Lecture 8: Brief Reminder

We are learning about the MCMC algorithm:

- What it is and why does it work
- Elements of the algorithm:
  - Target distribution $\pi(x)$ - Problem determined
  - Proposal distribution $Q(y|x)$ - Choosen by us
  - Acceptance probability $\alpha(y|x)$ - Computed s.t. the detailed balance holds
- Mild conditions guarantee the convergence of the algorithm but no the convergence rate!
- We have looked at two special proposal densities:
  - The independence proposal $Q(y|x) = Q(y)$
  - The RW proposal $Q(y|x) = Q(x|y)$
- Importance of the tuning parameter

Review: Special cases Metropolis-Hastings

- Metropolis algorithm: The proposal density is symmetric around the current value, that means

$$Q(x_{i-1}|y) = Q(y|x_{i-1}).$$

Hence,

$$\alpha = \min\left(1, \frac{\pi(y)}{\pi(x_{i-1})} \times \frac{Q(x_{i-1}|y)}{Q(y|x_{i-1})}\right) = \min\left(1, \frac{\pi(y)}{\pi(x_{i-1})}\right)$$

- Independence sampler: The proposal distribution does not depend on the current value $\boldsymbol{x}_{i-1}$

$$Q(x|x_{i-1}) = Q(x).$$

$Q(x)$ is an approximation to $\pi(x) \Rightarrow$ acceptance rate should be high.

MCMC and iterative conditioning

The use of the MH-algorithms gains on importance when it is applied iteratively on components of x.

Let $\boldsymbol{x}$ be decomposed by several (for simplicity scalar) components.

$$\boldsymbol{x} = (x^1, \ldots, x^p)$$

Now the MH-algorithm is applied iteratively on the components x^j, conditioning on the current values of $\boldsymbol{x}^{-j}$ with

$$\boldsymbol{x}^{-j} = (x^1, \ldots, x^{j-1}, x^{j+1}, \ldots, x^p)$$

MCMC and iterative conditioning

To be concrete, one uses

- a proposal kernel $Q(y^j | x_{i-1}^j, \boldsymbol{x}_{i-1}^{-j})$, $j = 1, \ldots, p$.
- with acceptance probability

$$\alpha = \min \left(1, \frac{\pi(y^j | \boldsymbol{x}_{i-1}^{-j})}{\pi(x_{i-1}^j | \boldsymbol{x}_{i-1}^{-j})} \times \frac{Q(x_{i-1}^j | y^j, \boldsymbol{x}_{i-1}^{-j})}{Q(y^j | x_{i-1}^j, \boldsymbol{x}_{i-1}^{-j})} \right)$$

This algorithm converges to the stationary distribution with density $\pi(\boldsymbol{x})$, as long as all components are arbitrary often updated.

Conditional densities

Of note, the acceptance probability α only uses the full conditional densities $\pi(x^j|\boldsymbol{x}^{-j})$, $j = 1, \ldots, p$, and not the joint density $\pi(\boldsymbol{x})$.
Both are related as follows

$$\pi(x^j|\boldsymbol{x}^{-j}) = \frac{\pi(\boldsymbol{x})}{\pi(\boldsymbol{x}^{-j})} \propto \pi(\boldsymbol{x})$$

Thus, the (non-normalised) conditional densities of $x^j|\boldsymbol{x}^{-j}$ can be directly derived from $\pi(\boldsymbol{x})$ by omitting all multiplicative factors, that do not depend on x^j.

Gibbs sampling

Are all conditional densities $\pi(x^j|\boldsymbol{x}^{-j})$, $j = 1, \ldots, p$ *standard* it seems natural to use those as proposal kernel, i.e.

$$Q(y^j|x^j_{i-1}, \boldsymbol{x}^{-j}_{i-1}) = \pi(x^j|\boldsymbol{x}^{-j}_{i-1})$$

In this case, we get $\alpha = 1$ which leads to the well known Gibbs sampler, which updates parameters iteratively by sampling from the corresponding full conditional distributions.

Gibbs sampling

Let $\mathbf{x} = (x^1, \ldots, x^n)$, $\mathbf{x} \sim \pi(\mathbf{x})$, N proposal distribution are defined by:

- propose $y^i \sim \pi(y^i | \mathbf{x}^{-i})$
- keep $y^k = x^k$ for $k \neq i$

Notation:

- $\mathbf{x} = (x^1, \ldots, x^n)$
- $\mathbf{x}^{-i} = (x^1, \ldots, x^{i-1}, x^{i+1}, \ldots, x^n)$
- $\mathbf{y} = (y^1, \ldots, y^n) = (x^1, \ldots, x^{i-1}, y^i, x^{i+1}, \ldots, x^n)$

Why is the acceptance probability always 1?

Gibbs-Sampling algorithm

Idea: Sequentially sampling from univariate conditional distributions (which are often available in closed form).

1. Select starting values $\boldsymbol{x}_0$ and set $i = 0$.
2. Repeatedly:

$$
\begin{aligned}
&\text{Sample} \quad x_{i+1}^1 | \cdot \sim \pi(x^1 | x_i^2, \ldots, x_i^p) \\
&\text{Sample} \quad x_{i+1}^2 | \cdot \sim \pi(x^2 | x_{i+1}^1, x_i^3, \ldots, x_i^p) \\
&\vdots \\
&\text{Sample} \quad x_{i+1}^{p-1} | \cdot \sim \pi(x^{p-1} | x_{i+1}^1, x_{i+1}^2, \ldots, x_{i+1}^{p-2}, x_i^p) \\
&\text{Sample} \quad x_{i+1}^p | \cdot \sim \pi(x^p | x_{i+1}^1, \ldots, x_{i+1}^{p-1})
\end{aligned}
$$

   where $|\cdot$ denotes conditioning on the most recent updates of all other elements of $\boldsymbol{x}$.

3. Increment i and go to step 2.

Example: Simple linear regression

Let

$$Y_i = a + bx_i + e_i, \;\; e_i \sim \mathcal{N}(0, 1/\tau), \;\; i = 1, \ldots, n$$

and

$$a \sim \mathcal{N}(0, 1/\tau_a)$$

$$b \sim \mathcal{N}(0, 1/\tau_b)$$

$$\tau \sim \text{Gamma}(\alpha, \beta)$$

we are interested in

$$\pi(a, b, \tau | \mathbf{y})$$

(Show R-code `demo_linear_reg_Gibbs.R`)

Remarks on Gibbs sampling

- High dimensional updates of $\boldsymbol{x}$ can be boiled down to scalar updates.
- Visiting schedule: Various approaches exist (and can be justified) to ordering the variables in the sampling loop. One approach is random sweeps: variables are chosen at random to resample.
- Gibbs sampling assumes that it is easy to sample from the full-conditional distribution. This is sometimes not so easy. Alternatively, a Metropolis-Hastings proposal can be used for the j-th component, i.e. Metropolis-within-Gibbs $\Rightarrow$ Hybrid Gibbs sampler.

Remarks on Gibbs sampling

- Blocking or grouping is possible, that means not all elements of $\boldsymbol{x}$ are treated individually. Might be useful when elements of $\boldsymbol{x}$ are correlated.
- Care must be taken when improper prior are used, which may lead to an improper posterior distribution. Impropriety implies that there does not exist a joint density to which the full-conditional distributions correspond.

  Hobert, J. P. and Casella, G. (1996), JASA, 91: 1461–1473.

Example : Conjugate gamma-Poisson hierarchical model

Example from George et al. (1993) regarding the analysis of 10 power plants.

- y_i number of failures of pump i
- t_i length of operation time of pump i (in kilo hours)

Model:

$$y_i \mid \lambda_i \sim \mathsf{Po}(\lambda_i t_i)$$

Conjugate prior for λ_i:

$$\lambda_i \mid \alpha, \beta \sim \mathsf{G}(\alpha, \beta)$$

Hyper-prior on α and β:

$$\alpha \sim \mathsf{Exp}(1.0) \qquad\qquad \beta \sim \mathsf{G}(0.1, 10.0)$$

Conjugate gamma-Poisson hierarchical model (II)

The posterior of the 12 parameters $(\alpha, \beta, \lambda_1, \ldots, \lambda_{10})$ given $y_1, \ldots, y_{10}$ is proportional to

$$\pi(\alpha, \beta, \lambda_1, \ldots, \lambda_{10} \mid y_1, \ldots, y_{10}) \propto \pi(\alpha)\pi(\beta) \prod_{i=1}^{10} [\pi(\lambda_i \mid \alpha, \beta)\pi(y_i \mid \lambda_i)]$$

$$\propto e^{-\alpha} \beta^{0.1-1} e^{-10\beta} \left\{ \prod_{i=1}^{10} \exp(-\lambda_i t_i) \lambda_i^{y_i} \right\} \left\{ \prod_{i=1}^{10} \exp(-\beta\lambda_i) \lambda_i^{\alpha-1} \right\} \left[\frac{\beta^\alpha}{\Gamma(\alpha)} \right]^{10}.$$

This posterior is not of closed form.

What are the full conditional distributions?

Update scheme for gamma-Poisson hierarchical model

For each iteration i

- For $k = 1, \ldots, 10$
  - Simulate new value $\lambda_k \sim \text{Gamma}(y_i + \alpha, t_i + \beta)$ Gibbs step
- Simulate new value $\beta \sim \text{Gamma}(10\alpha + 0.1, \sum \lambda_k + 1)$ Gibbs step
- Propose new value $\alpha_{new} \sim \mathcal{N}(\alpha_{i-1}, \tau)$ MH step
- Compute acceptance probability

$$a = \min \left\{ 1, \frac{\pi(\alpha_{new} | \ldots)}{\pi(\alpha_{old} | \ldots)} \right\}$$

- if $u < a$
  - set $\alpha_i = \alpha_{new}$
- else
  - set $\alpha_i = \alpha_{old}$

Blocking Strategies

Blocking (ie simulating some variables together) might improve the algorithm especially when variables are correlated.

Example: Korsbetningen

In the year of our Lord 1361, on the third day after S:t Jacob, the Goth fell outside the gates of Visby at the hands of the Danish. They are buried here. Pray for them.

- Archeoloical excavation found 493 femurs, 256 right and 237 left
- At least 256 person were buried here....but how many more??

A simple model

Let x_1 and x_2 be the number of left and right femurs found.

Assume x_1 and x_2 to be two independent observations from a Bin(N, ϕ) distribution.

With

- N total number of people buried
- ϕ probability of finding a femur, left or right

The unkown parameter vector is $\theta = (N, \phi)$. Assume a Beta(a, b) prior for ϕ, and a Unif$(256, 2500)$ prior for N.

Updating schemes

Single site update

- Simulate new $\phi \sim \text{Beta}(\cdot, \cdot)$ (Gibbs step)
- Propose $N_{new} \sim \text{Unif}(N_{old} - d, N_{old} + d)$
- Compute

$$\alpha = \min \left\{ 1, \frac{\pi(N_{new}|\dots)}{\pi(N_{old}|\dots)} \right\}$$

- Accept or reject the new value for N

Block update

- Propose a new value N_{new} for N from $\text{Unif}(N_{old} - d, N_{old} + d)$
- Propose a new value ϕ_{new} for ϕ from

$$\text{Beta}(\alpha + x_1 + x + 2, \beta + 2N_{new} - x_1 - x_2)$$

- Compute α
- Accept or reject N_{new} and ϕ_{new} simultaneously

(Show R-code `Vikings.R`)

Implementation and convergence diagnostics

Source: http://i.telegraph.co.uk/multimedia/archive/02365/coding_alamy_2365972b.jpg

Convergence

- If well constructed, the Markov chain is guaranteed to have the posterior as limiting distribution.
- However, this does not tell you how long you have to run the MCMC algorithm til convergence.
  - The initial position may have a big influence.
  - The proposal distribution may lead to low acceptance rates.
  - The chain may get caught in a local maximum of the likelihood surface.
- We say the Markov chain mixes well if it can
  - reach the posterior quickly, and
  - moves quickly around the posterior modes.

Convergence diagnostics

Valid inferences from sequences of MCMC outputs are based on the assumption that the outputs are from the desired target distribution.

- There is no overall minimum number of samples to ensure approximation.
- Consequently methods for testing convergence, known as convergence diagnostics, have to be applied.
- However it has to emphasised that these diagnostics do not guarantee convergence.

Trace plots

An initial possibility for deciding if a MCMC output does not converge to the desired posterior distributions is to look at the sample trace for each variable.

- If our chain is taking a long time to move around the parameter space, then it will take longer to converge.
- If the samples form a homogene band (no wave movements or other rare fluctuations), convergence might be indicated.
- Vastly different values at the beginning of the trace indicate burn-in iterations, which should be discarded.

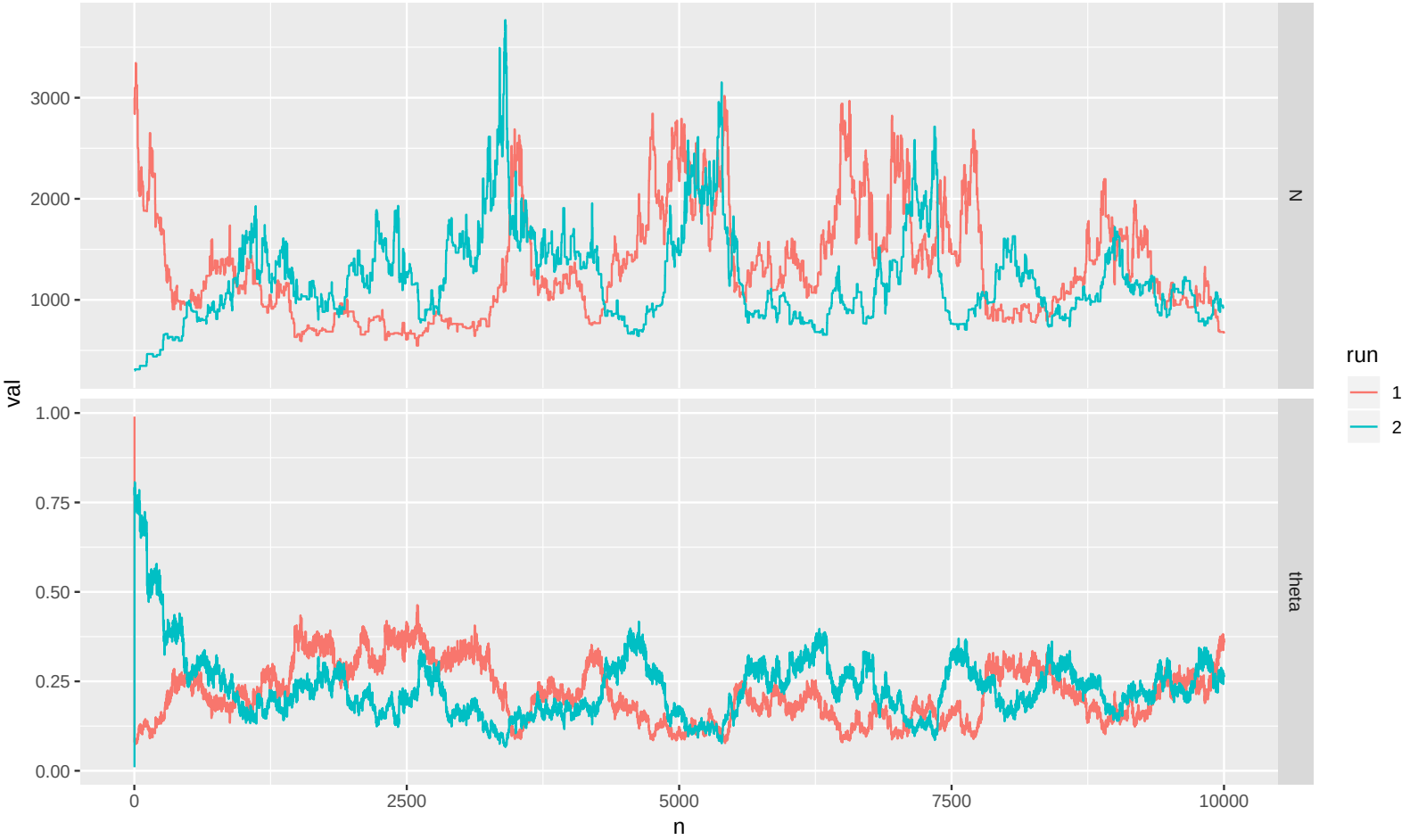
Output analysis

Standard starting point to evaluate convergence:

- Look at the trace plot for each variable
- consider different scalar function of x
- may run different Markov chain with different (extreme) starting values

Example: Korsbetningen

Single site update, two chains with different starting values



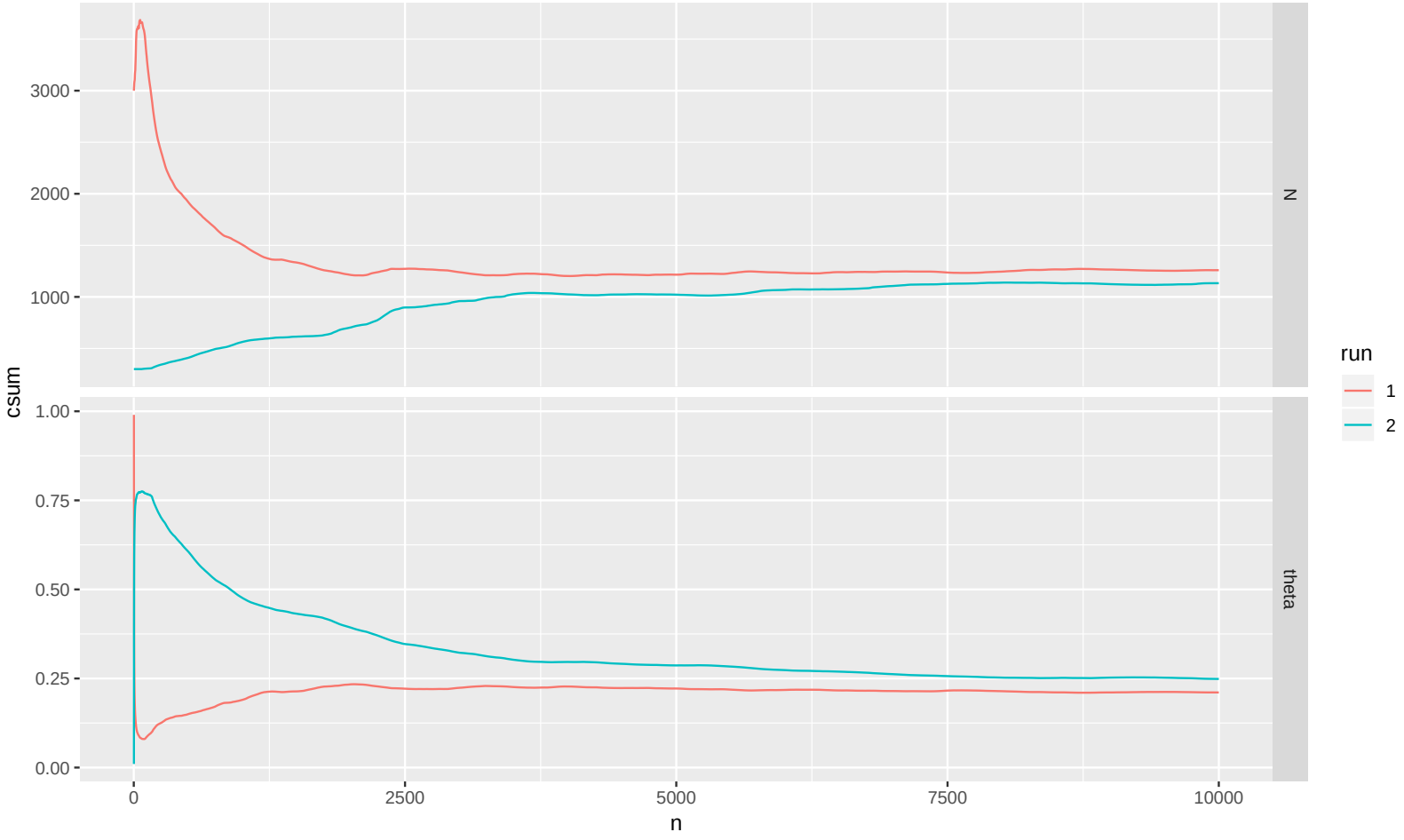
Example: Korsbetningen

Block update, two chains with different starting values

Example: Korsbetningen

Single site update, two chains with different starting values.

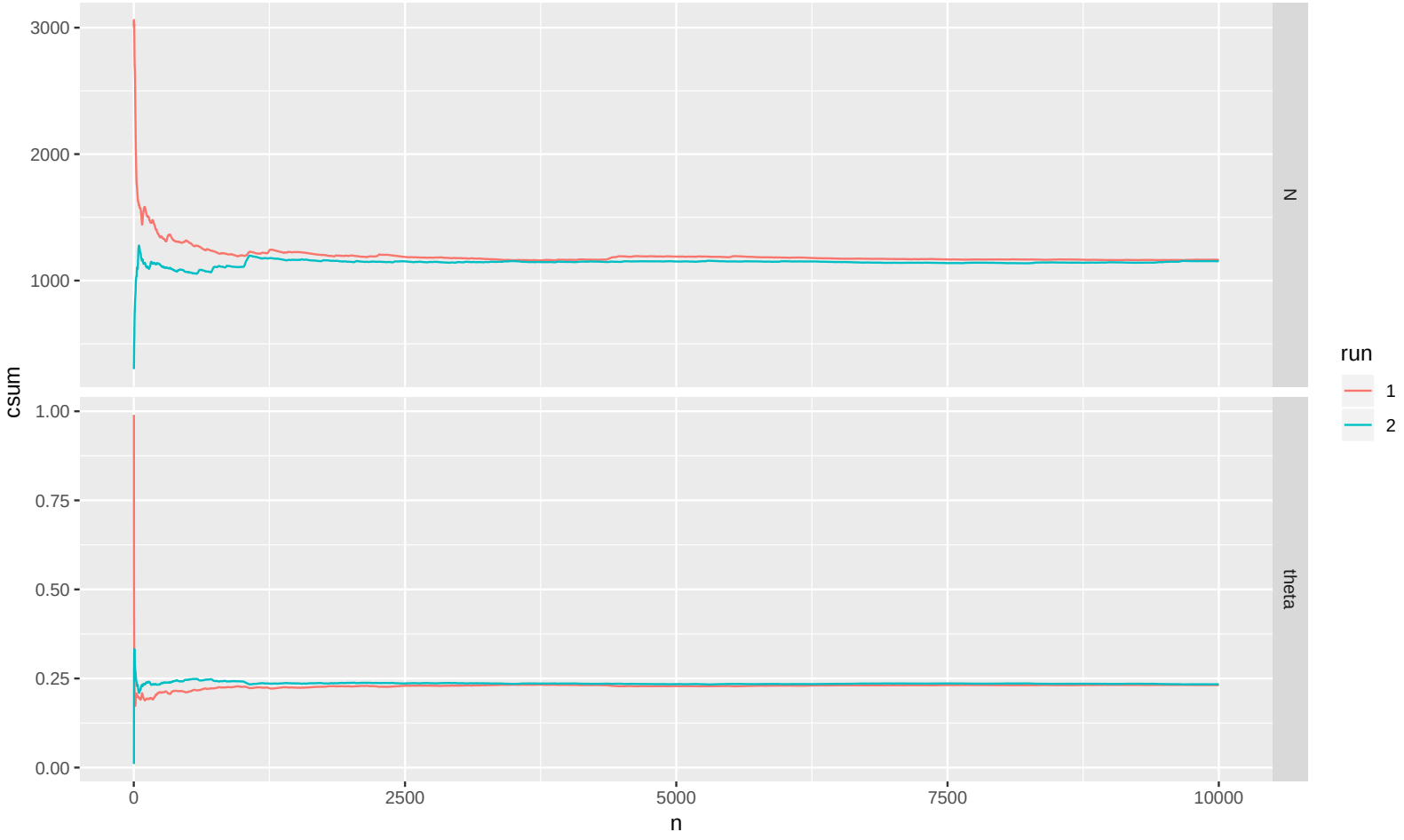
Estiamte of the mean



Example: Korsbetningen

Block update, two chains with different starting values.

Estiamte of the mean



Convergence Diagnostic

With a fixed cpu-time shoud we:

- use all time in one long Markov chain, or
- run several shorter Markov chains?
- One long chain:
  - only one bunr-in period to discard
  - more likely that you really have converged
- Several shorter runs:
  - easier to evaluate convergence
  - easier to estimate the variance of the estimator (the chains are independent)

In practice one often use a combination of the two strategies

Variance of the MCMC estimator

Recall: We want to estimate $\mu = \int g(x)\pi(x)\, dx$ with $\hat{\mu} = \frac{1}{n}\sum g(x_i)$ where $x_i \sim \pi(x)$.

In standard MC we have

$$x_1, x_2, \ldots, x_n \sim \pi(x), \text{ i.i.d.}$$

This gives

$$\mathrm{E}(\hat{\mu}) = \mu \text{ and } \mathrm{Var}(\hat{\mu}) = \frac{\mathrm{Var}(g(X))}{n}$$

We can estimate the variance $\mathrm{Var}(\hat{\mu})$ as

$$\widehat{\mathrm{Var}(\hat{\mu})} = \frac{\widehat{\mathrm{Var}(g(X))}}{n}$$

$$\widehat{\mathrm{Var}(g(X))} = \frac{1}{n-1}\sum(g(x_i) = \hat{\mu})$$

MCMC gives dependence samples, what is the variance then??

Autocorrelation

To examine dependencies of successive MCMC samples, the autocorrelation function can be used. Let $x_1, \ldots, x_N$, where N denotes the number of samples, denote our MCMC chain.

The lag k autocorrelation $\rho(k)$ is the correlation between every draw and its k-th lag. For N reasonably large

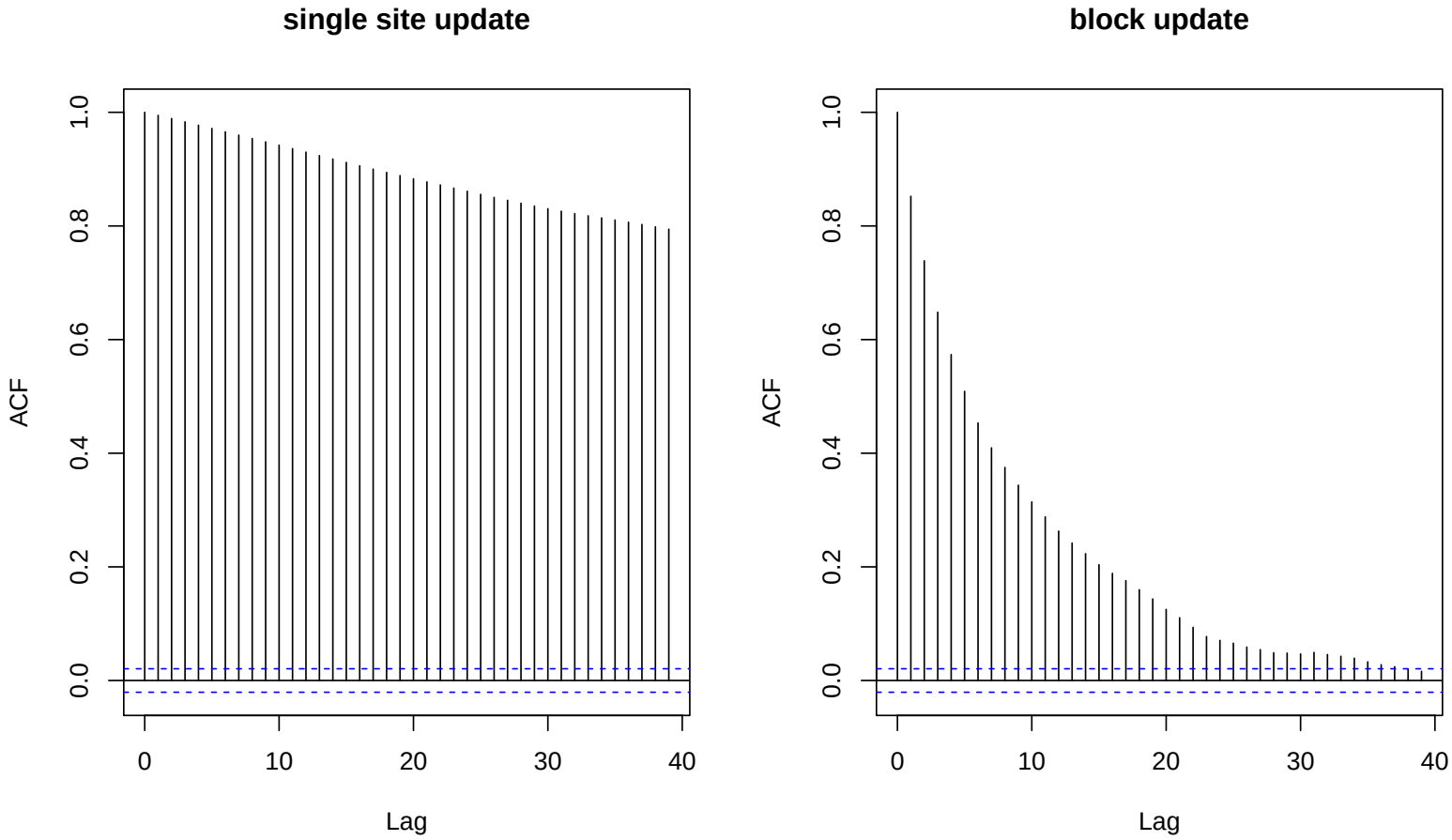
$$\rho(k) \approx \frac{\sum_{i=1}^{N-k}(x_i - \bar{x})(x_{i+k} - \bar{x})}{\sum_{i=1}^{N}(x_i - \bar{x})^2},$$

where $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ is the overall mean.

- With increasing lag k we expect lower autocorrelations.
- If autocorrelation is still relatively high for higher values of k, this indicates high degree of correlation between our draws and slow mixing.

Example: Korsbetningen

Autocorrelation function for N (after discarding the burn-in period)



Effective sample size

A useful measure to compare the performance of different MCMC samplers is the effective sample size (ESS) Kass et al. (1998) American Statistician 52, 93–100..

- The ESS is the estimated number of independent samples needed to obtain a parameter estimate with the same precision as the MCMC estimate based on N dependent samples.

$$\text{ESS} = \frac{N}{\tau}, \quad \tau = 1 + 2 \cdot \sum_{k=1}^{\infty} \rho(k),$$

where τ is the autocorrelation time and $\rho(k)$ the autocorrelation at lag k.

Estimate of ESS

$$\text{ESS} = \frac{N}{\tau}, \quad \tau = 1 + 2 \cdot \sum_{k=1}^{\infty} \rho(k),$$

Estimate τ as

$$\tau = 1 + 2 \cdot \sum_{k=1}^{m} \hat{\rho}(k)$$

where $\hat{\rho}(k)$ is the sample autocorrelation function at lag k, and m is choosen to fullfill some criteria.

Different criteria exists.

Example: Korsbetningen - Effective sample size (ESS)

```
> library(coda)
> nsamples

[1] 8000

> ## single site
> effectiveSize(as.mcmc(res1))

       N    theta
26.39381 23.24576

> ## block update
> effectiveSize(as.mcmc(res2))

       N    theta
624.4336 872.2591

>
```

The precision of the MCMC estimate of the posterior mean of N based on 8000 samples from a single site update is a good as taking 16 independent samples!

Geweke diagnostics

The MCMC chain is divided into two windows

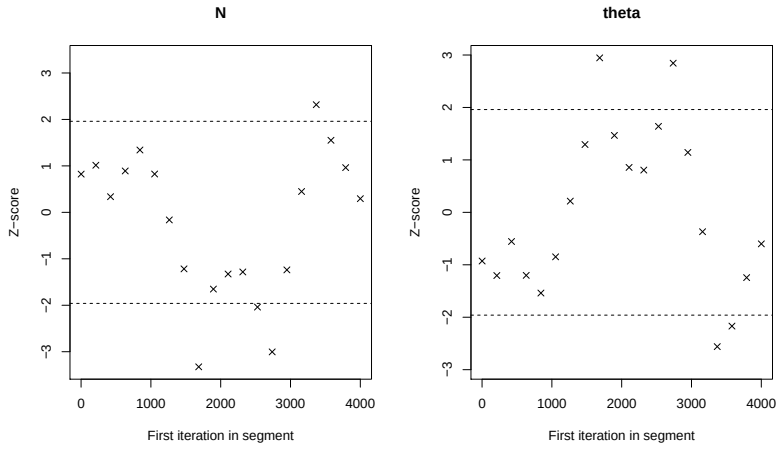
- the first x%, and
- the last y% of the iterates

(`coda` default: $x = 10$, $y = 50$). For both windows the mean is calculated.

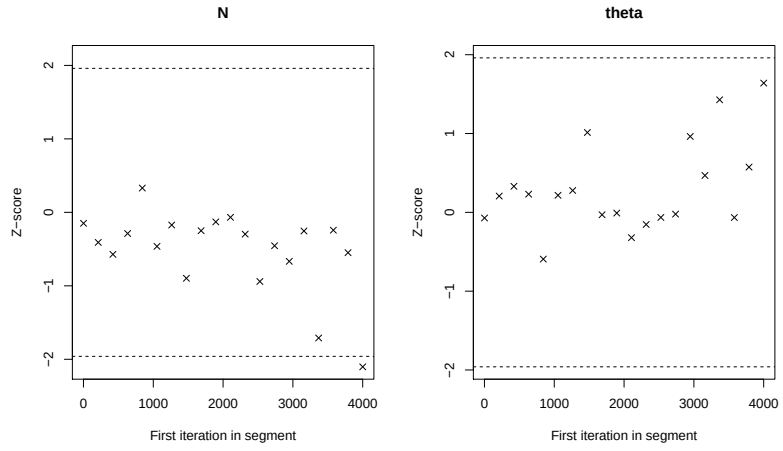
If the chain is stationary both values should be equal and Geweke's test statistic (z-score) follows an asymptotical standard normal distribution.

Example: Korsbetningen - Geweke plot

Single Site



Block Update



Further reading

There are several convergence diagnostics:

- some are based on a single Markov chain run
- some are based on several Markov chain runs

There are no guarantees!

For further reading see for example

- Gilks, W. R., Richardson, S. and Spiegelhalter, D.J. (1996) *Markov Chain Monte Carlo in Practice, Chapman & Hall, London*,

Different approaches are implemented in the

- R-package `coda`. (Plummer et al., 2006)

Summary

- Diagnostics cannot guarantee that chain has converged
- Can indicate that it has not converged

Solutions?

- Run longer and thin output
- Reparametrize model
- "Block" correlated variables together
  - Joint update might be more efficient however for some parameter combination the acceptance rate can be very slow!
- integrate out variables
- ...