UiO : Matematisk institutt

Det matematisk-naturvitenskapelige fakultet

STK-4051/9051 Computational Statistics Spring 2021
Markov Chain Monte Carlo part 2

Instructor: Odd Kolbjørnsen, oddkol@math.uio.no

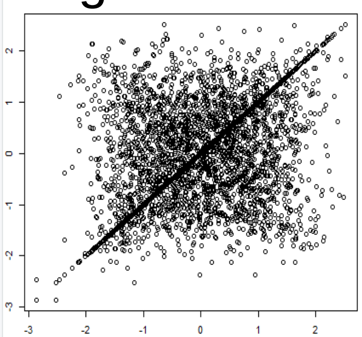
Last time McMC

- Markov chain Monte Carlo
  - Make a Markov P(**y**|**x**) chain with f(**x**) as limiting distribution / stationary distribution
  - Markov chain discrete/continuous
  - Irreducible, recurrent, aperiodic
  - Detailed balance $f(\mathbf{x})P(\mathbf{y}|\mathbf{x}) = f(\mathbf{y})P(\mathbf{x}|\mathbf{y})$
  - Metropolis-Hastings algorithm

$$\mathbf{y} \neq \mathbf{x} \quad P(\mathbf{y}|\mathbf{x}) = g(\mathbf{y}|\mathbf{x}) \min\left\{1, \frac{f(\mathbf{y})g(\mathbf{x}|\mathbf{y})}{f(\mathbf{x})g(\mathbf{y}|\mathbf{x})}\right\}$$

  - Random walk g(y|x)=h(y-x), h() symmetric
  - Independent sampler g(y|x)=h(y)
  - M-H and multivariate settings

Lag 1 scatter



Metropolis Hastings is a general form:

- Specific chains:
  - Random walk chain
  - Independent chain
  - Gibbs sampler
- Tricks to customize sampling
  - Augmentation
  - Reparametrize
  - Hybrid
  - Block update
  - Griddy-Gibbs

M-H and multivariate settings

- $\mathbf{X} = (X_1, \ldots, X_p)$
- Typical in this case: Only change one or a few components at a time.
  1. Choose index j (randomly)
  2. Sample $X_j^* \sim g_j(\cdot|\mathbf{x})$, put $X_k^* = X_k$ for $k \neq j$
  3. Compute
  $$R(\mathbf{x}, \mathbf{X}^*) = \frac{f(\mathbf{X}^*)g(\mathbf{x}|\mathbf{X}^*)}{f(\mathbf{x})g(\mathbf{X}^*|\mathbf{x})}$$
  4. Put
  $$\mathbf{Y} = \begin{cases} \mathbf{X}^* & \text{with probability } \min\{1, R(\mathbf{x}, \mathbf{X}^*)\} \\ \mathbf{x} & \text{otherwise} \end{cases}$$
- Can show that this version also satisfies detailed balance
- Can even go through indexes systematic
  - Should then consider the whole loop through all components as one iteration

Gibbs sampler

- Assume $\mathbf{X} = (X_1, \ldots, X_p)$
- Aim: Simulate $\mathbf{X} \sim f(\mathbf{x})$
- Gibbs sampling:
  1. Select starting values $\mathbf{x}^{(0)}$ and set $t = 0$
  2. Generate, in turn

$$\begin{aligned}
X_1^{(t+1)} &\sim f(x_1 | x_2^{(t)}, x_3^{(t)}, \ldots, x_p^{(t)}) \\
X_2^{(t+1)} &\sim f(x_2 | x_1^{(t+1)}, x_3^{(t)}, \ldots, x_p^{(t)}) \\
&\vdots \\
X_{p-1}^{(t+1)} &\sim f(x_{p-1} | x_1^{(t+1)}, \ldots, x_{p-2}^{(t+1)}, x_p^{(t)}) \\
X_p^{(t+1)} &\sim f(x_p | x_1^{(t+1)}, \ldots, x_{p-1}^{(t+1)})
\end{aligned}$$

  3. Increment t and go to step 2.
- Completion of step 2 is called a cycle

Example capture recapture

- Aim: Estimate population size, N, of a species
- Procedure:
  - At time t_1: Catch $c_1 = m_1$ individuals, each with probability α_1. Mark and release
  - At time $t_i, i > 1$: Catch c_i individuals, each with probability α_i. Count number of newly caught individuals, m_i, mark the unmarked and release all
- Likelihood:
  - At time t_1:

$$\Pr(C_1 = c_1) = \Pr(C_1 = m_1) = \binom{N}{m_1}\alpha_1^{m_1}(1-\alpha_1)^{N-m_1}$$

c = catch
m = mark (new)

Note: N is a parameter
no proportionality trick ☺

Capture recapture cont…

- At time t_i, $i > 1$ (number of marked individuals are $\sum_{k=1}^{i-1} m_k$)

$$
\begin{aligned}
\Pr(C_i = & c_i, M_j = m_i | N, \mathbf{c}_{1:i-1}, \mathbf{m}_{1:i-1}) \\
& = \Pr(C_i = c_i | N) \Pr(M_i = m_i | N, C_i = c_i, \mathbf{m}_{1:i-1})
\end{aligned}
$$

$$
= \binom{N}{c_i} \alpha_i^{c_i} (1 - \alpha_i)^{N - c_i} \frac{\binom{N - \sum_{k=1}^{i-1} m_k}{m_i} \binom{\sum_{k=1}^{i-1} m_k}{c_i - m_i}}{\binom{N}{c_i}}
$$

$$
= \alpha_i^{c_i} (1 - \alpha_i)^{N - c_i} \binom{N - \sum_{k=1}^{i-1} m_k}{m_i} \binom{\sum_{k=1}^{i-1} m_k}{c_i - m_i}
$$

Capture recapture cont…

- Likelihood:

$$L(N,\boldsymbol{\alpha}|\mathbf{c},\mathbf{m}) \propto \binom{N}{m_1}\alpha_1^{m_1}(1-\alpha_1)^{N-m_1}\times$$

$$\prod_{i=2}^{I}\alpha_j^{c_i}(1-\alpha_i)^{N-c_i}\binom{N-\sum_{k=1}^{i-1}m_k}{m_i}\binom{\sum_{k=1}^{i-1}m_k}{c_i-m_i}$$

$$\propto \prod_{i=1}^{I}\alpha_i^{c_i}(1-\alpha_i)^{N-c_i}\binom{N-\sum_{k=1}^{i-1}m_k}{m_i}$$

$$\propto \binom{N}{\sum_{k=1}^{I}m_k}\prod_{i=1}^{I}\alpha_i^{c_i}(1-\alpha_j)^{N-c_i}$$

$$\boxed{\frac{N!}{(N-m_1)!\,m_1!}\cdot\frac{(N-m_1)!}{(N-m_1-m_2)!\,m_2!}\cdots\frac{\left(N-\sum_{k=1}^{l-1}m_k\right)!}{\left(N-\sum_{k=1}^{l}m_k\right)!\,m_l!}\propto\frac{N!}{\left(N-\sum_{k=1}^{l}m_k\right)!}\propto\binom{N}{\sum m_k}}$$

- Prior:

$$f(N)\propto 1$$

$$f(\alpha_i|\theta_1,\theta_2)\sim\text{Beta}(\theta_1,\theta_2)$$

The conditional distribution of α_i

- Prior:

$$f(\alpha_i|\theta_1,\theta_2) \sim \text{Beta}(\theta_1,\theta_2) \quad \propto \alpha_i^{\theta_1-1} \; (1-\alpha_i)^{\theta_2-1}$$

Likelihood:

$$\propto \binom{N}{\sum_{k=1}^{l} m_k} \prod_{k=1}^{l} \alpha_i^{c_i} \, (1-\alpha_i)^{N-c_i}$$

Everything except α_i is constant!

Posterior:

$$\propto \alpha_i^{\theta_1-1} \; (1-\alpha_i)^{\theta_2-1} \cdot \alpha_i^{c_i} \; (1-\alpha_i)^{N-c_i}$$

$$\propto \alpha_i^{\theta_1+c_i-1} \; (1-\alpha_i)^{\theta_2+N-c_i-1}$$

The conditional distribution of N

- Prior:

$$f(N) \propto 1$$

Likelihood:

$$\propto \binom{N}{\sum_{k=1}^{l} m_k} \prod_{i=1}^{l} \alpha_i^{c_i} (1-\alpha_i)^{N-c_i}$$

Everything except N is constant!

Posterior:

$$\propto \binom{N}{\sum_{k=1}^{l} m_k} \prod_{k=1}^{l} \alpha_i^{c_i} (1-\alpha_i)^{N-c_i}$$

$$\propto \binom{N}{\sum_{k=1}^{l} m_k} \prod_{k=1}^{l} (1-\alpha_i)^{N} \propto \binom{N}{\sum_{k=1}^{l} m_k} \left(\prod_{k=1}^{l} (1-\alpha_i) \right)^{N}$$

Binomial

$$\Pr(X=k) = \binom{n}{k} p^k q^{n-k}$$

Negative binomial $k = n - r \geq 0$

$$\Pr(X=n) = \binom{n-1}{k} p^r (1-p)^k$$

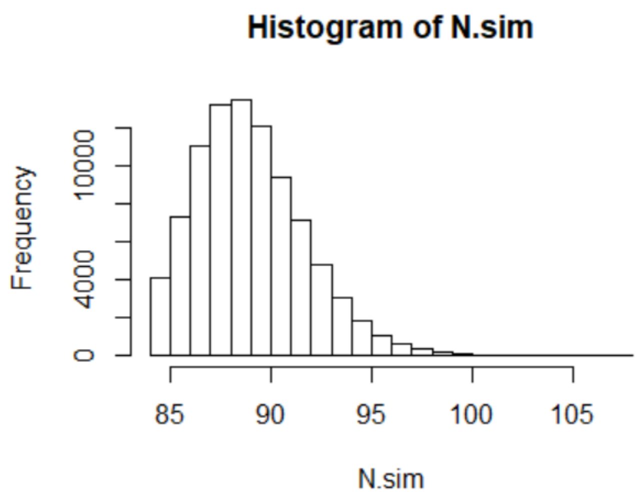
Capture recapture cont…

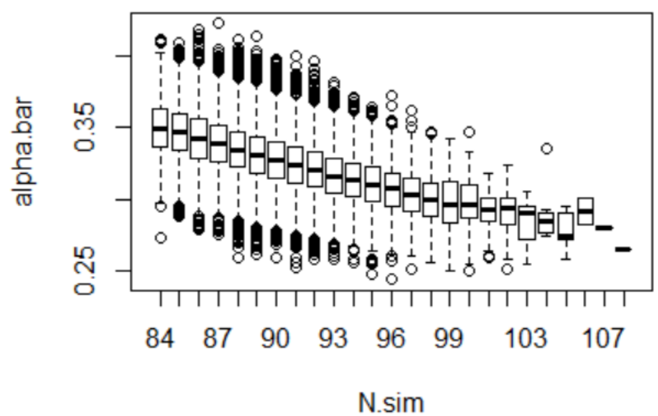
- Can derive ($r = \sum_{k=1}^{I} m_k$):

$$N|\boldsymbol{\alpha}, \mathbf{c}, \mathbf{m} \sim r + \texttt{NegBinom}(r+1, 1-\prod_{i=1}^{I}(1-\alpha_i))$$

$$\alpha_i|N, \boldsymbol{\alpha}_{-i}, \mathbf{c}, \mathbf{m} \sim \text{Beta}(c_i + \theta_1, N - c_i + \theta_2)$$

- `Example_7_6.R`





Properties of Gibbs sampler-random scan

- Gibbs sampling (random scan):
  1. Select starting values $\mathbf{x}^{(0)}$ and set $t = 0$
  2. Sample $j \sim \text{Uniform}\{1, ..., p\}$
  3. Sample $X_j^{(t+1)} \sim f(x_j|\mathbf{x}_{-j}^{(t)})$
  4. Put $X_k^{(t+1)} = X_k^{(t)}$ for $k \neq j$
- The chain $\{\mathbf{X}^{(t)}\}$ is Markov
- Detailed balance:
  - Consider $\mathbf{x}, \mathbf{x}^*$ where $x_j \neq x_j^*$ while $x_k = x_k^*$ for $k \neq j$

Pr(x_j is changed)

Proposal density given x_j is changed

$$
\begin{aligned}
f(\mathbf{x})P(\mathbf{x}^*|\mathbf{x}) =& f(\mathbf{x})p^{-1}f(x_j^*|\mathbf{x}_{-j}) \\
=& f(\mathbf{x}_{-j})f(x_j|\mathbf{x}_{-j})p^{-1}f(x_j^*|\mathbf{x}_{-j}) \\
=& f(\mathbf{x}^*_{-j})f(x_j|\mathbf{x}^*_{-j})p^{-1}f(x_j^*|\mathbf{x}^*_{-j}) \\
=& f(\mathbf{x}^*)p^{-1}f(x_j|\mathbf{x}^*_{-j}) \\
=& f(\mathbf{x}^*)P(\mathbf{x}|\mathbf{x}^*)
\end{aligned}
$$

$\boxed{\mathbf{x}_{-j} = \mathbf{x}^*_{-j}}$

Gibbs sampler-deterministic scan

- Gibbs sampling (deterministic scan):
  1. Select starting values $\mathbf{x}^{(0)}$ and set $t = 0$
  2. Generate, in turn

$$\begin{aligned}
X_1^{(t+1)} &\sim f(x_1 | x_2^{(t)}, x_3^{(t)}, \ldots, x_p^{(t)}) \\
X_2^{(t+1)} &\sim f(x_2 | x_1^{(t+1)}, x_3^{(t)}, \ldots, x_p^{(t)}) \\
&\vdots \\
X_{p-1}^{(t+1)} &\sim f(x_{p-1} | x_1^{(t+1)}, \ldots, x_{p-2}^{(t+1)}, x_p^{(t)}) \\
X_p^{(t+1)} &\sim f(x_p | x_1^{(t+1)}, \ldots, x_{p-1}^{(t+1)})
\end{aligned}$$

  3. Increment t and go to step 2.
- The chain $\{\mathbf{X}^{(t)}\}$ is Markov
- Do not fulfill detailed balance (going backwards will revert order of components visited)
- Will still satisfy

$$f(\mathbf{x}^*) = \int_{\mathbf{x}} f(\mathbf{x}) P(\mathbf{x}^* | \mathbf{x}) d\mathbf{x}$$

Proof d=2

- Assume $p = 2$: $P(\mathbf{x}^*|\mathbf{x}) = f(x_1^*|x_2)f(x_2^*|x_1^*)$:

$$\int_{\mathbf{x}} f(\mathbf{x})P(\mathbf{x}^*|\mathbf{x})d\mathbf{x} = \int_{x_2}\int_{x_1} f(\mathbf{x})f(x_1^*|x_2)f(x_2^*|x_1^*)dx_1\,dx_2$$

$$= \int_{x_2}\int_{x_1} f(x_1|x_2)\boxed{f(x_2)f(x_1^*|x_2)}f(x_2^*|x_1^*)dx_1\,dx_2$$

$$= \int_{x_2}\int_{x_1} f(x_1|x_2)f(x_2|x_1^*)\boxed{f(x_1^*)f(x_2^*|x_1^*)}dx_1\,dx_2$$

$$= \boxed{f(x_1^*, x_2^*)}\int_{x_2} f(x_2|x_1^*)\underbrace{\int_{x_1} f(x_1|x_2)dx_1}_{=1}dx_2$$

$$= f(x_1^*, x_2^*)\underbrace{\int_{x_2} f(x_2|x_1^*)dx_2}_{=1}$$

$$= f(x_1^*, x_2^*) = f(\mathbf{x}^*)$$

- Proof similar for general p

We start again 14.15

Metropolis Hastings is a general form:

- Specific chains:
  - Random walk chain
  - Independent chain
  - Gibbs sampler
- Tricks to customize sampling
  - Reparametrize
  - Augmentation
  - Hybrid
  - Block update
  - Griddy-Gibbs

Reparameterization

- Sometimes easier to transform variables to another scale $Y = h^{-1}(X)$
- Avoid boundary effects (Exercise 7.1)
- May improve convergence (Exercise 7.8)
- Two approaches (identical results)
  - Possible to work directly in transformed space
  - Run the MCMC in X-space, but construct proposal through $X^* = h(Y^*)$

Reparameterization version 1

- Possible to work directly in transformed space
  - Need to transform target distribution

$$f_Y(y) = f_X(h(y))|h'(y)|$$

$$R(y, y^*) = \frac{f_X(h(y^*))|h'(y^*)|g_y(y|y^*)}{f_X(h(y))|h'(y)|g_y(y^*|y)}$$

$$= \frac{f_X(x^*)|h'(y^*)|g_y(y|y^*)}{f_X(x)|h'(y)|g_y(y^*|y)}$$

- Given sample Y, easy to obtain sample $X = h(Y)$

Reparameterization version 2, 1D

- Run the MCMC in X-space, but construct proposal through $X^* = h(Y^*)$
  - Need to transform proposal distribution

$$g_x(x^*|x) = g_y(h^{-1}(x^*)|h^{-1}(x)) \cdot |(h^{-1})'(x^*)|$$

$$R(x, x^*) = \frac{f_x(x^*) \cdot g_y\,(h^{-1}(x)|h^{-1}(x^*)) \cdot |(h^{-1})'(x^*)|}{f_x(x) \cdot g_y(h^{-1}(x^*)|h^{-1}(x)) \cdot |(h^{-1})'(x)|}$$

$$= \frac{f_X(x^*)g_Y(y|y^*)|h'(y^*)|}{f_X(x)g_Y(y^*|y)|h'(y)|}$$

since $(h^{-1})'(x) = 1/h'(y)$

The derivative of the inverse function
with respect to the argument. $h(x) = y; x = h^{-1}(x)$

Hybrid Gibbs sampler

- If $f(x_j|\mathbf{x}_{-j})$ is difficult to sample from, use an Metropolis-Hastings step for this component
- Example ($p = 5$)
  1. Sample $X_1^{(t+1)} \sim f(x_1|\mathbf{x}_{-1}^{(t)})$
  2. Sample $(X_2^{(t+1)}, X_3^{(t+1)})$ through an M-H step
  3. Sample $X_4^{(t+1)}$ through another M.H step
  4. Sample $X_5^{(t+1)} \sim f(x_5|\mathbf{x}_{-5}^{(t+1)})$

Capture-recapture - extended approach

- Assume now a prior $f(\theta_1, \theta_2) \propto \exp\{-(\theta_1 + \theta_2)/1000\}$
- Conditional distributions:

$$N|\cdot \sim r + \texttt{NegBinom}(r+1, 1-\prod_{i=1}^{I}(1-\alpha_i))$$

$$\alpha_i|\cdot \sim \text{Beta}(c_i + \theta_1, N - c_i + \theta_2)$$

$$(\theta_1, \theta_2)|\cdot \sim \underbrace{k\left[\frac{\Gamma(\theta_1+\theta_2)}{\Gamma(\theta_1)\Gamma(\theta_2)}\right]^I \prod_{i=1}^{I} \alpha_i^{\theta_1}(1-\alpha_i)^{\theta_2}\exp\left\{-\frac{\theta_1+\theta_2}{1000}\right\}}_{\text{Sample using M.H}}$$

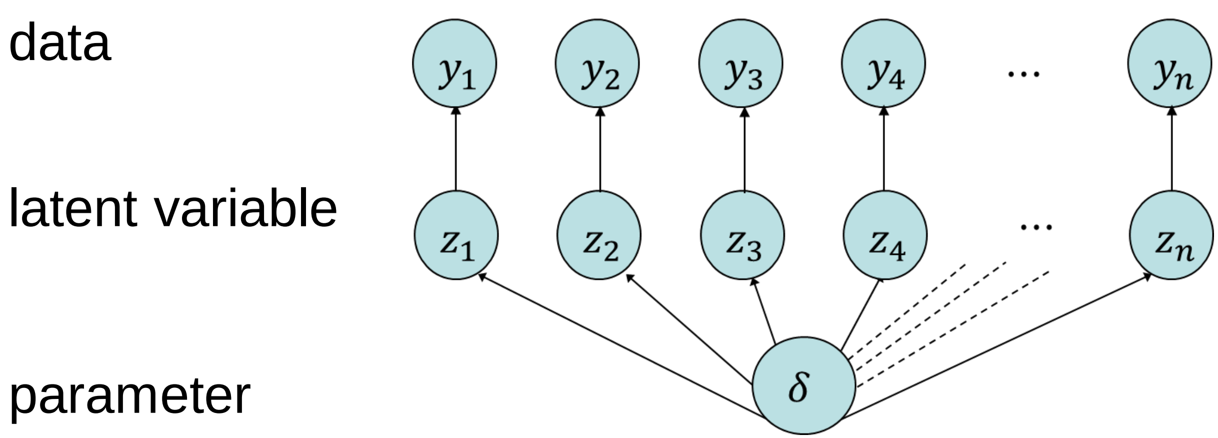
- `Example_7_7.R`

= Hybrid Gibbs sampler

Variable augmentation

- It is difficult or impossible to sample δ directly, but there exists a “latent variable” z such that it is possible to conditionally sample $z|\delta$ and $\delta|z$

Example



Example mixture distribution (augmenting)

- Mixture distribution

$$Y \sim f(y) = \delta\phi(y, \mu_0, 0.5) + (1-\delta)\phi(y, \mu_1, 0.5), \quad \mu_0 = 7, \mu_1 = 10$$

- Prior $\delta \sim \text{Uniform}[0, 1]$
- Aim: Simulate $\delta \sim p(\delta|y_1, ..., y_n)$

$$p(\delta|y_1, ..., y_n) \propto \prod_{i=1}^{n} [\delta\phi(y_i, 7, 0.5) + (1-\delta)\phi(y_i, 10, 0.5)]$$

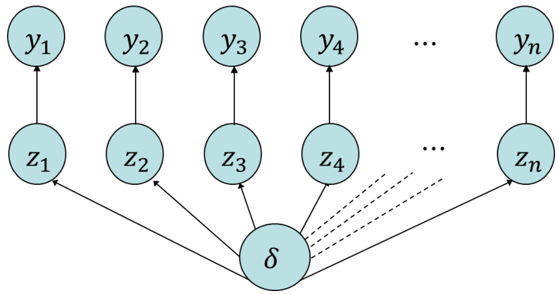
Difficult to simulate from directly

- Note, can write model for Y by

$$\Pr(Z = z) = \delta^{1-z}(1-\delta)^z, \qquad z = 0, 1$$

$$Y|Z = z \sim \phi(y, \mu_z, 0.5), \qquad \mu_0 = 7, \mu_1 = 10$$

Augmenting the variable set with z (similar to EM-algorithm)



- Note:

$$\begin{aligned} p(\delta|y_1, ..., y_n, z_1, ..., z_n) &\propto \prod_{i=1}^{n} \delta^{1-z_i}(1-\delta)^{z_i}\phi(y_i, \mu_{z_i}, 0.5) \\ &\propto \delta^{n-\sum_{i=1}^n z_i}(1-\delta)^{\sum_{i=1}^n z_i} \\ &\propto \text{Beta}(\delta, n - \sum_{i=1}^n z_i + 1, \sum_{i=1}^n z_i + 1) \end{aligned}$$

Example mixture distribution cont…

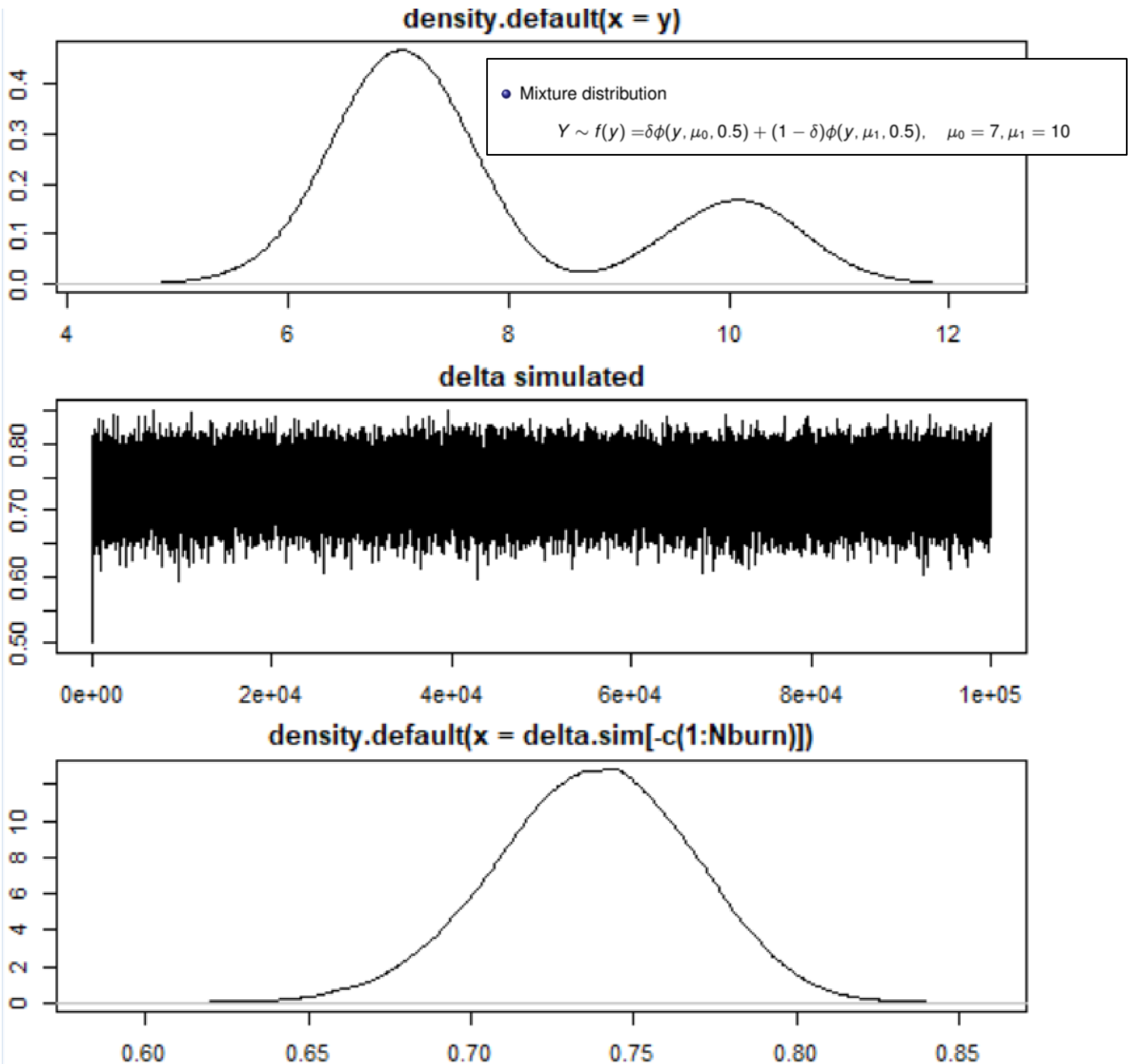
- Conditional distribution for **z**:

$$\begin{aligned} p(\mathbf{z}|\delta,\mathbf{y}) &\propto p(\delta)p(\mathbf{z}|\delta)p(\mathbf{y}|\mathbf{z},\delta)\\ &\propto \prod^{n} \delta^{1-z_i}(1-\delta)^{z_i}\phi(y_i,\mu_{z_i},0.5) \end{aligned}$$

- Independence between z_i's:

$$\begin{aligned} \Pr(Z_i=z_i|\delta,y_i) &\propto \delta^{1-z_i}(1-\delta)^{z_i}\phi(y_i,\mu_{z_i},0.5)\\ &\propto \begin{cases} \frac{\delta\phi(y_i,\mu_0,0.5)}{\delta\phi(y_i,\mu_0,0.5)+(1-\delta)\phi(y_i,\mu_1,0.5)} & z_i=0\\ \frac{(1-\delta)\phi(y_i,\mu_1,0.5)}{\delta\phi(y_i,\mu_0,0.5)+(1-\delta)\phi(y_i,\mu_1,0.5)} & z_i=1 \end{cases} \end{aligned}$$

- Aim: Simulate $\delta \sim p(\delta|y_1,...,y_n)$
- Approach: Simulate from $p(\delta,\mathbf{Z}|y_1,...,y_n)$
- Gibbs sampling
  1. Initialize $\delta^{(0)}$, set $t=0$
  2. Simulate $\mathbf{Z}^{(t+1)} \sim p(\mathbf{z}|\delta^{(t)},\mathbf{y})$
  3. Simulate $\delta^{(t+1)} \sim p(\delta|\mathbf{z}^{(t+1)},\mathbf{y})$
  4. Increment t and go to step 2.


density.default(x = y)
Mixture distribution
$Y \sim f(y) = \delta\phi(y, \mu_0, 0.5) + (1-\delta)\phi(y, \mu_1, 0.5), \quad \mu_0 = 7, \mu_1 = 10$
0.0
0.1
0.2
0.3
0.4
4
6
8
10
12
delta simulated
0.50
0.60
0.70
0.80
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
density.default(x = delta.sim[-c(1:Nburn)])
0
2
4
6
8
10
0.60
0.65
0.70
0.75
0.80
0.85

Blocking/ Block update

- When dividing $X = (X_1, \ldots, X_p)$, each X_j can be vectors
- Making each X_j as large as possible will typically improve convergence
- Especially beneficial when high correlation between single components

Griddy Gibbs sampler

- Many variants
- Assume that one dimension is particularly hard to sample, i.e. $f(x_j|\boldsymbol{x}_{-j})$
- Simplest version of Griddy Gibbs:
  - Initialize: Sample $z_1, z_2, \ldots, z_n$ from $g(z)$
  - Per iteration:
    - Compute weights $w_j^{(t)} \propto f(z_j|\boldsymbol{x}_{-j}^{(t)})/g(z_j)$
    - Sample $x_j^{(t)}|\boldsymbol{x}_{-j}^{(t)} \sim (z_j, w_j^{(t)})$
- Need to keep $z_1, z_2, \ldots, z_n$ fixed through iterations

Convergence issues of MCMC

- Theoretical properties:

$$\mathbf{X}^{(t)} \stackrel{\mathcal{D}}{\rightarrow} f(\mathbf{x})$$

$$\hat{\theta}_1 = \frac{1}{L}\sum_{t=1}^{L} h(\mathbf{X}^{(t)}) \rightarrow E^f[h(\mathbf{X})]$$

  as $t \rightarrow \infty$

- Note: We also have

$$\hat{\theta}_2 = \frac{1}{L}\sum_{t=D+1}^{D+L} h(\mathbf{X}^{(t)}) \rightarrow E^f[h(\mathbf{X})]$$

  - Advantage: Remove those variables with distribution very different from $f(\mathbf{x})$
  - Disadvantage: Need more samples
- Question: How to specify D and L?
  - D: Large enough so that $\mathbf{X}^{(t)} \approx f(\mathbf{x})$ for $t > D$ (bias small)
  - L: Large enough so that $\text{Var}[\hat{\theta}_2]$ is small enough

Mixing

- For $\hat{\theta} = \frac{1}{L}\sum_{t=D+1}^{D+L} h(\mathbf{X}^{(t)})$:

$$\text{Var}[\hat{\theta}] = \frac{1}{L^2}\left[\sum_{t=D+1}^{D+L} \text{Var}[h(\mathbf{X}^{(t)})] + 2\sum_{s=D+1}^{D+L-1}\sum_{t=s+1}^{D+L} \text{Cov}[h(\mathbf{X}^{(s)}), h(\mathbf{X}^{(t)})]\right]$$

  Assume D large, so "converged":

$$\text{Var}[h(\mathbf{X}^{(t)})] \approx \sigma_h^2, \quad \text{Cov}[h(\mathbf{X}^{(s)}), h(\mathbf{X}^{(t)})] \approx \sigma_h^2\rho(t-s)$$

  gives

$$\begin{aligned}\text{Var}[\hat{\theta}] \approx& \frac{1}{L^2}\left[\sum_{t=D+1}^{D+L} \sigma_h^2 + 2\sum_{s=D+1}^{D+L-1}\sum_{t=s+1}^{D+L} \sigma_h^2\rho(t-s)\right]\\ =& \frac{\sigma_h^2}{L}\left[1 + 2\sum_{k=1}^{L-1}\frac{L-k}{L}\rho(k)\right]\end{aligned}$$

- Good mixing: $\rho(k)$ decreases fast with k!

Effective sample size for MCMC

- For $\hat{\theta} = \frac{1}{L}\sum_{t=D+1}^{D+L} h(\mathbf{X}^{(t)})$:

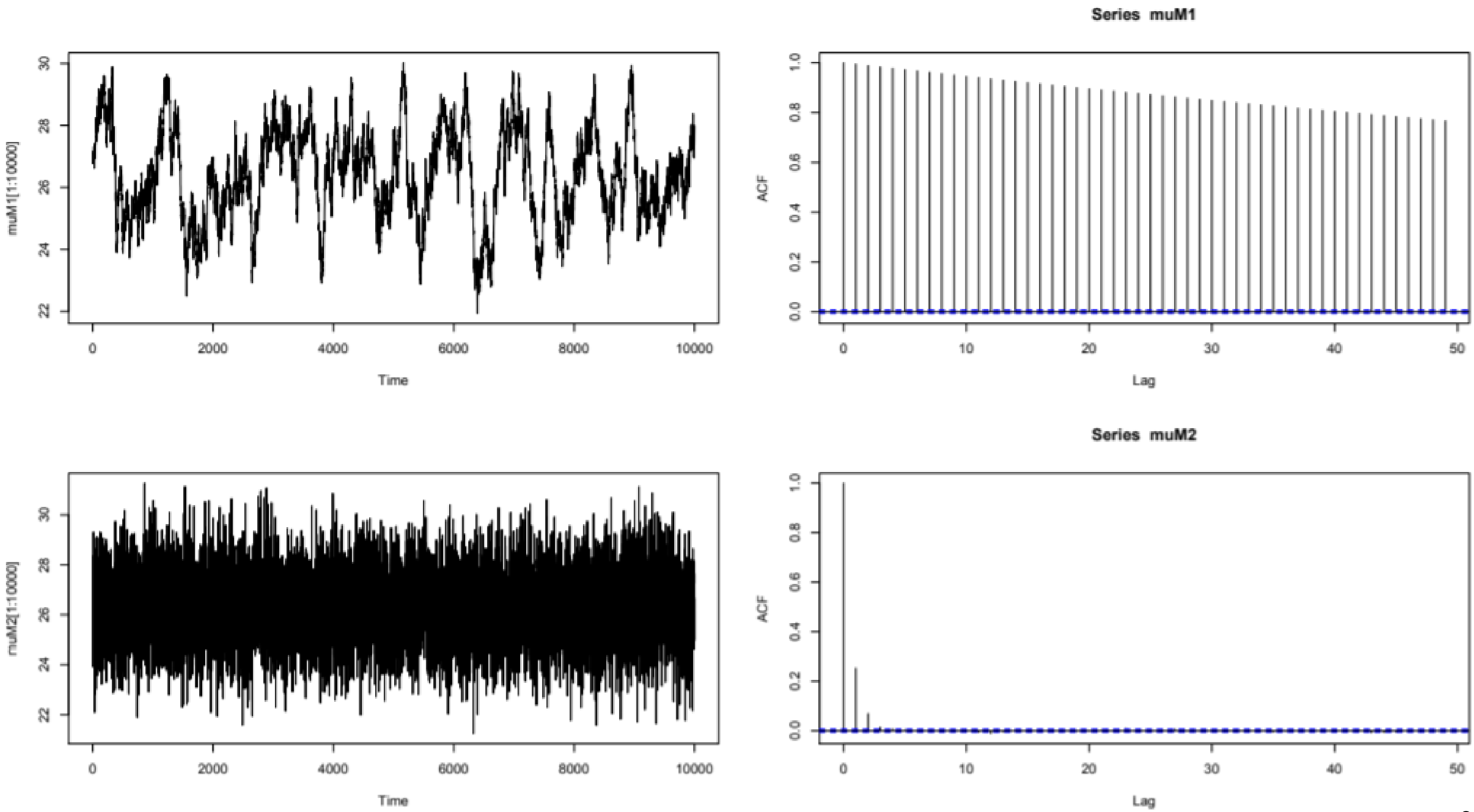
$$\text{Var}[\hat{\theta}] = \frac{\sigma_h^2}{L}\left[1 + 2\sum_{k=1}^{L-1}\frac{L-k}{L}\rho(k)\right] \stackrel{L\to\infty}{\to} \frac{\sigma_h^2}{L}[1 + 2\sum_{k=1}^{\infty}\rho(k)]$$

- If independent samples:

$$\text{Var}[\hat{\theta}] = \frac{\sigma_h^2}{L}$$

- Effective sample size: $\boxed{\frac{L}{1+2\sum_{k=1}^{\infty}\rho(k)}}$
- Use empirical estimates $\hat{\rho}(k)$
- Usual to truncate the summation when $\hat{\rho}(k) < 0.1$.

Example from exercise 7.8

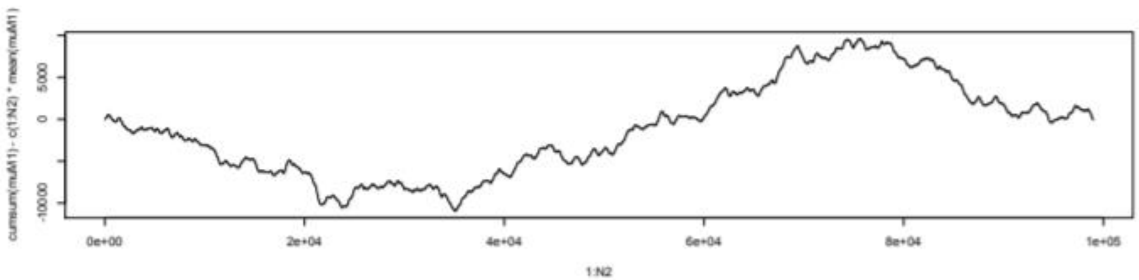


How to assess convergence

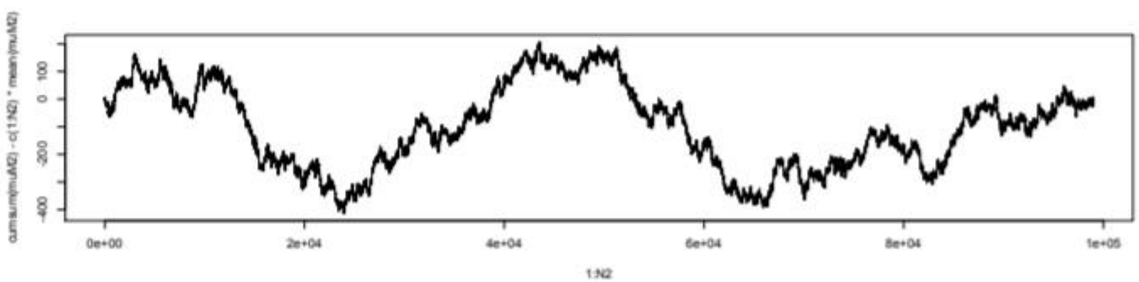
- Graphical diagnostics:
  - Sample paths:
    - Plot $h(\mathbf{X}^{(t)})$ as function of t
    - Useful with different $h(\cdot)$ functions!
  - Cusum diagnostics
    - Plot $\sum_{i=1}^{t}[h(\mathbf{X}^{(i)}) - \hat{\theta}_n]$ versus t
    - Wiggly and small excursions from 0: Indicate chain is mixing well

Note: We can have the situation that:
- some variables mix well
- other have bad mixing

Bad



Better



The Gelman-Rubin diagnostic

- Motivated from analysis of variance
- Assume J chains run in parallel
- jth chain: $x_j^{(D+1)}, \ldots, x_j^{(D+L)}$ (first D discarded)
- Define

$$\bar{x}_j = \frac{1}{L}\sum_{t=D+1}^{D+L} x_j^{(t)} \qquad \bar{x}_{\cdot} = \frac{1}{J}\sum_{j=1}^{J} \bar{x}_j$$

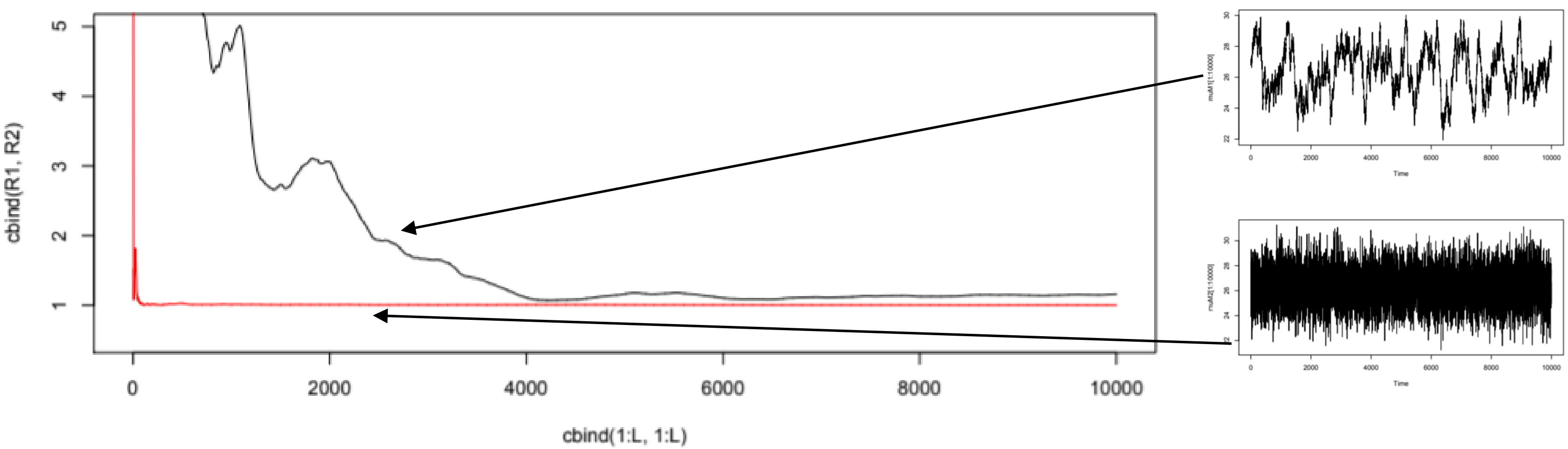
$$B = \frac{L}{J-1}\sum_{j=1}^{J}(\bar{x}_j - \bar{x}_{\cdot})^2$$

$$W = \frac{1}{J}\sum_{j=1}^{J} s_j^2 \qquad s_j^2 = \frac{1}{L-1}\sum_{t=D+1}^{D+L}(x_j^{(t)} - \bar{x}_j)^2$$

- If converged, both B and W estimates $\sigma^2 = \text{Var}_f[X]$
- Diagnostic: $R = \frac{\frac{L-1}{L}W + \frac{1}{L}B}{W}$
- "Rule": $\sqrt{R} < 1.1$ indicate D and L are sufficient

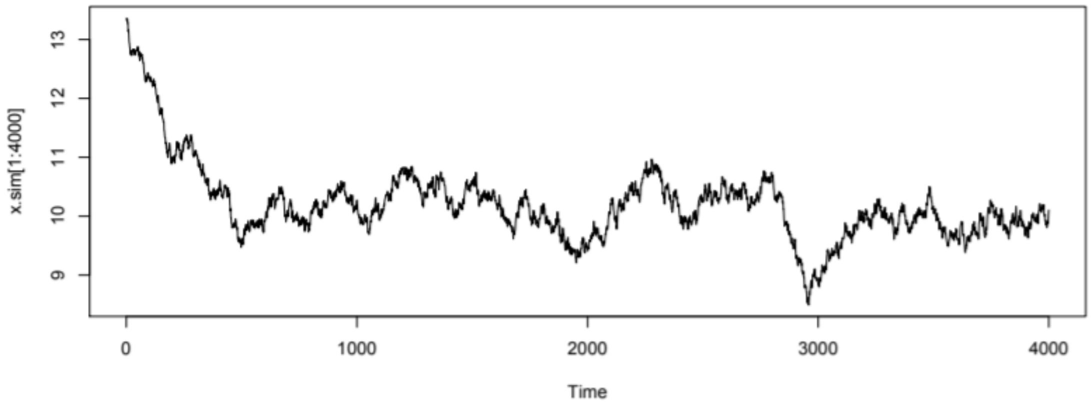
Example: Exercise 7.8

- $D = 100, L = 1000$: $\sqrt{R_1} = 1.588$, $\sqrt{R_2} = 1.002$,
- $D = 1000, L = 1000$: $\sqrt{R_1} = 1.700$, $\sqrt{R_2} = 1.004$,
- $D = 1000, L = 10000$: $\sqrt{R_1} = 1.049, \sqrt{R_2} = 1.0008$

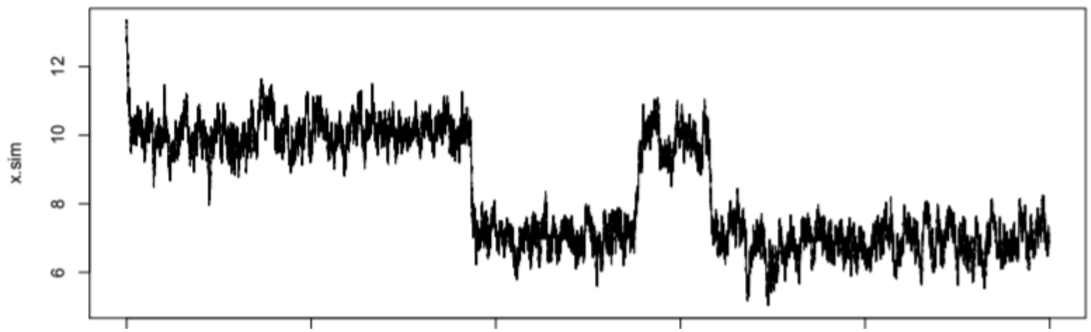


Apparent convergence

- $f(x) = 0.7 \cdot N(7, 0.5^2) + 0.3 \cdot N(10, 0.5^2)$
- Metropolis-Hastings with proposal $N(x^{(t)}, 0.05^2)$
- First 4000 samples (400 discarded)



- Full 10000 samples



Choices

My experience:
Random is robust
You should rather spend time improving other parts of code

- Gibbs sampler
  - Random or deterministic scan?
    - Deterministic scan most common (?)
    - When high correlation, random scan can be more efficient
- Independence chain:
  - $g(\cdot) \approx f(\cdot)$
  - High acceptance rate
  - Tail properties most important
  - f /g should be bounded
- Random walk proposal
  - Tune variance so that acceptance rate is between 25% and 50%

Number of chains

- Assume possible to perform N iterations
  - One long chain of length N, or
  - J parallel chains, each of length N/J?
- Burnin:
  - One long chain: Only need to discard D samples
  - Parallel chains: Need to discard $J \cdot D$ samples
- Check of convergence
  - Easier with many parallel chains
- Efficiency
  - Parallel chains give more independent samples
- Computational issues
  - Possible to utilize multiple cores with parallel chains

Data uncertainty and Monte Carlo uncertainty

- Parameter: $\theta = E^f[h(\mathbf{X})]$
- Estimator: $\hat{\theta} = \frac{1}{L}\sum_{t=D+1}^{D+L} h(\mathbf{X}^{(t)})$:
- Two types of uncertainty
  - Variability in $h(\mathbf{X})$: $\sigma_h^2 = \text{Var}^f[h(\mathbf{X})]$
    - Estimator: $\hat{\sigma}_h^2 = \frac{1}{L}\sum_{t=D+1}^{D+L}[h(\mathbf{X}^{(t)}) - \hat{\theta}]^2$
  - MC variability in $\hat{\theta}$:
    - Estimator: Divide data into batches of size $b = \lfloor L^{1/a} \rfloor$, make estimates $\hat{\theta}$ within each batch and variance from these
- Recommendation: Specify L so that MC variability is less than 5% of variability in $h(\mathbf{X})$.

Next time

- Advanced topics in MCMC
- Presentation of part 2 compulsory