UPPSALA UNIVERSITET

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

Bayesian Statistics and Data Analysis
Lecture 5

Måns Magnusson
Department of Statistics, Uppsala University
Thanks to Aki Vehtari, Aalto University

Seminar today at 13.15

- Topic: posteriordb: Testing, Benchmarking and Developing Bayesian Inference Algorithms
- Where: H317 Ekonomikum
- Speaker Jakob Torgander, Department of Statistics, Uppsala University
- Opponent David Broman, EECS, KTH Royal Institute of Technology

Assignment 4

- "More explicit examples of proper weight normalization would help prevent implementation errors in importance sampling."

Assignment 4

- "More explicit examples of proper weight normalization would help prevent implementation errors in importance sampling."
- "$\hat{R}$ diagnostics is mentioned in the information on the assignment"

Assignment 4

- "More explicit examples of proper weight normalization would help prevent implementation errors in importance sampling."
- "$\hat{R}$ diagnostics is mentioned in the information on the assignment"
- "It was hard to find information on how to solve the exercises from the book and lectures."

Assignment 4

- "More explicit examples of proper weight normalization would help prevent implementation errors in importance sampling."
- "$\hat{R}$ diagnostics is mentioned in the information on the assignment"
- "It was hard to find information on how to solve the exercises from the book and lectures."
- "..., maybe add a testable MCSE function so that we can know we have done it correct."

Assignment 4

- "More explicit examples of proper weight normalization would help prevent implementation errors in importance sampling."
- "$\hat{R}$ diagnostics is mentioned in the information on the assignment"
- "It was hard to find information on how to solve the exercises from the book and lectures."
- "..., maybe add a testable MCSE function so that we can know we have done it correct."
- "it might help if the workload were reduced or if there were more time between assignments"

Assignment 4

- "More explicit examples of proper weight normalization would help prevent implementation errors in importance sampling."
- "$\hat{R}$ diagnostics is mentioned in the information on the assignment"
- "It was hard to find information on how to solve the exercises from the book and lectures."
- "..., maybe add a testable MCSE function so that we can know we have done it correct."
- "it might help if the workload were reduced or if there were more time between assignments"
- Minor suggestions and improvements

UPPSALA UNIVERSITET

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

It's all about expectations

$$E_{p(\theta|y)}[f(\theta)] = \int f(\theta)p(\theta|y)d\theta,$$

$$\text{where} \quad p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta)p(\theta)d\theta}$$

It's all about expectations

$$E_{p(\theta|y)}[f(\theta)] = \int f(\theta)p(\theta|y)d\theta,$$

$$\text{where} \quad p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta)p(\theta)d\theta}$$

We can easily evaluate $p(y|\theta)p(\theta)$ for any θ, but the integral $\int p(y|\theta)p(\theta)d\theta$ is usually difficult.

It's all about expectations

$$E_{p(\theta|y)}[f(\theta)] = \int f(\theta) p(\theta|y) d\theta,$$

$$\text{where} \quad p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta)p(\theta)d\theta}$$

We can easily evaluate $p(y|\theta)p(\theta)$ for any θ, but the integral $\int p(y|\theta)p(\theta)d\theta$ is usually difficult.

We can use the unnormalized posterior $q(\theta|y) = p(y|\theta)p(\theta)$, for example, in

It's all about expectations

$$E_{p(\theta|y)}[f(\theta)] = \int f(\theta) p(\theta|y) d\theta,$$

$$\text{where} \quad p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta)p(\theta)d\theta}$$

We can easily evaluate $p(y|\theta)p(\theta)$ for any θ, but the integral $\int p(y|\theta)p(\theta)d\theta$ is usually difficult.

We can use the unnormalized posterior $q(\theta|y) = p(y|\theta)p(\theta)$, for example, in

- Monte Carlo methods which can sample from $p(\theta^{(s)}|y)$ using only $q(\theta^{(s)}|y)$

$$E_{p(\theta|y)}[f(\theta)] \approx \frac{1}{S}\sum_{s=1}^{S} f(\theta^{(s)})$$

Monte Carlo

- Monte Carlo methods we have discussed so far
  - Inverse CDF works mainly for 1D

Monte Carlo

- Monte Carlo methods we have discussed so far
  - Inverse CDF works mainly for 1D
  - Grid methods works in a few dimensions

UPPSALA UNIVERSITET

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

Monte Carlo

- Monte Carlo methods we have discussed so far
  - Inverse CDF works mainly for 1D
  - Grid methods works in a few dimensions
  - Rejection sampling works mostly in 1D (truncation is a special case)

Monte Carlo

- Monte Carlo methods we have discussed so far
  - Inverse CDF works mainly for 1D
  - Grid methods works in a few dimensions
  - Rejection sampling works mostly in 1D (truncation is a special case)
  - Importance sampling is reliable only in special cases

UPPSALA UNIVERSITET

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

Monte Carlo

- Monte Carlo methods we have discussed so far
  - Inverse CDF works mainly for 1D
  - Grid methods works in a few dimensions
  - Rejection sampling works mostly in 1D (truncation is a special case)
  - Importance sampling is reliable only in special cases
- What to do in high dimensions?

UPPSALA UNIVERSITET

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

Monte Carlo

- Monte Carlo methods we have discussed so far
  - Inverse CDF works mainly for 1D
  - Grid methods works in a few dimensions
  - Rejection sampling works mostly in 1D (truncation is a special case)
  - Importance sampling is reliable only in special cases
- What to do in high dimensions?
  - Markov chain Monte Carlo (Ch 11-12)

Monte Carlo

- Monte Carlo methods we have discussed so far
  - Inverse CDF works mainly for 1D
  - Grid methods works in a few dimensions
  - Rejection sampling works mostly in 1D (truncation is a special case)
  - Importance sampling is reliable only in special cases
- What to do in high dimensions?
  - Markov chain Monte Carlo (Ch 11-12)
  - Laplace, Variational*, EP* (Ch 4, 13*, next course)

Markov chains

- Andrey Markov proved weak law of large numbers and central limit theorem for certain dependent-random sequences, which were later named Markov chains

Markov chains

- Andrey Markov proved weak law of large numbers and central limit theorem for certain dependent-random sequences, which were later named Markov chains
- The probability of each event depends only on the previous event (or finite number of previous events)

$$p(\theta_t|\theta_{t-1}, \theta_{t-2}, ...) = p(\theta_t|\theta_{t-1})$$

Markov chains

- Andrey Markov proved weak law of large numbers and central limit theorem for certain dependent-random sequences, which were later named Markov chains
- The probability of each event depends only on the previous event (or finite number of previous events)

$$p(\theta_t|\theta_{t-1}, \theta_{t-2}, \dots) = p(\theta_t|\theta_{t-1})$$

- $T_t(\theta_t \mid \theta_{t-1}) \equiv p(\theta_t|\theta_{t-1})$ is usually refered to as the transition distribution

Markov chains

- Andrey Markov proved weak law of large numbers and central limit theorem for certain dependent-random sequences, which were later named Markov chains
- The probability of each event depends only on the previous event (or finite number of previous events)

$$p(\theta_t|\theta_{t-1}, \theta_{t-2}, ...) = p(\theta_t|\theta_{t-1})$$

- $T_t(\theta_t \mid \theta_{t-1}) \equiv p(\theta_t|\theta_{t-1})$ is usually refered to as the transition distribution
- Under some assumptions $p(\theta_t|\theta_{t-1})$ will converge (in total variation) to *one* stationary distribution $p(\theta)$

Markov chains

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

- Andrey Markov proved weak law of large numbers and central limit theorem for certain dependent-random sequences, which were later named Markov chains
- The probability of each event depends only on the previous event (or finite number of previous events)

$$p(\theta_t|\theta_{t-1}, \theta_{t-2}, ...) = p(\theta_t|\theta_{t-1})$$

- $T_t(\theta_t \mid \theta_{t-1}) \equiv p(\theta_t|\theta_{t-1})$ is usually refered to as the transition distribution
- Under some assumptions $p(\theta_t|\theta_{t-1})$ will converge (in total variation) to *one* stationary distribution $p(\theta)$
- Goal in MCMC: Construct a transition distribution with $p(\theta|y)$ as the stationary distribution

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

Markov chain Monte Carlo (MCMC)

- Produce draws $\theta_{(t)}$ given $\theta_{(t-1)}$ from a Markov chain, with stationary distribution $p(\theta|y)$

Markov chain Monte Carlo (MCMC)

- Produce draws $\theta_{(t)}$ given $\theta_{(t-1)}$ from a Markov chain, with stationary distribution $p(\theta|y)$
  - \+ generic
  - \+ combine sequence of easier Monte Carlo draws to form a Markov chain

Markov chain Monte Carlo (MCMC)

- Produce draws $\theta_{(t)}$ given $\theta_{(t-1)}$ from a Markov chain, with stationary distribution $p(\theta|y)$
  - \+ generic
  - \+ combine sequence of easier Monte Carlo draws to form a Markov chain
  - \+ converges (in total variation) to $p(\theta|y)$ as $t \to \infty$

Markov chain Monte Carlo (MCMC)

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

- Produce draws $\theta_{(t)}$ given $\theta_{(t-1)}$ from a Markov chain, with stationary distribution $p(\theta|y)$
  - $+$ generic
  - $+$ combine sequence of easier Monte Carlo draws to form a Markov chain
  - $+$ converges (in total variation) to $p(\theta|y)$ as $t \to \infty$
  - $+$ central limit theorem holds for expectations

Markov chain Monte Carlo (MCMC)

- Produce draws $\theta_{(t)}$ given $\theta_{(t-1)}$ from a Markov chain, with stationary distribution $p(\theta|y)$
  - + generic
  - + combine sequence of easier Monte Carlo draws to form a Markov chain
  - + converges (in total variation) to $p(\theta|y)$ as $t \to \infty$
  - + central limit theorem holds for expectations
  - - draws are dependent
  - - construction of an efficient Markov chains is not always easy

MCMC summary

- Random variables $\theta_1, \theta_2, \ldots$ where θ_t depends only on the previous $\theta_{(t-1)}$

$$p(\theta_t|\theta_1, \ldots, \theta_{(t-1)}) = p(\theta_t|\theta_{(t-1)})$$

MCMC summary

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

- Random variables $\theta_1, \theta_2, \ldots$ where θ_t depends only on the previous $\theta_{(t-1)}$

$$p(\theta_t | \theta_1, \ldots, \theta_{(t-1)}) = p(\theta_t | \theta_{(t-1)})$$

- Chain has to be initialized with some starting point θ_0

MCMC summary

- Random variables $\theta_1, \theta_2, \ldots$ where θ_t depends only on the previous $\theta_{(t-1)}$

$$p(\theta_t | \theta_1, \ldots, \theta_{(t-1)}) = p(\theta_t | \theta_{(t-1)})$$

- Chain has to be initialized with some starting point θ_0
- Choose a transition distribution so the stationary distribution of the Markov chain is $p(\theta | y)$

Subsection 1

Gibbs sampling

Gibbs sampling

- Alternate sampling from conditional distributions
- Basic algorithm, for $j \in \{1, ..., J\}$

$$\text{sample } \theta_{j,t} \text{ from } \quad p(\theta_j | \theta_{-j,t-1}, y),$$
$$\text{where } \quad \theta_{-j,t-1} = (\theta_{1,t}, \ldots, \theta_{j-1,t}, \theta_{j+1,t-1}, \ldots, \theta_{J,t-1})$$

- Will converge (in total variation) to $p(\theta|y)$ as $N \to \infty$

Gibbs sampling

- Alternate sampling from conditional distributions
- Basic algorithm, for $j \in \{1, ..., J\}$

$$\text{sample } \theta_{j,t} \text{ from } \quad p(\theta_j | \theta_{-j,t-1}, y),$$
$$\text{where} \quad \theta_{-j,t-1} = (\theta_{1,t}, \ldots, \theta_{j-1,t}, \theta_{j+1,t-1}, \ldots, \theta_{J,t-1})$$

- Will converge (in total variation) to $p(\theta|y)$ as $N \to \infty$
- j can be multiple (blocked) parameters
- 1D sampling ($|j| = 1$) is generally easy

Gibbs sampling

- Alternate sampling from conditional distributions
- Basic algorithm, for $j \in \{1, ..., J\}$

$$\text{sample } \theta_{j,t} \text{ from } \quad p(\theta_j | \theta_{-j,t-1}, y),$$
$$\text{where} \quad \theta_{-j,t-1} = (\theta_{1,t}, \ldots, \theta_{j-1,t}, \theta_{j+1,t-1}, \ldots, \theta_{J,t-1})$$

- Will converge (in total variation) to $p(\theta|y)$ as $N \to \infty$
- j can be multiple (blocked) parameters
- 1D sampling ($|j| = 1$) is generally easy
- Popular for discrete parameters

Example: Bivariate Normal using Gibbs

$$p(\theta \mid y) \sim \mathcal{N}\left(\bar{y}, \tfrac{1}{n}\Sigma\right).$$

where Σ is known and $\theta \in \mathcal{R}^2$

Example: Bivariate Normal using Gibbs

UPPSALA UNIVERSITET

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

$$p(\theta \mid y) \sim \mathcal{N}\left(\bar{y}, \tfrac{1}{n}\Sigma\right).$$

where Σ is known and $\theta \in \mathcal{R}^2$

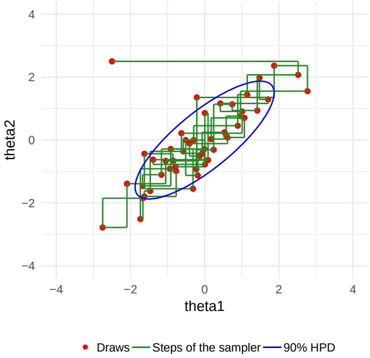
Using Gibbs sampling:

Given starting values $(\theta_1^{(0)}, \theta_2^{(0)})$, for $t = 1, 2, \ldots$:

$$\theta_1^{(t)} \sim \mathcal{N}\left(\bar{y}_1 + \frac{\rho}{\sigma_2}(\theta_2^{(t-1)} - \bar{y}_2),\ \frac{1}{n}\left(\sigma_1 - \frac{\rho^2}{\sigma_2}\right)\right),$$

$$\theta_2^{(t)} \sim \mathcal{N}\left(\bar{y}_2 + \frac{\rho}{\sigma_1}(\theta_1^{(t)} - \bar{y}_1),\ \frac{1}{n}\left(\sigma_2 - \frac{\rho^2}{\sigma_1}\right)\right).$$

Gibbs sampling



demo

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

Gibbs sampling

- With conditionally conjugate priors, the sampling from the conditional distributions is easy for wide range of models

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

Gibbs sampling

- With conditionally conjugate priors, the sampling from the conditional distributions is easy for wide range of models
- BUGS / WinBUGS / OpenBUGS / JAGS

Gibbs sampling

- With conditionally conjugate priors, the sampling from the conditional distributions is easy for wide range of models
- BUGS / WinBUGS / OpenBUGS / JAGS
- No algorithm parameters to tune

Gibbs sampling

- With conditionally conjugate priors, the sampling from the conditional distributions is easy for wide range of models
- BUGS / WinBUGS / OpenBUGS / JAGS
- No algorithm parameters to tune
- For not so easy conditionals, use e.g. inverse-CDF

Gibbs sampling

- With conditionally conjugate priors, the sampling from the conditional distributions is easy for wide range of models
- BUGS / WinBUGS / OpenBUGS / JAGS
- No algorithm parameters to tune
- For not so easy conditionals, use e.g. inverse-CDF
- Several parameters can be updated in blocks (*blocking*)

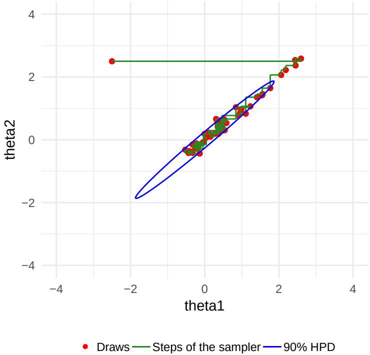
Gibbs sampling

- With conditionally conjugate priors, the sampling from the conditional distributions is easy for wide range of models
- BUGS / WinBUGS / OpenBUGS / JAGS
- No algorithm parameters to tune
- For not so easy conditionals, use e.g. inverse-CDF
- Several parameters can be updated in blocks (*blocking*)
- Slow if parameters are highly dependent in the posterior...

UPPSALA UNIVERSITET

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

Gibbs sampling



demo

Sampling conditional vs joint

- How about sampling θ jointly?
  - e.g. it is easy to sample from multivariate normal

Sampling conditional vs joint

UPPSALA UNIVERSITET

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

- How about sampling θ jointly?
  - e.g. it is easy to sample from multivariate normal
- Can we use that to form a Markov chain?

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

Subsection 2

Metropolis-Hastings

The Metropolis algorithm

- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \dots$
     (a) pick a proposal θ^* from a proposal distribution $J_t(\theta^*|\theta_{t-1})$.
     (Proposal distribution has to be symmetric, i.e. $J_t(\theta_a|\theta_b) = J_t(\theta_b|\theta_a)$, for all θ_a, θ_b)

The Metropolis algorithm

- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \ldots$
     (a) pick a proposal θ^* from a proposal distribution $J_t(\theta^*|\theta_{t-1})$.
     (Proposal distribution has to be symmetric, i.e. $J_t(\theta_a|\theta_b) = J_t(\theta_b|\theta_a)$, for all θ_a, θ_b)
     (b) calculate acceptance ratio

$$r = \frac{p(\theta^*|y)}{p(\theta_{t-1}|y)}$$

The Metropolis algorithm

- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \ldots$
     (a) pick a proposal θ^* from a proposal distribution $J_t(\theta^*|\theta_{t-1})$.
     (Proposal distribution has to be symmetric, i.e. $J_t(\theta_a|\theta_b) = J_t(\theta_b|\theta_a)$, for all θ_a, θ_b)
     (b) calculate acceptance ratio

$$r = \frac{p(\theta^*|y)}{p(\theta_{t-1}|y)}$$

     (c) set

$$\theta_t = \begin{cases} \theta^* & \text{with probability } \min(r, 1) \\ \theta_{t-1} & \text{otherwise} \end{cases}$$

The Metropolis algorithm

- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \ldots$
     (a) pick a proposal θ^* from a proposal distribution $J_t(\theta^* | \theta_{t-1})$.
     (Proposal distribution has to be symmetric, i.e. $J_t(\theta_a | \theta_b) = J_t(\theta_b | \theta_a)$, for all θ_a, θ_b)
     (b) calculate acceptance ratio
     $$r = \frac{p(\theta^* | y)}{p(\theta_{t-1} | y)}$$
     (c) set
     $$\theta_t = \begin{cases} \theta^* & \text{with probability } \min(r, 1) \\ \theta_{t-1} & \text{otherwise} \end{cases}$$
     ie, if $p(\theta^* | y) > p(\theta_{t-1} | y)$ accept the proposal always and otherwise accept the proposal with probability r

The Metropolis algorithm

- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \dots$
     (a) pick a proposal θ^* from a proposal distribution $J_t(\theta^* | \theta_{t-1})$.
     (Proposal distribution has to be symmetric, i.e. $J_t(\theta_a|\theta_b) = J_t(\theta_b|\theta_a)$, for all θ_a, θ_b)
     (b) calculate acceptance ratio
     $$r = \frac{p(\theta^*|y)}{p(\theta_{t-1}|y)}$$
     (c) set
     $$\theta_t = \begin{cases} \theta^* & \text{with probability } \min(r, 1) \\ \theta_{t-1} & \text{otherwise} \end{cases}$$

  - rejection of a proposal increments the time t also by one ie, the new state is the same as previous

The Metropolis algorithm

- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \ldots$
     (a) pick a proposal θ^* from a proposal distribution $J_t(\theta^* | \theta_{t-1})$.
     (Proposal distribution has to be symmetric, i.e. $J_t(\theta_a | \theta_b) = J_t(\theta_b | \theta_a)$, for all θ_a, θ_b)
     (b) calculate acceptance ratio
     $$r = \frac{p(\theta^* | y)}{p(\theta_{t-1} | y)}$$
     (c) set
     $$\theta_t = \begin{cases} \theta^* & \text{with probability } \min(r, 1) \\ \theta_{t-1} & \text{otherwise} \end{cases}$$

  - rejection of a proposal increments the time t also by one ie, the new state is the same as previous
  - step c is executed by generating a random number from $\mathcal{U}(0, 1)$

The Metropolis algorithm

- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \ldots$
     (a) pick a proposal θ^* from a proposal distribution $J_t(\theta^*|\theta_{t-1})$.
     (Proposal distribution has to be symmetric, i.e. $J_t(\theta_a|\theta_b) = J_t(\theta_b|\theta_a)$, for all θ_a, θ_b)
     (b) calculate acceptance ratio
     $$r = \frac{p(\theta^*|y)}{p(\theta_{t-1}|y)}$$
     (c) set
     $$\theta_t = \begin{cases} \theta^* & \text{with probability } \min(r, 1) \\ \theta_{t-1} & \text{otherwise} \end{cases}$$

  - rejection of a proposal increments the time t also by one ie, the new state is the same as previous
  - step c is executed by generating a random number from $\mathcal{U}(0, 1)$
  - $p(\theta^*|y)$ and $p(\theta_{t-1}|y)$ have the same normalization terms, and thus instead of $p(\cdot|y)$, unnormalized $q(\cdot|y)$ can be used, as the normalization terms cancel out!

Metropolis algorithm

- Example: one bivariate observation (y_1, y_2)
  - bivariate normal distribution with unknown mean and known covariance

$$\begin{pmatrix}\theta_1\\ \theta_2\end{pmatrix}\Bigg| y \sim \mathcal{N}\left(\begin{pmatrix}y_1\\ y_2\end{pmatrix}, \begin{pmatrix}1 & \rho\\ \rho & 1\end{pmatrix}\right)$$

  - proposal distribution $J_t(\theta^*|\theta_{t-1}) = \mathcal{N}(\theta^*|\theta_{t-1}, \sigma_p^2)$

demo

Why Metropolis algorithm works

- Intuitively more draws from the higher density areas as jumps to higher density are always accepted and only some of the jumps to the lower density are accepted

UPPSALA UNIVERSITET

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

Why Metropolis algorithm works

- Intuitively more draws from the higher density areas as jumps to higher density are always accepted and only some of the jumps to the lower density are accepted

- Theoretically
  1. Prove that simulated series is a Markov chain which has unique stationary distribution
  2. Prove that this stationary distribution is the desired target distribution $p(\theta|y)$

Why Metropolis algorithm works

1. Prove that simulated series is a Markov chain which has unique stationary distribution
   a) irreducible

   b) aperiodic

   c) recurrent / not transient

UPPSALA UNIVERSITET

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

Why Metropolis algorithm works

1. Prove that simulated series is a Markov chain which has unique stationary distribution
   a) irreducible
      = positive probability of eventually reaching any state from any other state
   b) aperiodic

   c) recurrent / not transient

Why Metropolis algorithm works

1. Prove that simulated series is a Markov chain which has unique stationary distribution
   a) irreducible
      = positive probability of eventually reaching any state from any other state
   b) aperiodic
      = aperiodic (return times are not periodic)
      - holds for a random walk on any proper distribution (except for trivial exceptions)
   c) recurrent / not transient

Why Metropolis algorithm works

1. Prove that simulated series is a Markov chain which has unique stationary distribution
   a) irreducible
      = positive probability of eventually reaching any state from any other state
   b) aperiodic
      = aperiodic (return times are not periodic)
      - holds for a random walk on any proper distribution (except for trivial exceptions)
   c) recurrent / not transient
      = probability to return to a state i is 1 as $T \to \infty$
      - holds for a random walk on any proper distribution (except for trivial exceptions)

Why Metropolis algorithm works

2. Prove that this stationary distribution is the desired target distribution $p(\theta|y)$: see book
   - Show *detailed balance* with respect to $p(\theta|y)$:

     $$p(\theta|y)\, T(\theta|\theta') = p(\theta'|y)\, T(\theta'|\theta), \quad \forall \theta, \theta'.$$

     Here, $T(\theta'|\theta)$ denotes the one-step transition probability (or density).

Why Metropolis algorithm works

2. Prove that this stationary distribution is the desired target distribution $p(\theta|y)$: see book
   - Show *detailed balance* with respect to $p(\theta|y)$:

     $$p(\theta|y)\,T(\theta|\theta') = p(\theta'|y)\,T(\theta'|\theta), \quad \forall \theta, \theta'.$$

     Here, $T(\theta'|\theta)$ denotes the one-step transition probability (or density).

   - Intuition:
     - Think of $p(\theta|y)$ as the "mass" of probability at state θ.

Why Metropolis algorithm works

2. Prove that this stationary distribution is the desired target distribution $p(\theta|y)$: see book
   - Show *detailed balance* with respect to $p(\theta|y)$:

     $$p(\theta|y)\, T(\theta|\theta') = p(\theta'|y)\, T(\theta'|\theta), \quad \forall \theta, \theta'.$$

     Here, $T(\theta'|\theta)$ denotes the one-step transition probability (or density).
   - Intuition:
     - Think of $p(\theta|y)$ as the "mass" of probability at state θ.
     - $p(\theta|y)P(\theta, \theta')$ is the "flow" of mass from θ to θ'.

Why Metropolis algorithm works

2. Prove that this stationary distribution is the desired target distribution $p(\theta|y)$: see book
   - Show *detailed balance* with respect to $p(\theta|y)$:

     $$p(\theta|y)\, T(\theta|\theta') = p(\theta'|y)\, T(\theta'|\theta), \quad \forall \theta, \theta'.$$

     Here, $T(\theta'|\theta)$ denotes the one-step transition probability (or density).

   - Intuition:
     - Think of $p(\theta|y)$ as the "mass" of probability at state θ.
     - $p(\theta|y)P(\theta, \theta')$ is the "flow" of mass from θ to θ'.
     - Detailed balance: the flow from θ to θ' equals the flow back from θ' to θ — nothing accumulates or leaks.

Why Metropolis algorithm works

2. Prove that this stationary distribution is the desired target distribution $p(\theta|y)$: see book
   - Show *detailed balance* with respect to $p(\theta|y)$:

     $$p(\theta|y)\, T(\theta|\theta') = p(\theta'|y)\, T(\theta'|\theta), \quad \forall \theta, \theta'.$$

     Here, $T(\theta'|\theta)$ denotes the one-step transition probability (or density).

   - Intuition:
     - Think of $p(\theta|y)$ as the "mass" of probability at state θ.
     - $p(\theta|y)P(\theta, \theta')$ is the "flow" of mass from θ to θ'.
     - Detailed balance: the flow from θ to θ' equals the flow back from θ' to θ — nothing accumulates or leaks.

More indepth in a proper Markov Theory course!

Metropolis-Hastings algorithm

- Generalization of Metropolis algorithm for non-symmetric proposal distributions
  - acceptance ratio includes ratio of proposal distributions

$$r = \frac{p(\theta^*|y)/J_t(\theta^*|\theta_{t-1})}{p(\theta_{t-1}|y)/J_t(\theta_{t-1}|\theta^*)}$$

Metropolis-Hastings algorithm

- Generalization of Metropolis algorithm for non-symmetric proposal distributions
  - acceptance ratio includes ratio of proposal distributions

$$r = \frac{p(\theta^*|y)/J_t(\theta^*|\theta_{t-1})}{p(\theta_{t-1}|y)/J_t(\theta_{t-1}|\theta^*)} = \frac{p(\theta^*|y)J_t(\theta_{t-1}|\theta^*)}{p(\theta_{t-1}|y)J_t(\theta^*|\theta_{t-1})}$$

Metropolis-Hastings algorithm

- Ideal proposal distribution is the distribution itself
  - $J(\theta^*|\theta) \equiv p(\theta^*|y)$ for all θ
  - acceptance probability is 1
  - independent draws
  - not usually feasible

Metropolis-Hastings algorithm

- Ideal proposal distribution is the distribution itself
  - $J(\theta^*|\theta) \equiv p(\theta^*|y)$ for all θ
  - acceptance probability is 1
  - independent draws
  - not usually feasible
- Good proposal distribution resembles the target distribution
  - if the shape of the target distribution is unknown, usually normal or t distribution is used

Metropolis-Hastings algorithm

- Ideal proposal distribution is the distribution itself
  - $J(\theta^*|\theta) \equiv p(\theta^*|y)$ for all θ
  - acceptance probability is 1
  - independent draws
  - not usually feasible
- Good proposal distribution resembles the target distribution
  - if the shape of the target distribution is unknown, usually normal or t distribution is used
- After the proposal distribution shape has been selected, it is important to select the scale
  - small scale
    $\rightarrow$ many steps accepted, but the chain moves slowly due to small steps
  - big scale
    $\rightarrow$ long steps proposed, but many of those rejected and again chain moves slowly

demo

Metropolis-Hastings algorithm

- Ideal proposal distribution is the distribution itself
  - $J(\theta^*|\theta) \equiv p(\theta^*|y)$ for all θ
  - acceptance probability is 1
  - independent draws
  - not usually feasible
- Good proposal distribution resembles the target distribution
  - if the shape of the target distribution is unknown, usually normal or t distribution is used
- After the proposal distribution shape has been selected, it is important to select the scale
  - small scale
    $\rightarrow$ many steps accepted, but the chain moves slowly due to small steps
  - big scale
    $\rightarrow$ long steps proposed, but many of those rejected and again chain moves slowly

demo

- Generic rule for rejection rate is 60-90%

UPPSALA UNIVERSITET

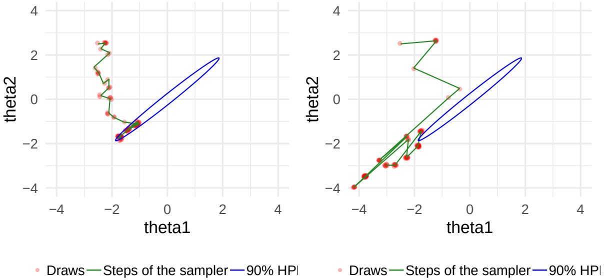
- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

Gibbs sampling as a special case

- Specific case of Metropolis-Hastings algorithm
  - single updated (or blocked)
  - proposal distribution is the conditional distribution
    $\rightarrow$ proposal and target distributions are same
    $\rightarrow$ acceptance probability is 1

Metropolis

- Usually doesn't scale well to high dimensions
  - if the shape doesn't match the whole distribution, the efficiency drops



Section 3

MCMC Diagnostics

What we want to know

1. Has the chains converged to the posterior distribution?

What we want to know

1. Has the chains converged to the posterior distribution?
2. How many efficient samples do I have?

Subsection 1

Warm-up

UPPSALA UNIVERSITET

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

Warm-up

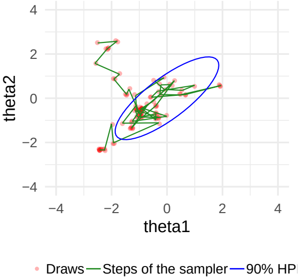
- Asymptotically chain spends the α% of time where α% posterior mass is

UPPSALA UNIVERSITET

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

Warm-up

- Asymptotically chain spends the $\alpha\%$ of time where $\alpha\%$ posterior mass is
  - but in finite time the initial part of the Markov chain may be non-representative

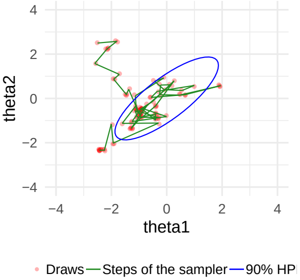


UPPSALA UNIVERSITET

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

Warm-up

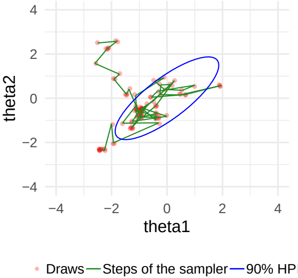
- Asymptotically chain spends the $\alpha\%$ of time where $\alpha\%$ posterior mass is
  - but in finite time the initial part of the Markov chain may be non-representative



- Warm-up period $=$ (non-representative) draws from the beginning of the Markov chain
  - remove warm-up before using samples for inference
  - warm-up may include phase for adapting algorithm parameters

Warm-up

- Asymptotically chain spends the $\alpha\%$ of time where $\alpha\%$ posterior mass is
  - but in finite time the initial part of the Markov chain may be non-representative



- Warm-up period = (non-representative) draws from the beginning of the Markov chain
  - remove warm-up before using samples for inference
  - warm-up may include phase for adapting algorithm parameters
- Also called burn-in

Subsection 2

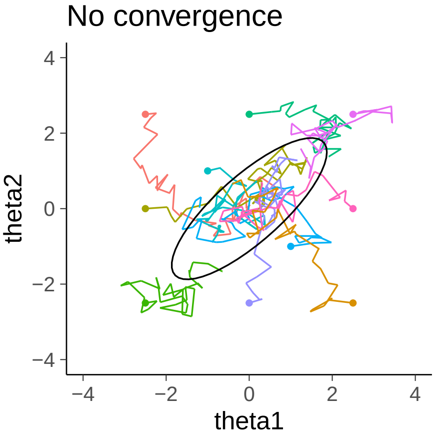
Convergence

Assesing convergence

- Several Markov chains make convergence diagnostics easier

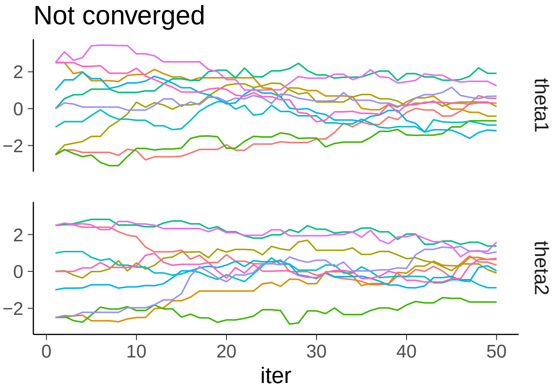
Assesing convergence

- Several Markov chains make convergence diagnostics easier
- Start chains from different starting points – preferably overdispersed



- Remove warm-up draws and run chains long enough

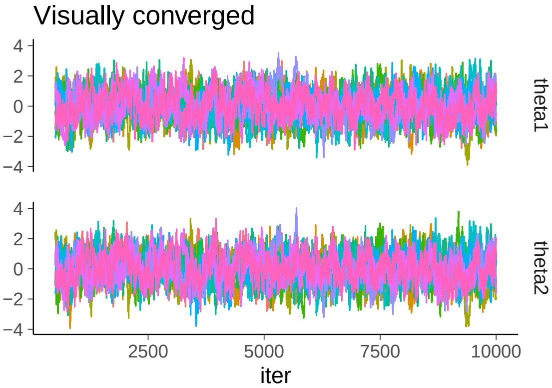
Several chains



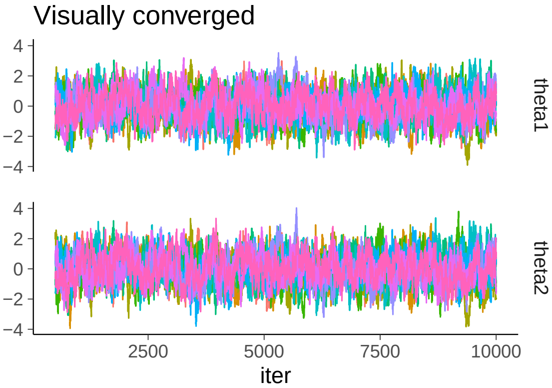
UPPSALA UNIVERSITET

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

Several chains



Several chains



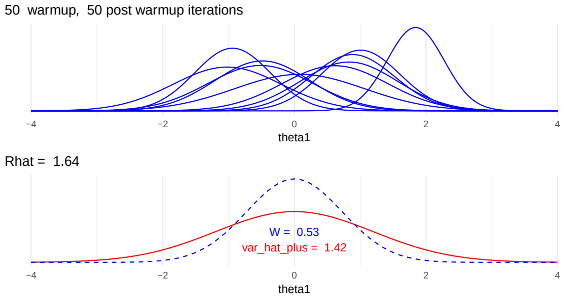
Visual convergence check is not sufficient

$\widehat{R}$: comparison of within and between variances of the chains

- $\widehat{R}$ or potential scale reduction factor (PSRF)
- Compare means and variances of the chains

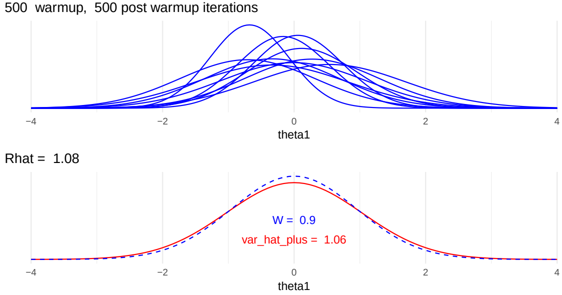
$\widehat{R}$: comparison of within and between variances of the chains

- $\widehat{R}$ or potential scale reduction factor (PSRF)
- Compare means and variances of the chains
  W = within chain variance estimate
  var_hat_plus = total variance estimate



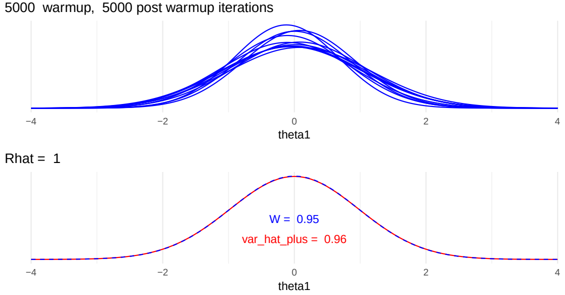
$\widehat{R}$: comparison of within and between variances of the chains

- $\widehat{R}$ or potential scale reduction factor (PSRF)
- Compare means and variances of the chains
  W = within chain variance estimate
  var_hat_plus = total variance estimate



$\widehat{R}$: comparison of within and between variances of the chains

- $\widehat{R}$ or potential scale reduction factor (PSRF)
- Compare means and variances of the chains
  W = within chain variance estimate
  var_hat_plus = total variance estimate



UPPSALA UNIVERSITET

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

$\widehat{R}$

- M chains, each having N draws

$\widehat{R}$

- M chains, each having N draws
- Within chains variance W

$$W = \frac{1}{M}\sum_{m=1}^{M} s_m^2, \text{ where } s_m^2 = \frac{1}{N-1}\sum_{n=1}^{N}(\theta_{nm} - \bar{\theta}_{.m})^2$$

$\widehat{R}$

- M chains, each having N draws
- Within chains variance W

$$W = \frac{1}{M}\sum_{m=1}^{M} s_m^2, \text{ where } s_m^2 = \frac{1}{N-1}\sum_{n=1}^{N}(\theta_{nm} - \bar{\theta}_{.m})^2$$

- Between chains variance B

$$B = \frac{N}{M-1}\sum_{m=1}^{M}(\bar{\theta}_{.m} - \bar{\theta}_{..})^2,$$

$$\text{where } \bar{\theta}_{.m} = \frac{1}{N}\sum_{n=1}^{N}\theta_{nm},\ \bar{\theta}_{..} = \frac{1}{M}\sum_{m=1}^{M}\bar{\theta}_{.m}$$

UPPSALA UNIVERSITET

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

$\widehat{R}$

- M chains, each having N draws
- Within chains variance W

$$W = \frac{1}{M}\sum_{m=1}^{M} s_m^2, \text{ where } s_m^2 = \frac{1}{N-1}\sum_{n=1}^{N}(\theta_{nm} - \bar{\theta}_{.m})^2$$

- Between chains variance B

$$B = \frac{N}{M-1}\sum_{m=1}^{M}(\bar{\theta}_{.m} - \bar{\theta}_{..})^2,$$

$$\text{where } \bar{\theta}_{.m} = \frac{1}{N}\sum_{n=1}^{N}\theta_{nm}, \ \bar{\theta}_{..} = \frac{1}{M}\sum_{m=1}^{M}\bar{\theta}_{.m}$$

  - B/N is variance of the means of the chains

$\widehat{R}$

- M chains, each having N draws
- Within chains variance W

$$W = \frac{1}{M}\sum_{m=1}^{M} s_m^2, \text{ where } s_m^2 = \frac{1}{N-1}\sum_{n=1}^{N}(\theta_{nm} - \bar{\theta}_{.m})^2$$

- Between chains variance B

$$B = \frac{N}{M-1}\sum_{m=1}^{M}(\bar{\theta}_{.m} - \bar{\theta}_{..})^2,$$

$$\text{where } \bar{\theta}_{.m} = \frac{1}{N}\sum_{n=1}^{N}\theta_{nm},\ \bar{\theta}_{..} = \frac{1}{M}\sum_{m=1}^{M}\bar{\theta}_{.m}$$

  - B/N is variance of the means of the chains

- Estimate total variance $\mathrm{var}(\theta|y)$ as a weighted mean of W and B

$$\widehat{\mathrm{var}}^+(\theta|y) = \frac{N-1}{N}W + \frac{1}{N}B$$

$\widehat{R}$

- Estimate total variance $\text{var}(\theta|y)$ as a weighted mean of W and B

$$\widehat{\text{var}}^{+}(\theta|y) = \frac{N-1}{N}W + \frac{1}{N}B$$

  - this *overestimates* marginal posterior variance if the starting points are overdispersed

$\widehat{R}$

- Estimate total variance $\mathrm{var}(\theta|y)$ as a weighted mean of W and B

$$\widehat{\mathrm{var}}^{+}(\theta|y) = \frac{N-1}{N} W + \frac{1}{N} B$$

  - this *overestimates* marginal posterior variance if the starting points are overdispersed

- Given finite N, W *underestimates* marginal posterior variance
  - single chains have not yet visited all points in the distribution
  - when $N \to \infty, \quad \mathrm{E}(W) \to \mathrm{var}(\theta|y)$

$\widehat{R}$

- Estimate total variance $\mathrm{var}(\theta|y)$ as a weighted mean of W and B

$$\widehat{\mathrm{var}}^{+}(\theta|y) = \frac{N-1}{N}W + \frac{1}{N}B$$

  - this *overestimates* marginal posterior variance if the starting points are overdispersed

- Given finite N, W *underestimates* marginal posterior variance
  - single chains have not yet visited all points in the distribution
  - when $N \to \infty, \quad \mathrm{E}(W) \to \mathrm{var}(\theta|y)$

- As $\widehat{\mathrm{var}}^{+}(\theta|y)$ overestimates and W underestimates, compute

$$\widehat{R} = \sqrt{\frac{\widehat{\mathrm{var}}^{+}}{W}}$$

Subsection 3

S_{eff}, MCSE, and autocorrelation

MCMC draws are dependent

- Monte Carlo estimates still valid (central limit theorem)

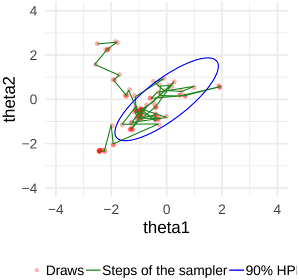
$$E_{p(\theta|y)}[f(\theta)] \approx \frac{1}{S}\sum_{s=1}^{S} f(\theta^{(s)})$$

- Estimation of Monte Carlo error is more difficult
  - evaluation of *effective* sample size, S_{eff}.

MCMC Autocorrelation

- Auto correlation function
  - describes the correlation given a certain lag
  - can be used to compare efficiency of MCMC methods

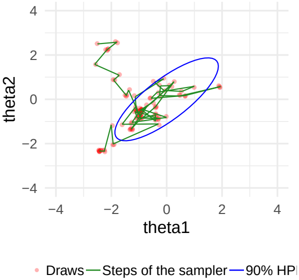
Autocorrelation



UPPSALA UNIVERSITET

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

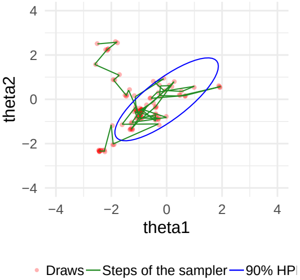
Autocorrelation

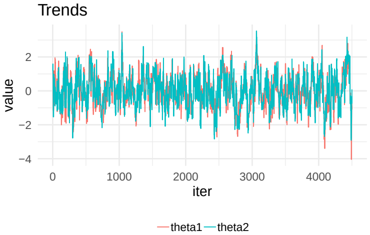


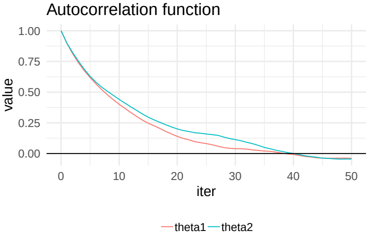
UPPSALA UNIVERSITET

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

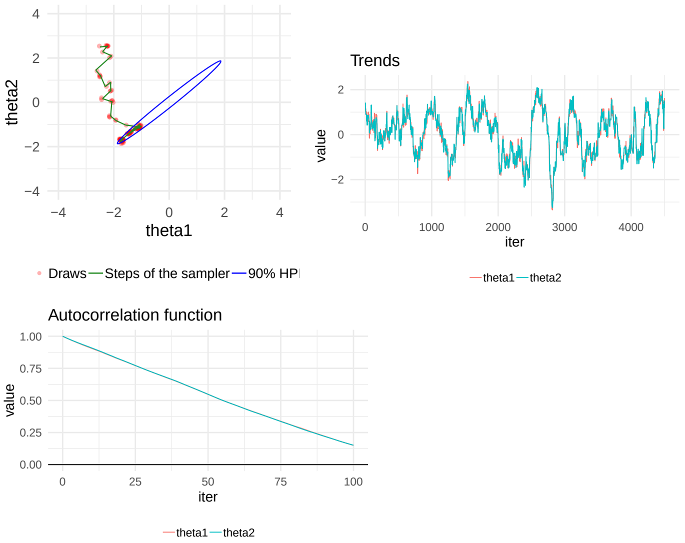
Autocorrelation



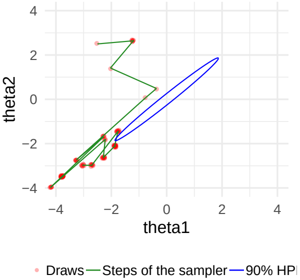


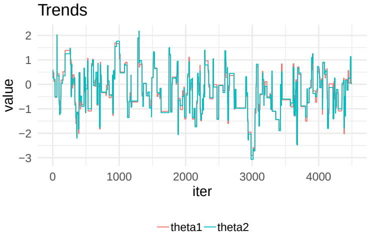


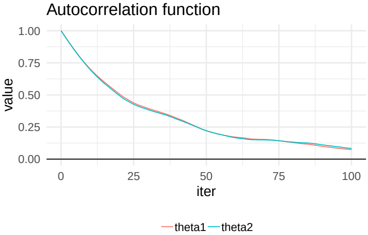
Autocorrelation



Autocorrelation







Autocorrelation

- The autocorrleation can be used to estimate Monte Carlo error in case of MCMC
- For expectation $\bar{\theta}$

$$\mathrm{Var}[\bar{\theta}] = \frac{\sigma_\theta^2}{S_{\mathrm{eff}}}$$

where $S_{\mathrm{eff}} = S/\tau$, and τ is sum of autocorrelations

Autocorrelation

- The autocorrleation can be used to estimate Monte Carlo error in case of MCMC
- For expectation $\bar{\theta}$

$$\mathrm{Var}[\bar{\theta}] = \frac{\sigma_\theta^2}{S_{\text{eff}}}$$

  where $S_{\text{eff}} = S/\tau$, and τ is sum of autocorrelations
  - τ describes how many dependent draws correspond to one independent draw

Autocorrelation

- The autocorrleation can be used to estimate Monte Carlo error in case of MCMC
- For expectation $\bar{\theta}$

$$\mathrm{Var}[\bar{\theta}] = \frac{\sigma_\theta^2}{S_{\text{eff}}}$$

where $S_{\text{eff}} = S/\tau$, and τ is sum of autocorrelations
  - τ describes how many dependent draws correspond to one independent draw
  - Here $S = NM$ (in BDA3 $N = nm$ and $n_{\text{eff}} = N/\tau$)

Autocorrelation

- The autocorrleation can be used to estimate Monte Carlo error in case of MCMC
- For expectation $\bar{\theta}$

$$\mathrm{Var}[\bar{\theta}] = \frac{\sigma_\theta^2}{S_{\text{eff}}}$$

where $S_{\text{eff}} = S/\tau$, and τ is sum of autocorrelations

  - τ describes how many dependent draws correspond to one independent draw
  - Here $S = NM$ (in BDA3 $N = nm$ and $n_{\text{eff}} = N/\tau$)
  - BDA3 focuses on S_{eff} and not the Monte Carlo error directly

Autocorrelation

- Estimation of the autocorrelation using several chains

$$\hat{\rho}_n = 1 - \frac{W - \frac{1}{M}\sum_{m=1}^{M} \hat{\rho}_{n,m}}{2\widehat{\mathrm{var}}^{+}}$$

where $\hat{\rho}_{n,m}$ is autocorrelation at lag n for chain m

UPPSALA UNIVERSITET

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

Autocorrelation

- Estimation of the autocorrelation using several chains

$$\hat{\rho}_n = 1 - \frac{W - \frac{1}{M}\sum_{m=1}^{M} \hat{\rho}_{n,m}}{2\widehat{\mathrm{var}}^{+}}$$

  where $\hat{\rho}_{n,m}$ is autocorrelation at lag n for chain m

- BDA3 has slightly different and less accurate equation. The above equation is used in Stan 2.18+

Autocorrelation

- Estimation of the autocorrelation using several chains

$$\hat{\rho}_n = 1 - \frac{W - \frac{1}{M}\sum_{m=1}^{M} \hat{\rho}_{n,m}}{2\widehat{\mathrm{var}}^{+}}$$

  where $\hat{\rho}_{n,m}$ is autocorrelation at lag n for chain m
- BDA3 has slightly different and less accurate equation. The above equation is used in Stan 2.18+
- Compared to a method which computes the autocorrelation from a single chain, the multi-chain estimate has smaller variance

Estimating τ

- Estimation of τ

$$\tau = 1 + 2\sum_{t=1}^{\infty} \hat{\rho}_t$$

where $\hat{\rho}_t$ is empirical autocorrelation

Estimating τ

- Estimation of τ

$$\tau = 1 + 2\sum_{t=1}^{\infty} \hat{\rho}_t$$

where $\hat{\rho}_t$ is empirical autocorrelation

- empirical autocorrelation function is noisy
- noise is larger for longer lags (less observations)

Estimating τ

- Estimation of τ

$$\tau = 1 + 2\sum_{t=1}^{\infty} \hat{\rho}_t$$

where $\hat{\rho}_t$ is empirical autocorrelation

- empirical autocorrelation function is noisy
- noise is larger for longer lags (less observations)
- less noisy estimate is obtained by truncating

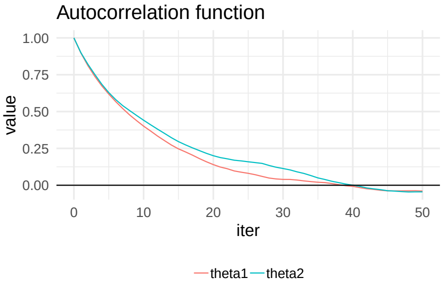
$$\hat{\tau} = 1 + 2\sum_{t=1}^{T} \hat{\rho}_t$$

Geyer's adaptive window estimator of τ

- Truncation T can be decided adaptively
  - for stationary, irreducible, recurrent Markov chain
  - let $\Gamma_m = \rho_{2m} + \rho_{2m+1}$, which is sum of two consequent autocorrelations
  - Γ_m is positive, decreasing and convex function of m

Geyer's adaptive window estimator of τ

- Truncation T can be decided adaptively
  - for stationary, irreducible, recurrent Markov chain
  - let $\Gamma_m = \rho_{2m} + \rho_{2m+1}$, which is sum of two consequent autocorrelations
  - Γ_m is positive, decreasing and convex function of m
- Initial positive sequence estimator (Geyer's IPSE)
  - Choose the largest m so, that all values of the sequence $\hat{\Gamma}_1, \ldots, \hat{\Gamma}_m$ are positive

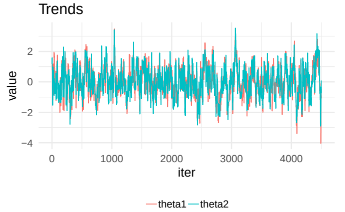


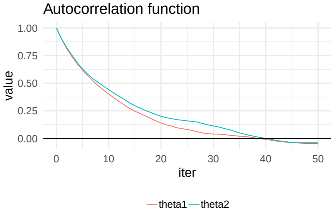
Effective sample size, S_{eff}

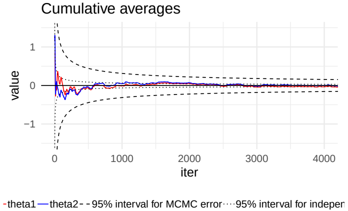
Effective sample size $\mathrm{ESS} = S_{\text{eff}} \approx S/\hat{\tau}$

Effective sample size, S_{eff}

Effective sample size $\mathrm{ESS} = S_{\text{eff}} \approx S/\hat{\tau}$







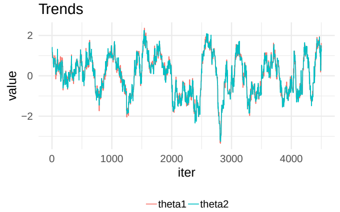
$$\hat{\tau} = 1 + 2\sum_{t=1}^{T} \hat{\rho}_t$$

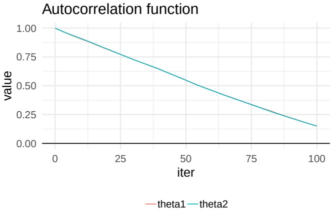
$$\approx 24$$

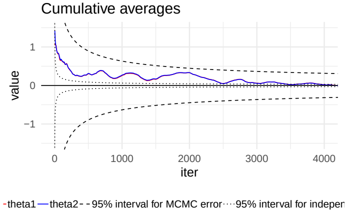
$$\hat{S}_{\text{eff}} = \frac{4500}{24} \approx 188$$

Effective sample size, S_{eff}

Effective sample size $\mathrm{ESS} = S_{\text{eff}} \approx S/\hat{\tau}$







$$\hat{\tau} = 1 + 2\sum_{t=1}^{T} \hat{\rho}_t$$

$$\approx 104$$

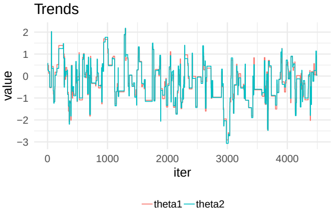
$$\hat{S}_{\text{eff}} = \frac{4500}{104} \approx 43$$

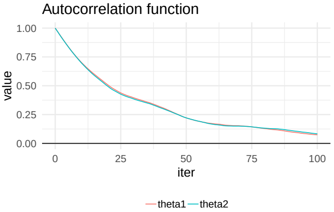
UPPSALA UNIVERSITET

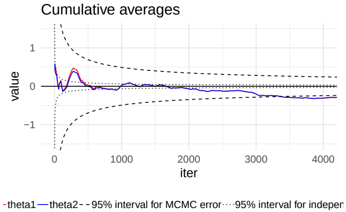
- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

Effective sample size, S_{eff}

Effective sample size $\mathrm{ESS} = S_{\text{eff}} \approx S/\hat{\tau}$







$$\hat{\tau} = 1 + 2\sum_{t=1}^{T} \hat{\rho}_t$$

$$\approx 63$$

$$\hat{S}_{\text{eff}} = \frac{4500}{63} \approx 71$$

UPPSALA UNIVERSITET

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

Section 4

Difficult geometries

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

Problematic distributions

- Nonlinear dependencies
  - optimal proposal depends on location

demo

Problematic distributions

- Nonlinear dependencies
  - optimal proposal depends on location

demo

- Funnels
  - optimal proposal depends on location

demo

Problematic distributions

- Nonlinear dependencies
  - optimal proposal depends on location

demo

- Funnels
  - optimal proposal depends on location

demo

- Multimodal
  - difficult to move from one mode to another

demo

UPPSALA UNIVERSITET

- Monte Carlo recap
- Markov Chain Monte Carlo (MCMC)
  - Gibbs sampling
  - Metropolis-Hastings
- MCMC Diagnostics
  - Warm-up
  - Convergence
  - S_{eff}, MCSE, and autocorrelation
- Difficult geometries

Problematic distributions

- Nonlinear dependencies
  - optimal proposal depends on location

demo

- Funnels
  - optimal proposal depends on location

demo

- Multimodal
  - difficult to move from one mode to another

demo

- Non-identifiable models
  - set of connected points is the mode

demo

Problematic distributions

- Nonlinear dependencies
  - optimal proposal depends on location

demo

- Funnels
  - optimal proposal depends on location

demo

- Multimodal
  - difficult to move from one mode to another

demo

- Non-identifiable models
  - set of connected points is the mode

demo

- Long-tailed with non-finite variance and mean
  - central limit theorem for expectations does not hold