an invariant (or "stationary") distribution of a Markov chain when $\theta^{(t-1)} \sim \pi$ implies $\theta^{(t)} \sim \pi$.

We say that π is also the limiting distribution of the Markov chain if the distribution of $\theta^{(t)}$ converges to π as $T \to \infty$.

Perfect Start

MCMC algorithms are constructed so that the posterior is an invariant distribution of the Markov chain.

Perfect Start

MCMC algorithms are constructed so that the posterior is an invariant distribution of the Markov chain.

Consequence:
If we start at $\theta^{(0)} \sim \pi$, then our chain is a set of correlated samples from π.
The estimator

$$\hat{I}_{\text{MCMC}} = \frac{1}{T} \sum_{t=1}^{T} g(\theta^{(t)})$$

is unbiased and its variance depends on the correlation in the Markov chain.

Asymptotic Properties

Under mild conditions[1], the Markov chain is ergodic so the posterior π is the limiting distribution.

[1] Possible to get from each state to each other state in a finite number of steps & doesn't get stuck in periodic cycles.

Asymptotic Properties

Under mild conditions[1], the Markov chain is ergodic so the posterior π is the limiting distribution.

Consequence:
Regardless of $\theta^{(0)}$, we are guaranteed eventual convergence to π and the ergodic theorem tells us that

$$\hat{I}_{\text{MCMC}} = \frac{1}{T}\sum_{t=1}^{T} g(\theta^{(t)}) \to I \text{ as } T \to \infty.$$

The estimator is consistent and its variance depends on the correlation in the Markov chain.

[1] Possible to get from each state to each other state in a finite number of steps & doesn't get stuck in periodic cycles.

Central Limit Theoreom

We can make the dependence of the variance of $\hat{I}_{\text{MCMC}}$ on the correlation can be made explicit for the case where $\theta^{(0)} \sim \pi$ and $T \to \infty$.

Consequence:
In this setting, the variance of $\hat{I}_{\text{MCMC}}$ is $\text{Var}_{\text{MCMC}}[g(\theta)]/T$, where

$$\text{Var}_{\text{MCMC}}[g(\theta)] = \text{Var}[g(\theta^{(0)})] + 2\sum_{t=1}^{\infty} \text{Cov}[g(\theta^{(0)}), g(\theta^{(t)})]$$

Markov chain Monte Carlo

So we have some nice theoretical guarantees for

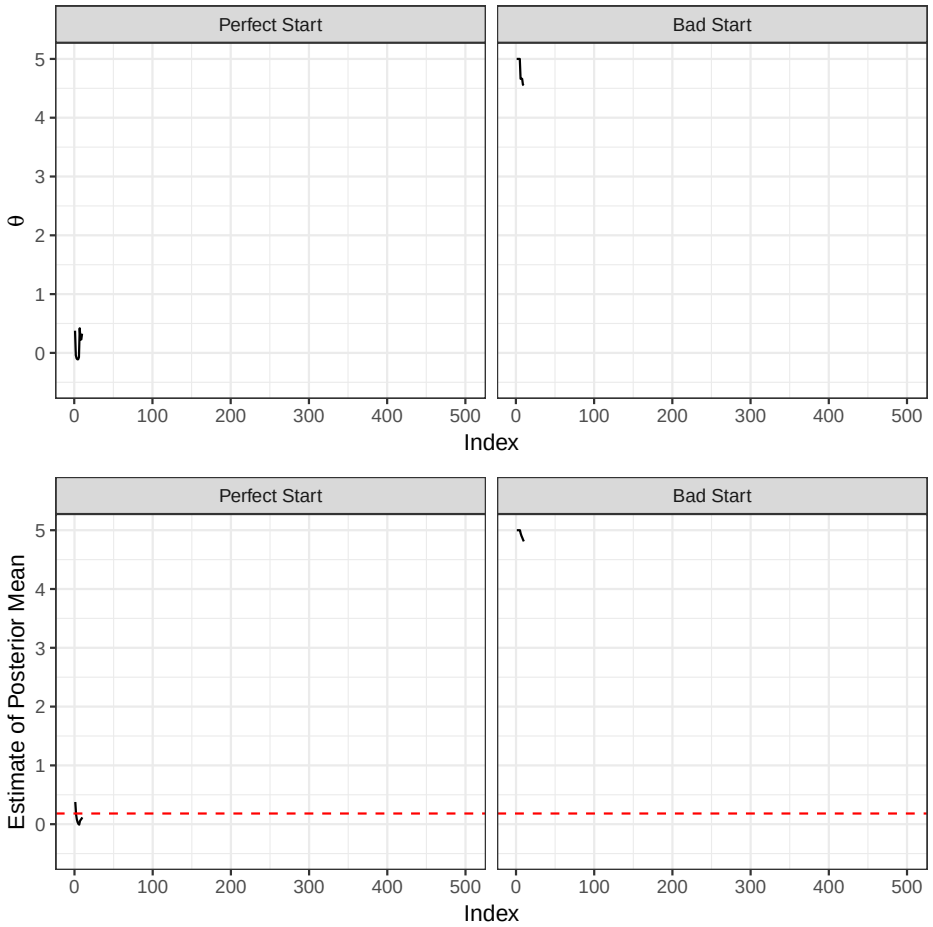
- Finite T and $\theta^{(0)} \sim \pi$ (unbiasedness)
- $T \to \infty$ (consistency).

These theoretical guarantees don't apply for the situation we're in of

- Being unable to draw $\theta^{(0)} \sim \pi$ (otherwise we wouldn't need MCMC!)
- Being unable to perform an infinite number of iterations.

The MCMC estimator $\hat{I}_{\text{MCMC}} = \frac{1}{T}\sum_{t=1}^{T} g(\theta^{(t)})$ is not unbiased.

An Example



Given that we can't trust our (early) samples, which samples should we use?

Which Samples?

Which Samples?

We often don't use the full MCMC chain for estimation.

There are two main practices for discarding samples in MCMC:

- Removing a "burn-in": Removing the first M iterations which are the samples prior to convergence.
- Thinning: using only every kth sample.

Burn-In

After *a large enough number of iterations*, the dependence on $\theta^{(0)}$ is small enough to be ignored and we informally say that the Markov chain has converged to π.

We can reduce the bias by removing the initial samples that are clearly not from π. The new estimator is:

$$\frac{1}{T-M}\sum_{t=M+1}^{T} g(\theta^{(t)})$$

where the $\theta_1, \ldots, \theta_M$ are discarded as *burn-in* or *warm-up*. The challenge is to choose M.

Convergence Assessment

Assessing the convergence of the Markov process is a challenge.

At best we can say we are not aware that something has gone wrong.

We pick the smallest M such that we have no evidence of non-convergence for the remaining samples.

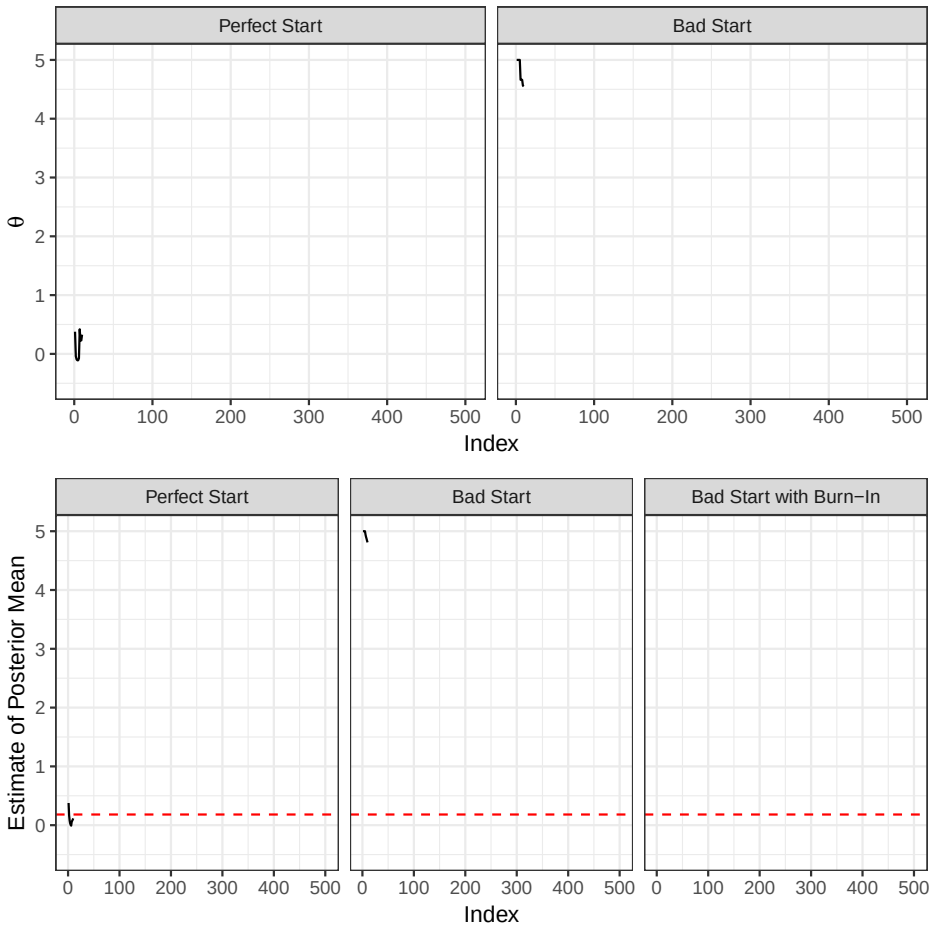
Convergence Assessment

Some approaches for assessing convergence are:

- Look at trace plots - can be subjective
- Formal (inconclusive) tests, e.g.
  - Brooks-Gelman-Rubin diagnostic, aka $\hat{R}$ (gelman.diag)
  - Geweke diagnostic (geweke.diag)
  - Raftery-Lewis diagnostic (raftery.diag)
  - Heidelberger-Welch diagnostic (heidel.diag)

All of these formal tests are included in the coda package in R.

Trace Plots 1D Example



In this case, we say that the Markov chain hasn't converged until around iteration 100.

Trace Plots

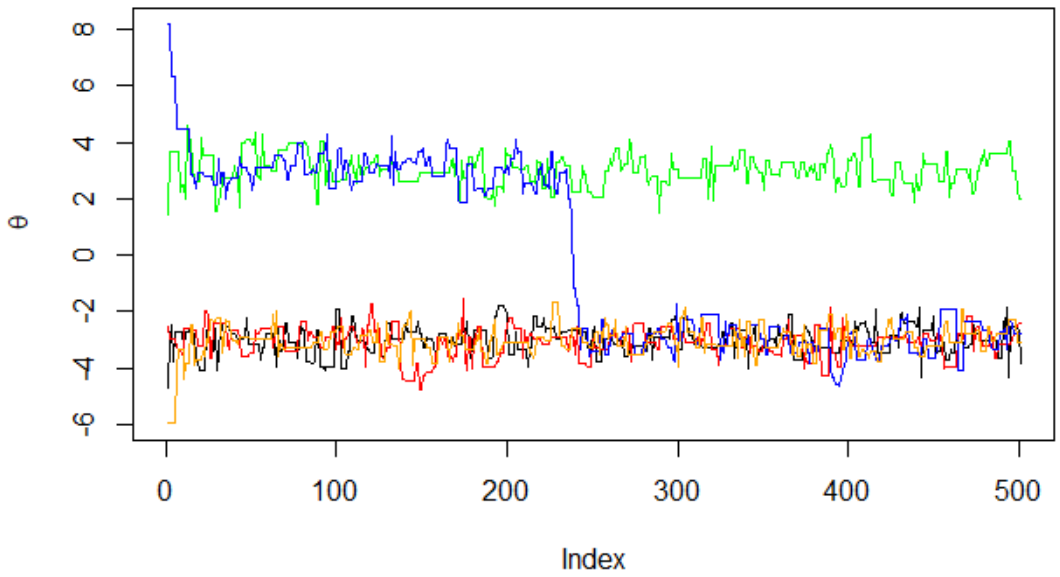
Non-convergence might appear as samples moving towards a region. After convergence, the samples should be moving around the posterior space.

We should look at the trace plot for all parameters and potentially for other functions such as $\log \ell(y|\theta) + \log p(\theta)$. We have only converged if the full distribution of θ has converged.

Let's take a look at the radiata pine example.

Trace Plots

We should also consider multiple starting points.



Let's take a look at the radiata pine example.

The $\hat{R}$ Diagnostic

The most commonly used diagnostic is $\hat{R}$ (aka potential scale reduction factor) which was first introduced in Gelman & Rubin (1992), corrected in Brooks & Gelman (1998) and simplified in Gelman et al (2003).

The idea is to run M different chains of length T from different, overdispersed starting points.

We compare the variance between chains and the variance within chains using ideas similar to ANOVA.

The $\hat{R}$ Diagnostic

There are two ways to estimate the posterior variance of a function $g(\theta)$ from M chains of length T:

- The mean within-chain variance:

$$W = \frac{1}{M} \sum_{m=1}^{M} \hat{\sigma}_m^2.$$

- The empirical variance from all chains:

$$V = \frac{T-1}{T} W + \frac{1}{M-1} \sum_{m=1}^{M} (\hat{g}_m - \frac{1}{M} \sum_{m=1}^{M} \hat{g}_m)^2.$$

Where $\hat{g}_m$ and $\hat{\sigma}_m^2$ are the estimated posterior mean and variance of $g(\theta)$, respectively, from mth chain.

The $\hat{R}$ Diagnostic

As $T \to \infty$, the within-chain estimates W and the overall estimate V both approach the true posterior variance.

Prior to convergence, W is an underestimate (because we haven't explored the posterior fully) and V is an overestimate (because of the overdispersed starting points).

If $\frac{V}{W} \approx 1$ then this is a good sign.

The $\hat{R}$ Diagnostic

To account for sampling variability, Brooks and Gelman propose the corrected $\hat{R}$:

$$\hat{R} = \sqrt{\frac{\hat{d}+3}{\hat{d}+1}\frac{V}{W}}.$$

where $\hat{d}$ is an estimated degrees of freedom parameter from a t-distribution approximation of estimated variances.

For an ergodic Markov chain, $\hat{R} \to 1$ as $T \to \infty$. Confidence in convergence is increased if $\hat{R} <$ say 1.1 for all parameters. We should pick the burn-in such as that, after removal of the burn-in, $\hat{R} < 1.1$ for all marginals.

The $\hat{R}$ Diagnostic

Let's take a look at the radiata pine example.

Geweke Diagnostic

What if we only have a single Markov chain?

Geweke proposed a check for convergence based on to test if the mean of a function $g(\theta)$ for the first and last part of a Markov chain are equal.

We denote the first part of the chain by A and the second part of the chain by B. Typically section A is the first 10% of the chain and section B is the last 50% of the chain.

Geweke Diagnostic

The null hypothesis is that the mean of $g(\theta)$ for section A of the chain is equal to the mean of $g(\theta)$ for section B of the chain.

$$H_0 : \mu_A(g) = \mu_B(g)$$
$$H_a : \mu_A(g) \neq \mu_B(g)$$

If we reject H_0 then this indicates that we haven't achieved convergence yet in section A. Failing to reject H_0 doesn't mean that the Markov chain has converged.

Geweke Diagnostic

The Geweke diagnostic is the Z statistic:

$$Z = \frac{\bar{g}_A - \bar{g}_B}{\sqrt{S_g^A / T_A + S_g^B / T_B}}$$

where $\bar{g}_A$ and $\bar{g}_B$ are the sample means for the two subsamples, T_A and T_B are the sample sizes and $S_g^A(0)/T_A$ and $S_g^B(0)/T_B$ are the estimated standard errors of the means that use the estimated correlation between the samples.

If components A and B are independent, then asymptotically $Z \sim \mathcal{N}(0, 1)$.

Geweke Diagnostic

We reject the null hypothesis when $|Z| > 2$.

We should pick the smallest burn-in (length of section A) such that $|Z| \leq 2$.

We can use coda's geweke.diag function for this.

Geweke Diagnostic

Let's take a look at the radiata pine example.

Other Convergence Diagnostics

After a pilot run, the *Raftery-Lewis diagnostic* will recommend how many iterations to run and how many to burn-in when the goal is to estimate quantiles of the posterior. It does this using a single MCMC chain.

Heidelberg-Welch proposed to test the null hypothesis that the sampled values come from a stationary distribution using the Cramer-von-Mises statistic.

Thinning

Even after convergence, autocorrelation is a problem.

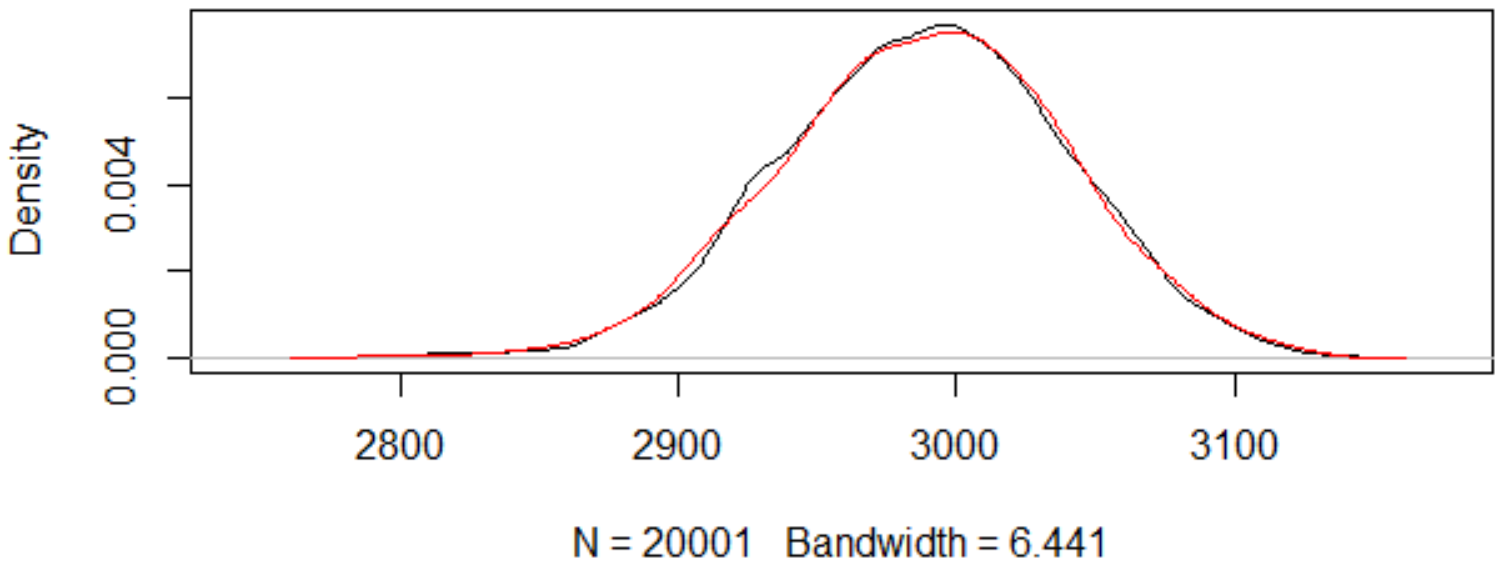
Thinning is the practice of only storing and using every k^{th} draw of the Markov chain.

Thinning reduces the effect of autocorrelation on density plots, is good for storage requirements and can be efficient if evaluation of the function g is very expensive.

HOWEVER, we should not generally do thinning because it generally increases the variance in the Monte Carlo estimates.

Thinning

```
plot(density(samples[,1]),main="")
lines(density(samples[seq(10,M+1,by=10),1]),col='red')
```



Summary

The answer to which samples we should use is:

- We use all samples after convergence, except when doing density plots
- We find the minimum burn-in M such that we have no evidence of non-convergence in the remaining samples. We check convergence using
  - Trace plots for multiple chains
  - $\hat{R}$ diagnostic for multiple chains
  - Geweke diagnostic for a single chain
  - Other options...

Sample Quality

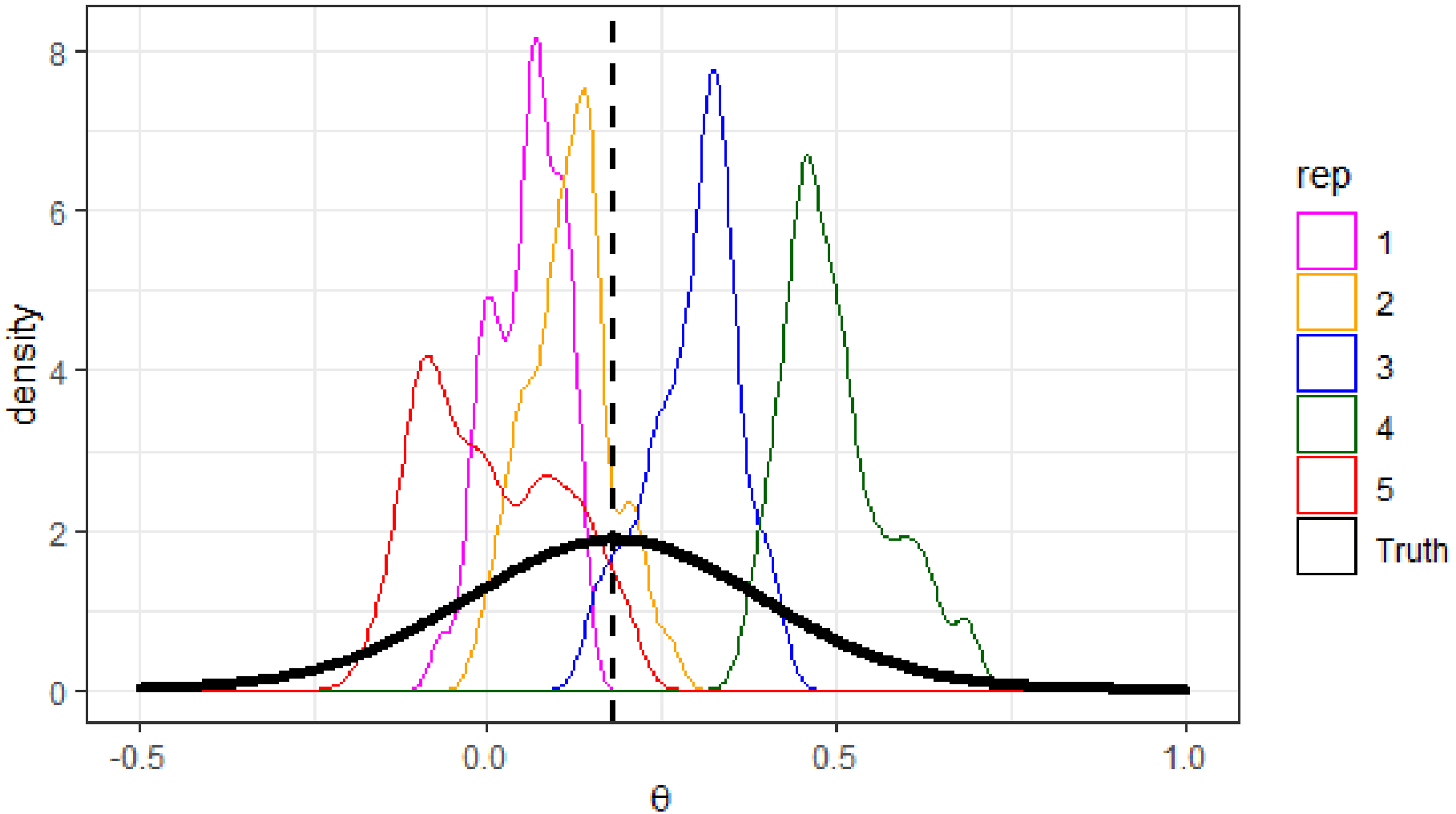
Sample Quality

We've decided that we've reached stationarity, but how reliable are our estimates? What else can go wrong?

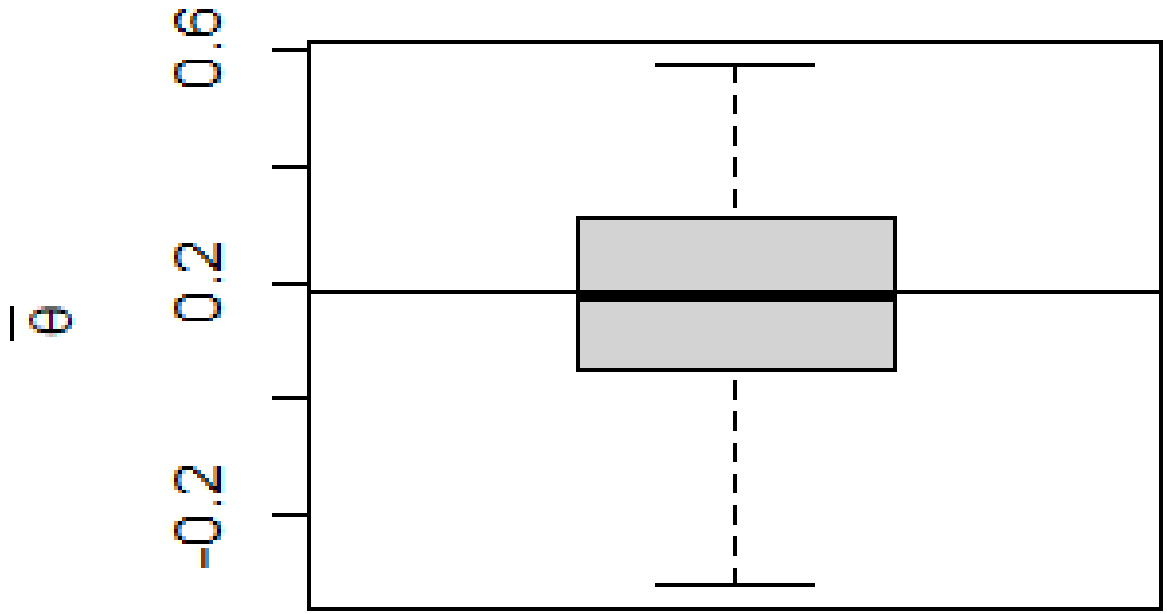
Let's say we've performed 1000 MCMC iterations, starting with a draw from the posterior.

We know that, on average, estimates of our expectation should be correct. But that doesn't mean that we can trust any single run:

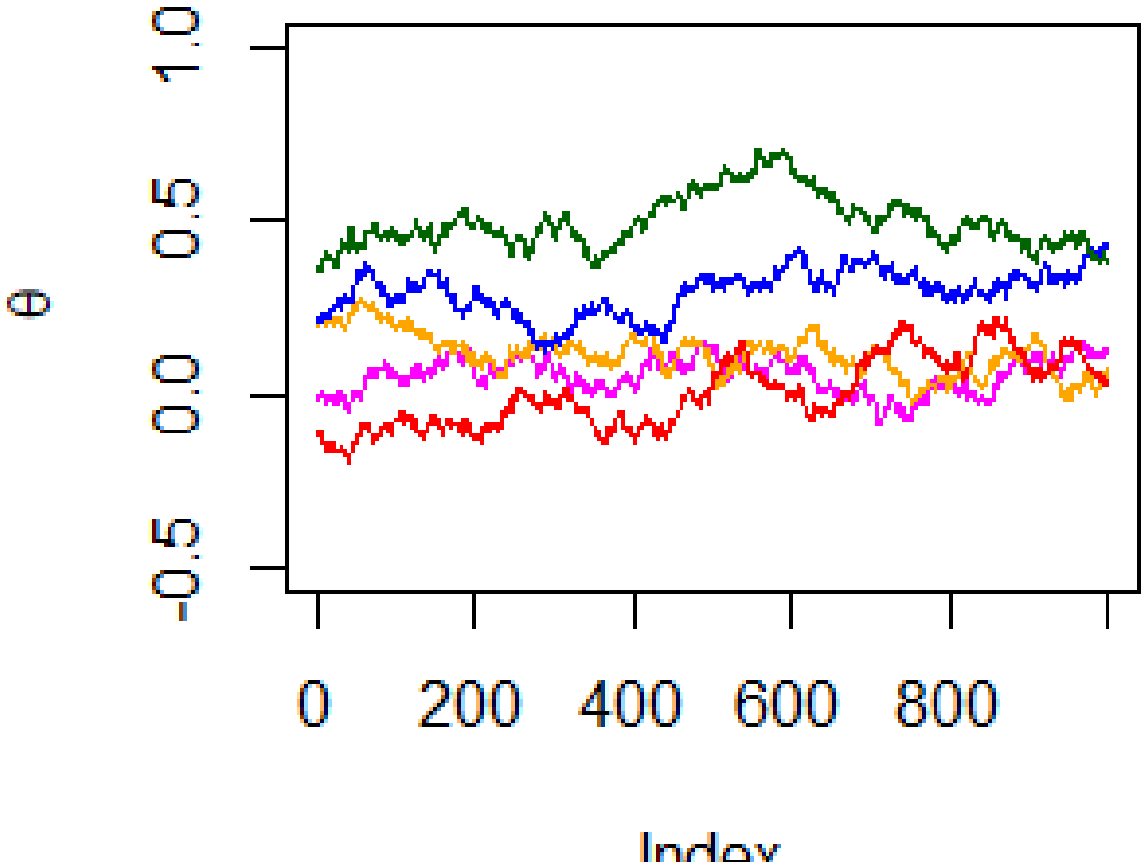
Sample Quality



Sample Quality



Sample Quality



Sample quality

Poor performance despite convergence is due to poor *mixing*, in other words inefficient exploration of the posterior.

This could be due to

- Making small jumps
- Making large jumps that are often rejected
- Proposals that struggle to get into certain regions (e.g. rarely jump modes)

Sample quality

The approaches we will use to assess sample quality are

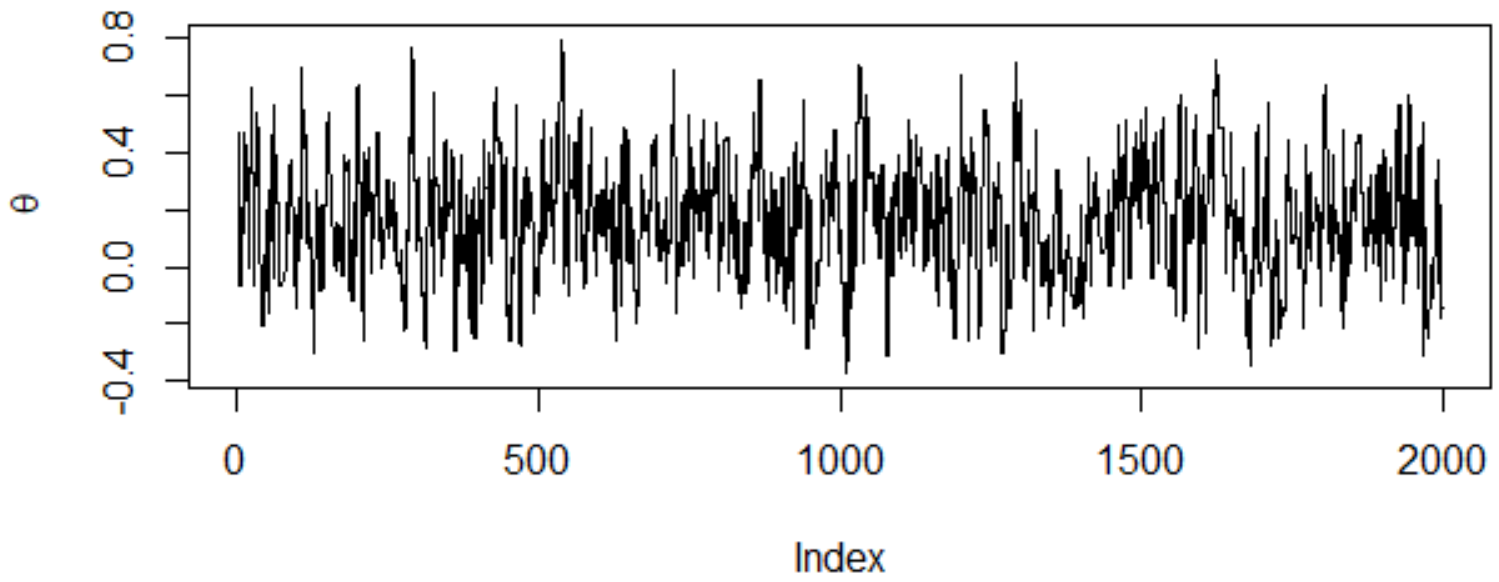
- Trace plots
- Autocorrelation plots
- Effective sample size (ESS)

These methods can be used to assess whether the samples are good enough and to compare different tuning parameters and methods. WARNING! These methods assume you have reached stationarity. You can't use these to assess convergence.

The 'kernel Stein discrepancy' from the next lecture can also be used to compare approaches.

Trace plots

We should look at trace plots for a variety of functions. We want to see a hairy caterpillar-like figure, e.g.



RWM

Set $\theta^{(0)}$.
For $t = 1, ..., T$

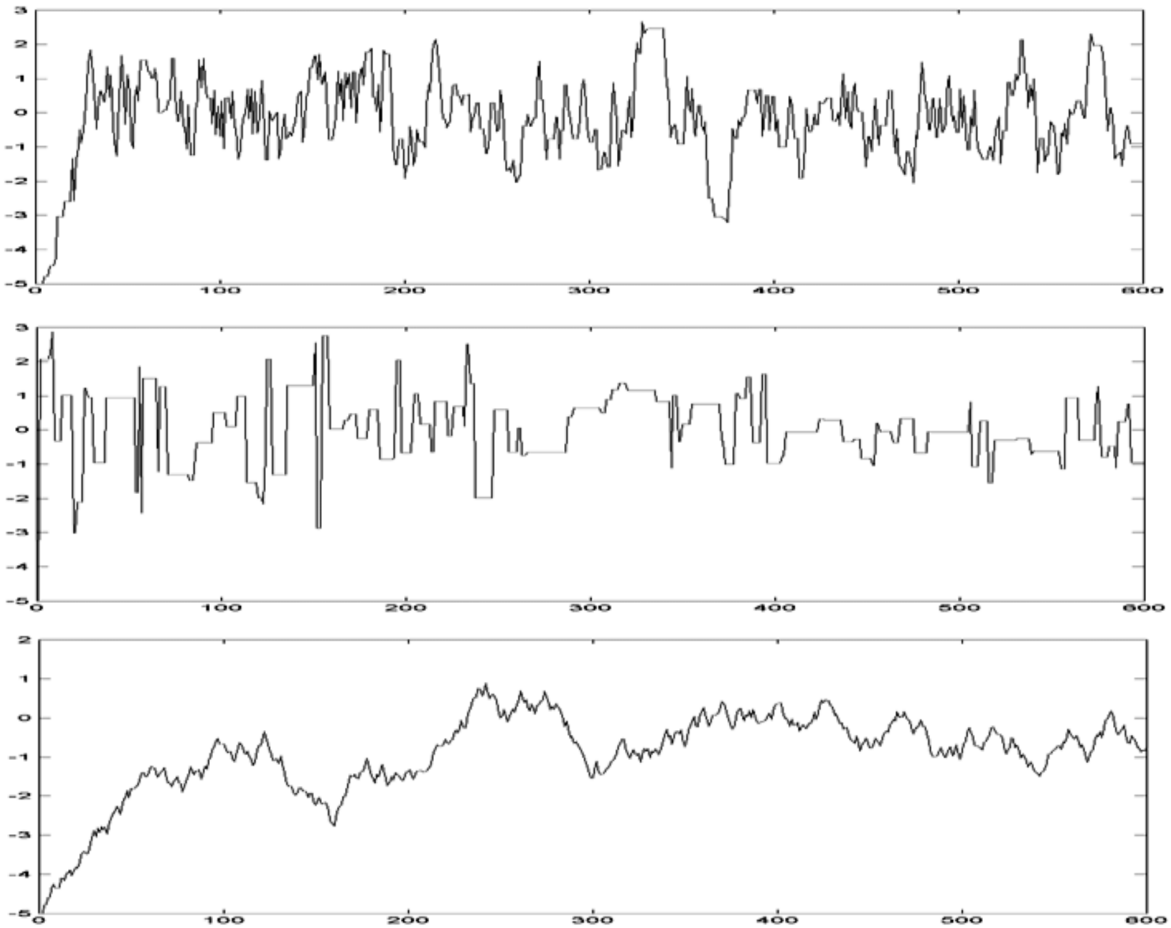
- Propose $\theta^* \sim \mathcal{N}(\theta^{(t-1)}, h^2\Sigma)$
- Compute MH acceptance probability

$$\alpha_{\text{HM}} = \min\left(1, \frac{\ell(\theta^*|y)p(\theta^*)}{\ell(\theta^{(t-1)}|y)p(\theta^{(t-1)})}\right)$$

- Draw $u \sim \text{Unif}[0, 1]$
  If $u < \alpha_{\text{MH}}$ set $\theta^{(t)} = \theta^*$
  otherwise set $\theta^{(t)} = \theta^{(t-1)}$

- If h is too high we don't accept often but move far
- If h is low we accept often but don't move far

RWM - Trace Plots



Gibbs - Slow Mixing

Set $\theta^{(0)}$.

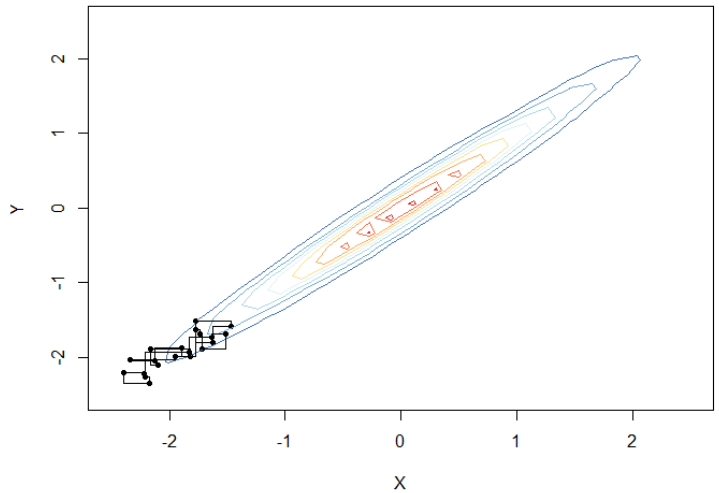
For $t = 1, \ldots, T$

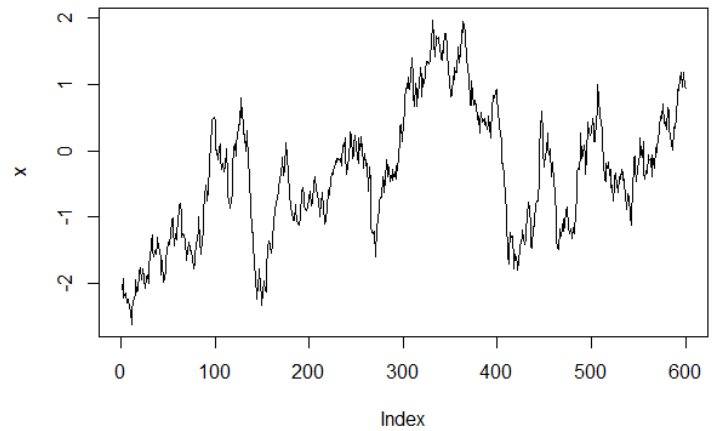
Sample $\theta_1^{(t)} \sim \pi(\theta_1 | \theta_2^{(t-1)}, \theta_3^{(t-1)}, \ldots, \theta_d^{(t-1)})$

Sample $\theta_2^{(t)} \sim \pi(\theta_2 | \theta_1^{(t)}, \theta_3^{(t-1)}, \ldots, \theta_d^{(t-1)})$

$\vdots$

Sample $\theta_d^{(t)} \sim \pi(\theta_p | \theta_1^{(t)}, \theta_2^{(t)}, \ldots, \theta_{d-1}^{(t)})$





Gibbs - Good Mixing

Set $\theta^{(0)}$.

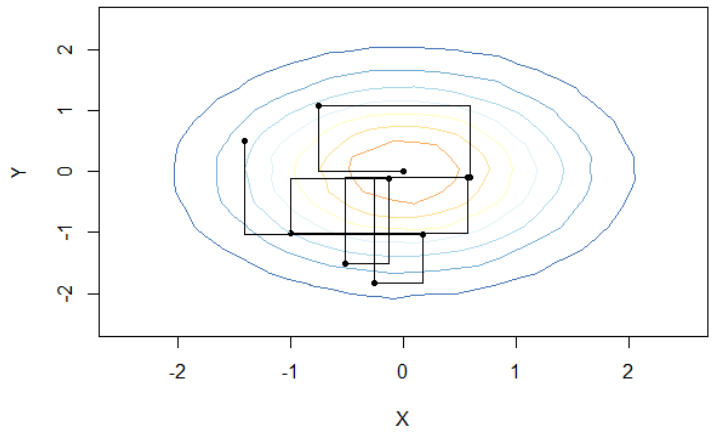
For $t = 1, \ldots, T$

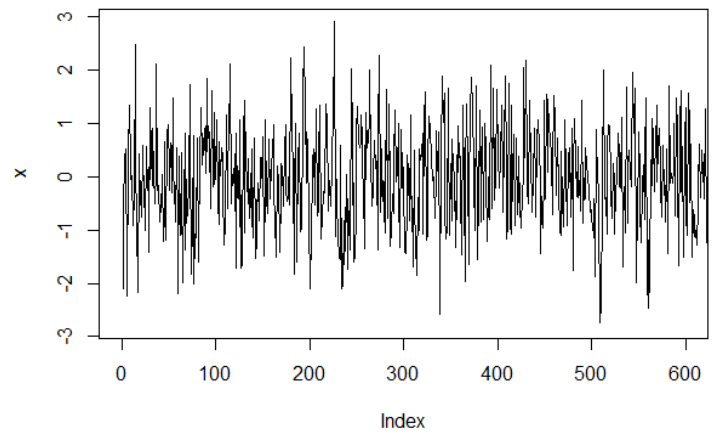
Sample $\theta_1^{(t)} \sim \pi(\theta_1 | \theta_2^{(t-1)}, \theta_3^{(t-1)}, \ldots, \theta_p^{(t-1)})$

Sample $\theta_2^{(t)} \sim \pi(\theta_2 | \theta_1^{(t)}, \theta_3^{(t-1)}, \ldots, \theta_p^{(t-1)})$

$\vdots$

Sample $\theta_p^{(t)} \sim \pi(\theta_p | \theta_1^{(t)}, \theta_2^{(t)}, \ldots, \theta_{p-1}^{(t)})$





Trace plots

Let's take a look at the radiata pine example.

Autocorrelation Plots

High autocorrelation is an indicator of poor mixing.

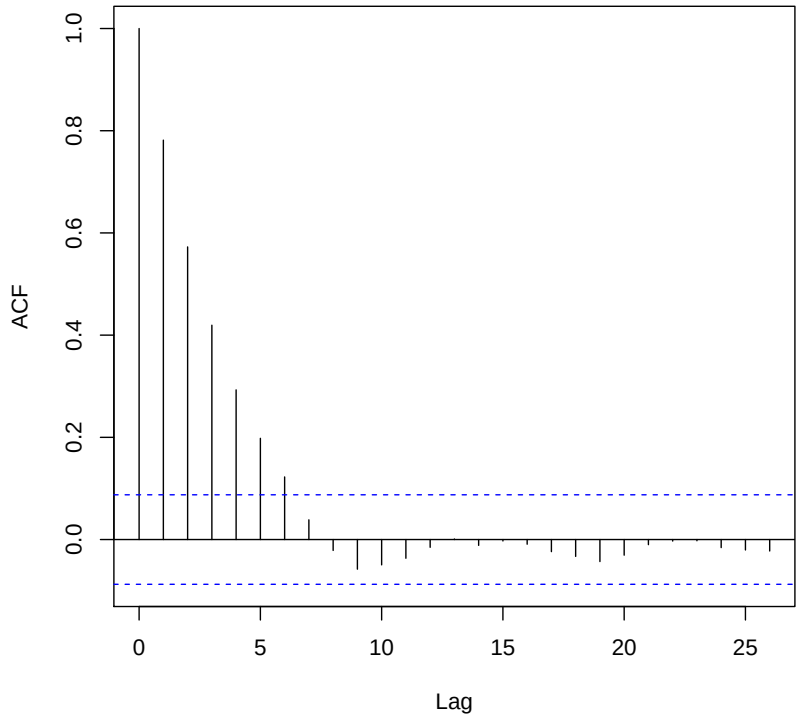
The k^{th} lag autocorrelation between every draw and its k^{th} lag can be estimated by

$$\hat{\rho}_k = \frac{\sum_{t=0}^{T-k}(\theta^{(t)} - \bar{\theta})(\theta^{(t+k)} - \bar{\theta})}{\sum_{t=0}^{T}(\theta^{(t)} - \bar{\theta})^2}$$

Use plot(acf(x)) in R.

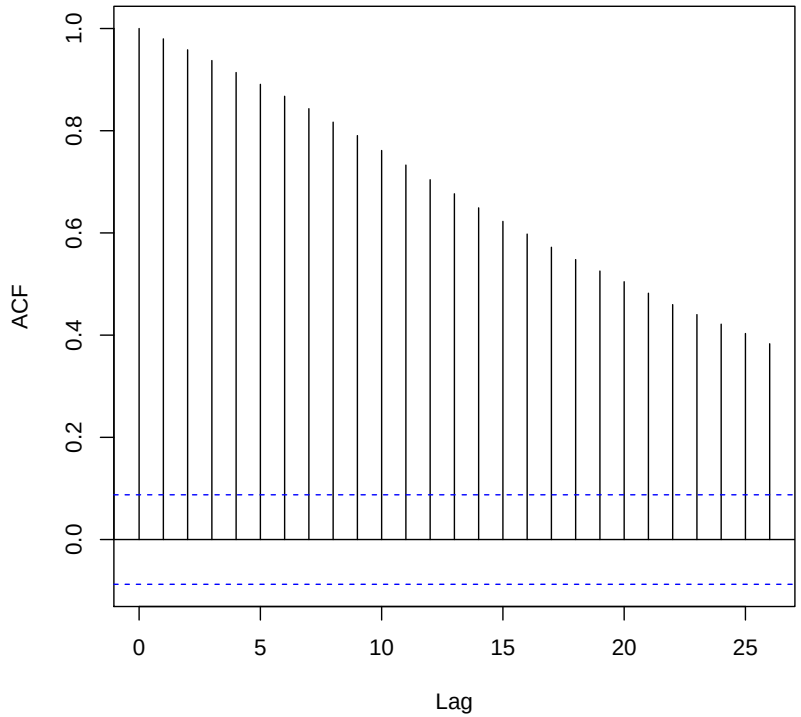
Autocorrelation Plots

GOOD MIXING: The autocorrelation decays quickly.



Autocorrelation Plots

WORSE MIXING: The autocorrelation decays slowly.



Autocorrelation Plots

Let's take a look at the radiata pine example.

Effective Sample Size

The ESS is an estimate of the equivalent number of independent iterations that the chain represents:

$$\text{ESS} = \frac{T}{1 + 2\sum_{k=1}^{\infty} \rho_k},$$

where T is the original sample size and ρ_k is the autocorrelation at the k^{th} lag.

There is no hard threshold on the required ESS. I usually like an ESS of at least 100 but this will depend on the complexity, and dimension of your posterior and your computational resources.

Effective Sample Size

This formula can be used parameter-by-parameter and for other functions such as $\log \ell(y|\theta) + \log p(\theta)$. To be conservative, look at the minimum ESS.

Use effectiveSize(x) in the `coda` package.

Recently, a multivariate ESS has been developed by Vats et al (2019)[2] (available in `mcmcse` `R` package).

[2] Vats, D., Flegal, J. M., & Jones, G. L. (2019). Multivariate output analysis for Markov chain Monte Carlo. Biometrika, 106(2), 321-337.

Effective Sample Size

Let's take a look at the radiata pine example.

Summary

Once we've removed the burn-in, we can assess sample quality by looking at

- Trace plots
- The (minimum) ESS across all parameters

These methods are most useful as comparative tools.

Biased MCMC

MCMC is Expensive

Standard MCMC is expensive for big data and other applications.

Recall that the posterior for independent observations is

$$\pi(\theta|y) \propto \prod_{t=1}^{T} \ell(y^{(t)}|\theta)p(\theta)$$

This is expensive to compute for big n.

Allowing for Some Bias

What if we can't afford to run standard MCMC for long enough? In this situation, we often resort to biased alternatives:

- Using some pre-convergence samples
- Using a biased but fast MCMC method that doesn't have π as its limiting distribution
- Using another fast but biased mechanism for sampling

We want to perform a bias-variance tradeoff.

Post-Processing for Biased MCMC

The post-processing methods we've covered today break down.

For example, in biased MCMC

- We never achieve convergence so assessing convergence and removing a burn-in don't make sense.
- We can't trust the ESS because it assumes we're sampling from π and only gives us information about autocorrelation.
- The vanilla estimate $\mathbb{E}_\pi[g(\theta)]$ will be biased.

Using Gradients

The next lecture will discuss new methods that can be applied to standard MCMC, biased MCMC and beyond.

These methods use additional information about π through the gradients

$$\nabla_\theta \log \pi(\theta|y) = \nabla_\theta \log \ell(y|\theta) + \nabla_\theta \log p(\theta),$$

which are often available in closed form or they can be unbiasedly estimated for big (independent data).

These gradients are also very useful in sampling and they appear in many standard and biased MCMC methods.

Metropolis-Adjusted Langevin Algorithm

An alternative to RW MH-MCMC is the Metropolis-Adjusted Langevin Algorithm:

Set $\theta^{(0)}$.

For $t = 1, ..., T$

- Propose $\theta^* \sim \mathcal{N}(\theta^{(t-1)} + h^2 \Sigma \nabla_\theta \log \pi(\theta^{(t-1)}|y)/2, h^2\Sigma)$
- Compute MH acceptance probability

$$\alpha_{\text{HM}} = \min\left(1, \frac{\ell(\theta^*|y)p(\theta^*)\mathcal{N}(\theta^{(t-1)}; \theta^* + h^2\Sigma\nabla_\theta \log \pi(\theta^*|y)/2, h^2\Sigma)}{\ell(\theta^{(t-1)}|y)p(\theta^{(t-1)})\mathcal{N}(\theta^*; \theta^{(t-1)} + h^2\Sigma\nabla_\theta \log \pi(\theta^{(t-1)}|y)/2, h^2\Sigma)}\right)$$

- Draw $u \sim \text{Unif}[0, 1]$
  If $u < \alpha_{\text{MH}}$ set $\theta^{(t)} = \theta^*$
  otherwise set $\theta^{(t)} = \theta^{(t-1)}$

Unadjusted Langevin Algorithm

The unadjusted Langevin algorithm (ULA) is a biased MCMC alternative:

Set $\theta^{(0)}$.
For $t = 1, ..., T$

- $\theta^{(t)} \sim \mathcal{N}(\theta^{(t-1)} + h^2 \Sigma \nabla_\theta \log \pi(\theta^{(t-1)}|y)/2, h^2\Sigma)$

For a fixed h, ULA is not π-invariant. We have no bias in the limit as $h \to 0$ and $T \to \infty$. We call this method stochastic gradient Langevin dynamics (SGLD) when we used unbiased estimates of $\nabla_\theta \log \pi(\theta|y)$.

Breakdown of ESS Illustration

We have different problems in ULA (biased MCMC) compared to MALA (standard MCMC):

MALA

- Small h leads to increased autocorrelation
- Large h leads to increased autocorrelation (rejections)

ULA

- Small h leads to increased autocorrelation
- Large h leads to bias

Breakdown of ESS Illustration

Figure 1 from Nemeth & Fearnhead (2021)[3]:

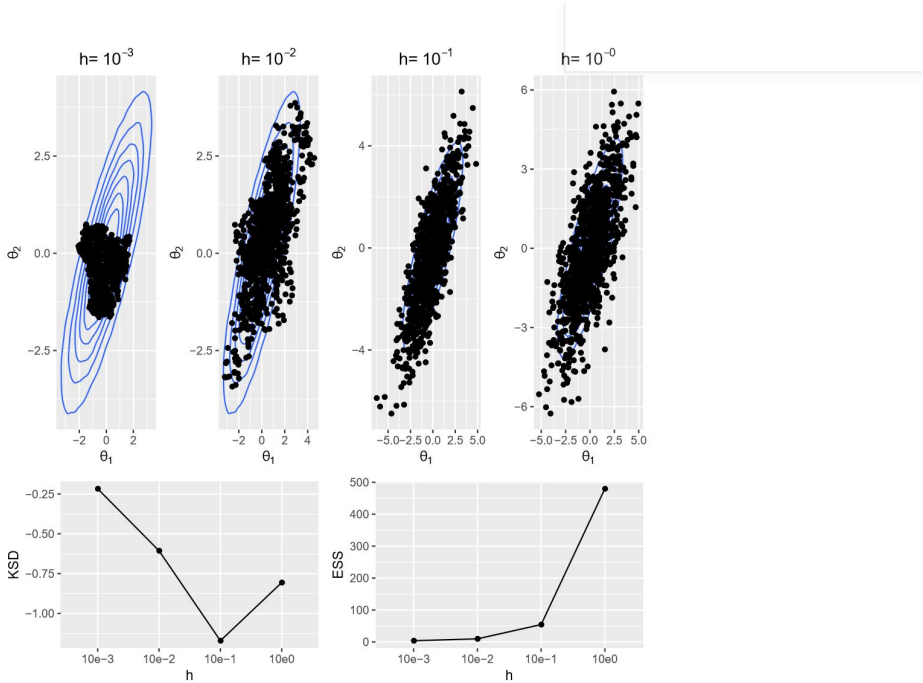


Figure 1: Top: Samples generated from the Langevin dynamics (7) are plotted over the bivariate Gaussian target. The samples are thinned to 1,000 for the ease of visualisation. Bottom: The kernel Stein discrepancy (log10) and effective sample size are calculated for each Markov chain with varying step size parameter h.

[3]Nemeth, C., & Fearnhead, P. (2021). Stochastic gradient Markov chain Monte Carlo. Journal of the American Statistical Association, 116(533), 433-450.

Teaser

Next lecture we will cover

- Kernel Stein discrepancy and related methods for assessing sample quality
- Stein-based estimates of $\mathbb{E}_\pi[g(\theta)]$ - some are bias-correcting!
- Stein thinning for selecting a subset of samples

Additional Resources

- Art Owen's Monte Carlo book: `https://statweb.stanford.edu/~owen/mc`
- Roy, V. (2020). Convergence diagnostics for Markov chain Monte Carlo. Annual Review of Statistics and Its Application, 7, 387-412.
- South, LF, Riabiz, M, Teymur, O & Oates, CJ. 2022. Post-Processing of MCMC. Annual Review of Statistics and Its Application. 9: Submitted.