Chapter 11

- 11.1 Gibbs sampler
- 11.2 Metropolis and Metropolis-Hastings
- 11.3 Using Gibbs and Metropolis as building blocks (can be skipped)
- 11.4 Inference and assessing convergence (important)
  - potential scale reduction $\widehat{R}$
- 11.5 Effective number of simulation draws (important)
  - effective sample size S_{eff}
- 11.6 Example: hierarchical normal model (quick glance)

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

- Grid (equal spacing) evaluation with self-normalization

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

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

- Grid (equal spacing) evaluation with self-normalization

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

- 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 for 1D
  - Analytic transformations work for only certain distributions
  - Factorization works only for certain joint distributions
  - Grid evaluation and sampling works in less than a few dimensions
  - Rejection sampling works mostly in 1D (truncation is a special case)
  - Importance sampling is reliable only in special cases

Monte Carlo

- Monte Carlo methods we have discussed so far
    - Inverse CDF works for 1D
    - Analytic transformations work for only certain distributions
    - Factorization works only for certain joint distributions
    - Grid evaluation and sampling works in less than 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 approximation, Variational Bayes*, Expectation Propagation* (Ch 4,13*)

Markov chain

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

Markov chain

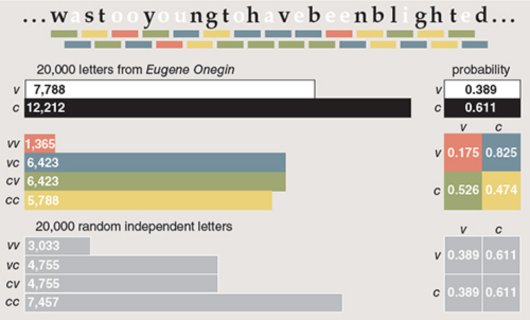
- Andrey Markov proved weak law of large numbers and central limit theorem for certain dependent-random sequences, which were later named Markov chains
- The probability of each event depends only on the state attained in the previous event (or finite number of previous events)

Markov chain

- 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 state attained in the previous event (or finite number of previous events)
- Markov's one example was the sequence of letters in Pushkin's novel "Yevgeniy Onegin" https://www.americanscientist.org/article/first-links-in-the-markov-chain

Markov chain

He was too young to have been blighted
by the cold world's corrupt finesse;
his soul still blossomed out, and lighted
at a friend's word, a girl's caress.
In heart's affairs, a sweet beginner,
he fed on hope's deceptive dinner;
the world's éclat, its thunder-roll,
still captivated his young soul.
He sweetened up with fancy's icing
the uncertainties within his heart;
for him, the objective on life's chart
was still mysterious and enticing—
something to rack his brains about,
suspecting wonders would come out.



- Example of a simple Markov chain: weather example

Markov chain Monte Carlo (MCMC)

- Produce draws $\theta^{(t)}$ given $\theta^{(t-1)}$ from a Markov chain, which has been constructed so that its equilibrium distribution is $p(\theta|y)$

Markov chain Monte Carlo (MCMC)

- Produce draws $\theta^{(t)}$ given $\theta^{(t-1)}$ from a Markov chain, which has been constructed so that its equilibrium distribution is $p(\theta|y)$
  - + 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, which has been constructed so that its equilibrium distribution is $p(\theta|y)$
  - + generic
  - + combine sequence of easier Monte Carlo draws to form a Markov chain
  - + chain goes where most of the posterior mass is
  - + asymptotically chain spends the α% of time where α% posterior mass is

Markov chain Monte Carlo (MCMC)

- Produce draws $\theta^{(t)}$ given $\theta^{(t-1)}$ from a Markov chain, which has been constructed so that its equilibrium distribution is $p(\theta|y)$
  - + generic
  - + combine sequence of easier Monte Carlo draws to form a Markov chain
  - + chain goes where most of the posterior mass is
  - + asymptotically chain spends the α% of time where α% posterior mass is
  - + central limit theorem holds for expectations

Markov chain Monte Carlo (MCMC)

- Produce draws $\theta^{(t)}$ given $\theta^{(t-1)}$ from a Markov chain, which has been constructed so that its equilibrium distribution is $p(\theta|y)$
  - + generic
  - + combine sequence of easier Monte Carlo draws to form a Markov chain
  - + chain goes where most of the posterior mass is
  - + asymptotically chain spends the α% of time where α% posterior mass is
  - + central limit theorem holds for expectations
  - - draws are dependent
  - - construction of efficient Markov chains is not always easy

Markov chain

- Set of random variables $\theta^1, \theta^2, \ldots$, so that with all values of t, 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
- Transition distribution $T_t(\theta^t|\theta^{t-1})$ (may depend on t)
- By choosing a suitable transition distribution, the stationary distribution of Markov chain is $p(\theta|y)$

Gibbs sampling (or alternating conditional sampling)

- Given $\theta = (\theta_1, \ldots, \theta_d)$, alternate sampling from 1D conditional distributions
  - 1D is easy even if no conjugate prior and analytic posterior

Gibbs sampling (or alternating conditional sampling)

- Given $\theta = (\theta_1, \ldots, \theta_d)$, alternate sampling from 1D conditional distributions
  - 1D is easy even if no conjugate prior and analytic posterior
- Basic algorithm: at each iteration t

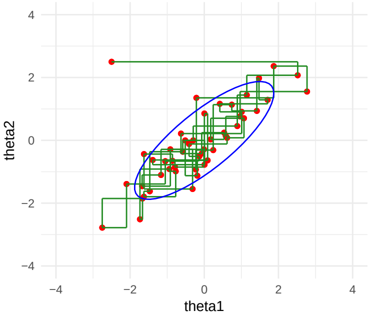
$$\begin{aligned} \text{sample } \theta_j^t \text{ from} \quad & p(\theta_j | \theta_{-j}^{t-1}, y), \\ \text{where} \quad & \theta_{-j}^{t-1} = (\theta_1^t, \ldots, \theta_{j-1}^t, \theta_{j+1}^{t-1}, \ldots, \theta_d^{t-1}) \end{aligned}$$

Gibbs sampling (or alternating conditional sampling)

- Given $\theta = (\theta_1, \ldots, \theta_d)$, alternate sampling from 1D conditional distributions
  - 1D is easy even if no conjugate prior and analytic posterior
- Basic algorithm: at each iteration t

$$\text{sample } \theta_j^t \text{ from} \quad p(\theta_j | \theta_{-j}^{t-1}, y),$$
$$\text{where} \quad \theta_{-j}^{t-1} = (\theta_1^t, \ldots, \theta_{j-1}^t, \theta_{j+1}^{t-1}, \ldots, \theta_d^{t-1})$$

- demo



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
  (cf. proposal distribution in Metropolis algorithm)

Gibbs sampling

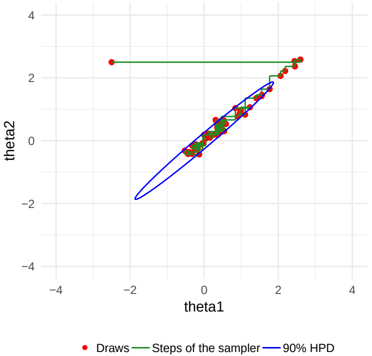
- With *conditionally* conjugate priors, the sampling from the conditional distributions is easy for wide range of models
  - BUGS/WinBUGS/OpenBUGS/JAGS
- No algorithm parameters to tune
  (cf. proposal distribution in Metropolis algorithm)
- For not so easy conditionals, use e.g. inverse-CDF

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
  (cf. proposal distribution in Metropolis algorithm)
- 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
  (cf. proposal distribution in Metropolis algorithm)
- 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
  - demo continues



Conditional vs joint

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

Conditional vs joint

- How about sampling θ jointly?
  - e.g. it is easy to sample from multivariate normal
- Can we use that to form a Markov chain?
  http://elevanth.org/blog/2017/11/28/build-a-better-markov-chain/

Metropolis algorithm

- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \ldots$
     (a) pick a proposal θ^* from the 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

Metropolis algorithm

- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \ldots$
     (a) pick a proposal θ^* from the 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

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

Metropolis algorithm

- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \ldots$
     (a) pick a proposal θ^* from the 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

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

Metropolis algorithm

- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \ldots$
     (a) pick a proposal θ^* from the 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

Metropolis algorithm

- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \ldots$
     (a) pick a proposal θ^* from the 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 (see 'Note')

Metropolis algorithm

- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \ldots$
     (a) pick a proposal θ^* from the 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 (see 'Note')
  - step c is executed by generating a random number from U$(0, 1)$

Metropolis algorithm

- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \ldots$
     (a) pick a proposal θ^* from the 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 (see 'Note')
  - step c is executed by generating a random number from 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

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

  - proposal distribution $J_t(\theta^*|\theta^{t-1}) = \mathrm{N}(\theta^*|\theta^{t-1}, \sigma_p^2)$
- Demo
  http://elevanth.org/blog/2017/11/28/build-a-better-markov-chain/,
  source: https://chi-feng.github.io/mcmc-demo/app.html?
  algorithm=RandomWalkMH&target=standard

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

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

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

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
      - 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 $p(\theta|y)$
    - consider starting algorithm at time $t-1$ with a draw $\theta^{t-1} \sim p(\theta|y)$

Why Metropolis algorithm works

2. Prove that this stationary distribution is the desired target distribution $p(\theta|y)$
   - consider starting algorithm at time $t-1$ with a draw $\theta^{t-1} \sim p(\theta|y)$
   - consider any two such points θ_a and θ_b drawn from $p(\theta|y)$ and labeled so that $p(\theta_b|y) \geq p(\theta_a|y)$

Why Metropolis algorithm works

2. Prove that this stationary distribution is the desired target distribution $p(\theta|y)$
   - consider starting algorithm at time $t-1$ with a draw $\theta^{t-1} \sim p(\theta|y)$
   - consider any two such points θ_a and θ_b drawn from $p(\theta|y)$ and labeled so that $p(\theta_b|y) \geq p(\theta_a|y)$
   - the unconditional probability density of a transition from θ_a to θ_b is
   $$p(\theta^{t-1} = \theta_a, \theta^t = \theta_b) = p(\theta_a|y)J_t(\theta_b|\theta_a),$$

Why Metropolis algorithm works

2. Prove that this stationary distribution is the desired target distribution $p(\theta|y)$
   - consider starting algorithm at time $t-1$ with a draw $\theta^{t-1} \sim p(\theta|y)$
   - consider any two such points θ_a and θ_b drawn from $p(\theta|y)$ and labeled so that $p(\theta_b|y) \geq p(\theta_a|y)$
   - the unconditional probability density of a transition from θ_a to θ_b is
   $$p(\theta^{t-1} = \theta_a, \theta^t = \theta_b) = p(\theta_a|y)J_t(\theta_b|\theta_a),$$
   - the unconditional probability density of a transition from θ_b to θ_a is
   $$\begin{aligned} p(\theta^t = \theta_a, \theta^{t-1} = \theta_b) &= p(\theta_b|y)J_t(\theta_a|\theta_b)\left(\frac{p(\theta_a|y)}{p(\theta_b|y)}\right) \\ &= p(\theta_a|y)J_t(\theta_a|\theta_b), \end{aligned}$$

Why Metropolis algorithm works

2. Prove that this stationary distribution is the desired target distribution $p(\theta|y)$
   - consider starting algorithm at time $t-1$ with a draw $\theta^{t-1} \sim p(\theta|y)$
   - consider any two such points θ_a and θ_b drawn from $p(\theta|y)$ and labeled so that $p(\theta_b|y) \geq p(\theta_a|y)$
   - the unconditional probability density of a transition from θ_a to θ_b is
   $$p(\theta^{t-1} = \theta_a, \theta^t = \theta_b) = p(\theta_a|y)J_t(\theta_b|\theta_a),$$
   - the unconditional probability density of a transition from θ_b to θ_a is
   $$\begin{aligned} p(\theta^t = \theta_a, \theta^{t-1} = \theta_b) &= p(\theta_b|y)J_t(\theta_a|\theta_b)\left(\frac{p(\theta_a|y)}{p(\theta_b|y)}\right) \\ &= p(\theta_a|y)J_t(\theta_a|\theta_b), \end{aligned}$$

   which is the same as the probability of transition from θ_a to θ_b, since we have required that $J_t(\cdot|\cdot)$ is symmetric

Why Metropolis algorithm works

2. Prove that this stationary distribution is the desired target distribution $p(\theta|y)$
   - consider starting algorithm at time $t-1$ with a draw $\theta^{t-1} \sim p(\theta|y)$
   - consider any two such points θ_a and θ_b drawn from $p(\theta|y)$ and labeled so that $p(\theta_b|y) \geq p(\theta_a|y)$
   - the unconditional probability density of a transition from θ_a to θ_b is
   $$p(\theta^{t-1} = \theta_a, \theta^t = \theta_b) = p(\theta_a|y) J_t(\theta_b|\theta_a),$$
   - the unconditional probability density of a transition from θ_b to θ_a is
   $$\begin{aligned} p(\theta^t = \theta_a, \theta^{t-1} = \theta_b) &= p(\theta_b|y) J_t(\theta_a|\theta_b) \left(\frac{p(\theta_a|y)}{p(\theta_b|y)} \right) \\ &= p(\theta_a|y) J_t(\theta_a|\theta_b), \end{aligned}$$
   which is the same as the probability of transition from θ_a to θ_b, since we have required that $J_t(\cdot|\cdot)$ is symmetric
   - since their joint distribution is symmetric, θ^t and θ^{t-1} have the same marginal distributions, and so $p(\theta|y)$ is the stationary distribution of the Markov chain of θ

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 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

Metropolis-Hastings algorithm

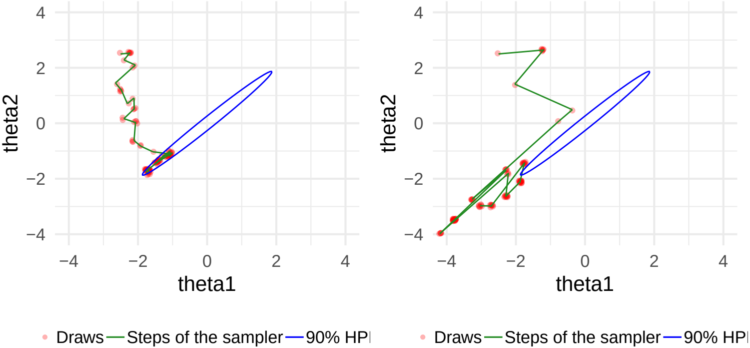
- Ideal proposal distribution is the distribution itself
  - $J(\theta^*|\theta) \equiv p(\theta^*|y)$ for all θ
  - acceptance probability is 1
  - independent draws
  - not usually feasible
- Good proposal distribution resembles the target distribution
  - if the shape of the target distribution is unknown, usually normal or *t* distribution is used
- After the 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
- Generic rule for rejection rate is 60-90% (but depends on dimensionality and a specific algorithm variation)

Gibbs sampling

- 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
  - demo



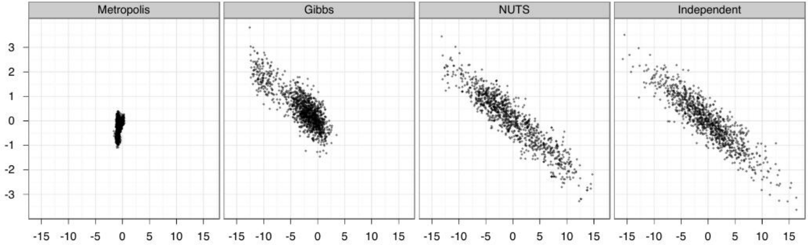
Dynamic Hamiltonian Monte Carlo and NUTS

- Chapter 12 presents some more advanced methods
  - Chapter 12 includes Hamiltonian Monte Carlo and NUTS, which is one of the most efficient methods
    - uses gradient information
    - Hamiltonian dynamic simulation reduces random walk
    - Demo http://elevanth.org/blog/2017/11/28/build-a-better-markov-chain/, source: https://chi-feng.github.io/mcmc-demo/app.html?algorithm=HamiltonianMC&target=donut

HMC / NUTS

Comparison of algorithms on **highly correlated** 250-dimensional Gaussian distribution

- Do **1,000,000** draws with both Random Walk Metropolis and Gibbs, thinning by 1000
- Do **1,000** draws using Stan's NUTS algorithm (no thinning)
- Do 1,000 independent draws (we can do this for multivariate normal)



Source: Jonah Gabry

Iterative simulation

Difficulties of inference from iterative simulation

1. if the iterations have not proceeded long enough, the simulations may be grossly unrepresentative of the target distribution.
2. within-sequence correlation

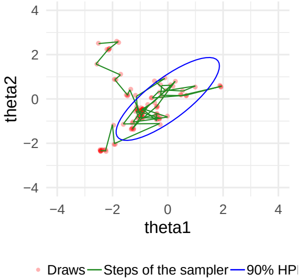
    aside from any convergence issues, simulation inference from correlated draws is generally less precise than from the same number of independent draws $\Rightarrow$ effective sample size

Warm-up and convergence diagnostics

- Asymptotically chain spends the $\alpha\%$ of time where $\alpha\%$ posterior mass is

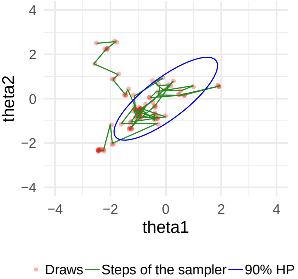
Warm-up and convergence diagnostics

- Asymptotically chain spends the α% of time where α% posterior mass is
  - but in finite time the initial part of the chain may be non-representative and lower error of the estimate can be obtained by throwing it away



Warm-up and convergence diagnostics

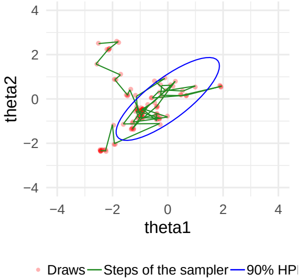
- Asymptotically chain spends the α% of time where α% posterior mass is
  - but in finite time the initial part of the chain may be non-representative and lower error of the estimate can be obtained by throwing it away



- Warm-up = remove draws from the beginning of the chain
  - warm-up may include also phase for adapting algorithm parameters

Warm-up and convergence diagnostics

- Asymptotically chain spends the α% of time where α% posterior mass is
  - but in finite time the initial part of the chain may be non-representative and lower error of the estimate can be obtained by throwing it away



- Warm-up = remove draws from the beginning of the chain
  - warm-up may include also phase for adapting algorithm parameters
- Convergence diagnostics
  - Do we get samples from the target distribution?

MCMC draws are dependent

- Monte Carlo estimates still valid (central limit theorem holds)

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

- Estimation of Monte Carlo error is more difficult
  - evaluation of *effective* sample size

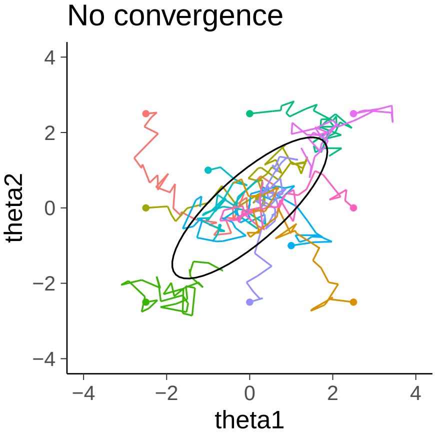
Remedies

We handle the special problems of iterative simulation in three ways

1. multiple sequences with starting points dispersed throughout parameter space
2. we monitor the convergence of all quantities of interest by comparing variation between and within simulated sequences
3. if the simulation efficiency is unacceptably low (in the sense of requiring too much real time), the algorithm can be altered

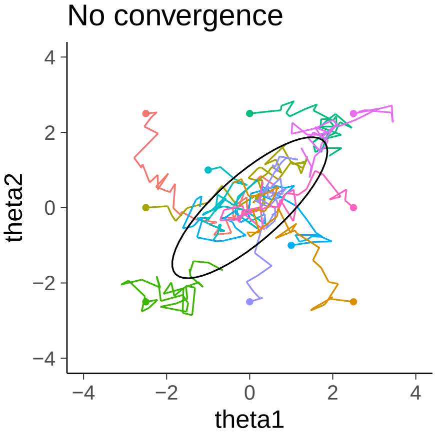
Several chains

- Use of several chains make convergence diagnostics easier
- Start chains from different starting points – preferably overdispersed



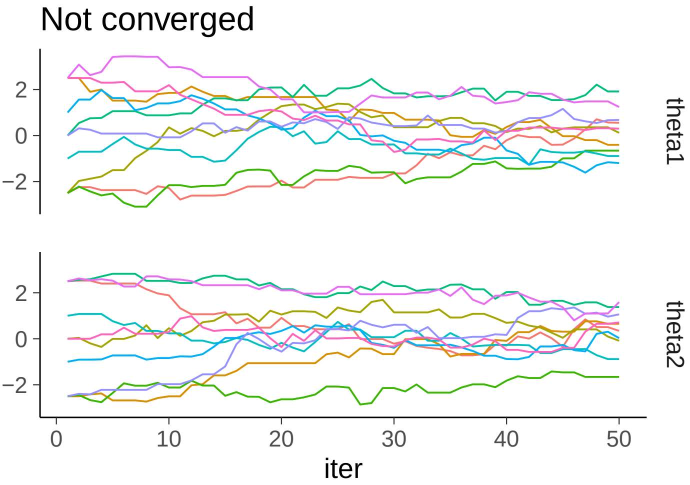
Several chains

- Use of several chains make convergence diagnostics easier
- Start chains from different starting points – preferably overdispersed

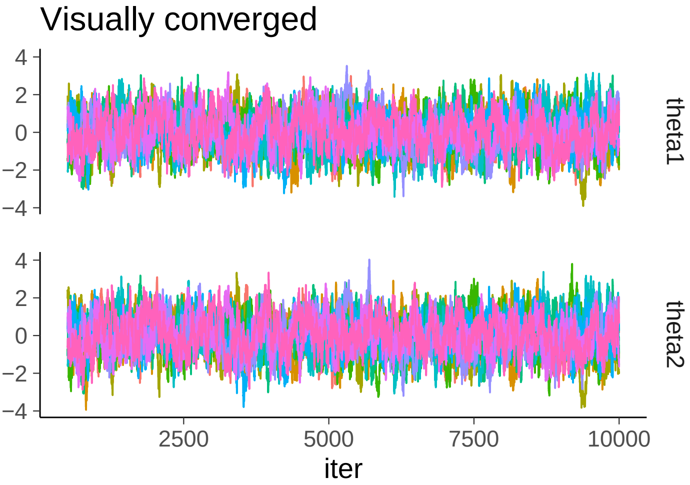


- Remove draws from the beginning of the chains and run chains long enough so that it is not possible to distinguish where each chain started and the chains are well mixed

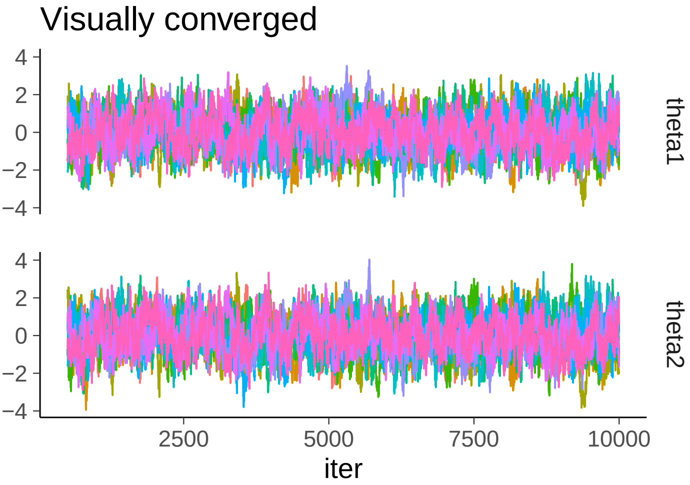
Several chains



Several chains



Several chains



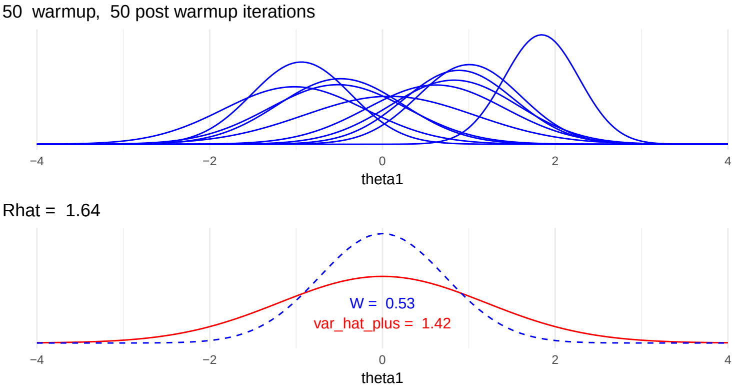
Visual convergence check is not sufficient

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

- BDA3: $\widehat{R}$ aka *potential scale reduction factor* (PSRF)
- Compare means and variances of the chains

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

- BDA3: $\widehat{R}$ aka *potential scale reduction factor* (PSRF)
- Compare means and variances of the chains
  W = within chain variance estimate
  var_hat_plus = total variance estimate

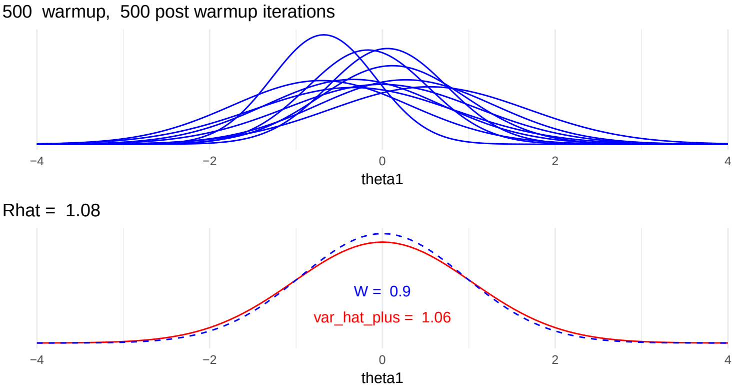


$\widehat{R}$: comparison of within and between variances of the chains

- BDA3: $\widehat{R}$ aka *potential scale reduction factor* (PSRF)
- Compare means and variances of the chains

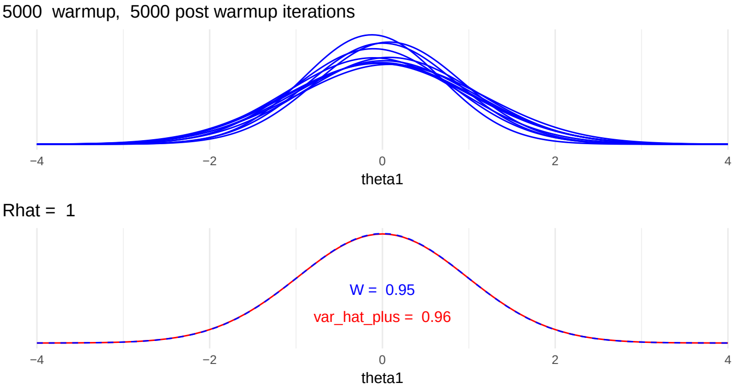
  W = within chain variance estimate

  var_hat_plus = total variance estimate



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

- BDA3: $\widehat{R}$ aka *potential scale reduction factor* (PSRF)
- Compare means and variances of the chains
  W = within chain variance estimate
  var_hat_plus = total variance estimate



$\widehat{R}$

- M chains, each having N draws (with new R-hat notation)

$\widehat{R}$

- M chains, each having N draws (with new R-hat notation)
- 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 (with new R-hat notation)
- 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

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

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

$\widehat{R}$

- M chains, each having N draws (with new R-hat notation)
- 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 (with new R-hat notation)
- 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
  - but is unbiased under stationarity (that is, if the starting distribution equals the target distribution), or in the limit $N \to \infty$

$\widehat{R}$

- 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
  - but is unbiased under stationarity (that is, if the starting distribution equals the target distribution), or in the limit $N \to \infty$
- Given finite N, W underestimates marginal posterior variance
  - single chains have not yet visited all points in the distribution
  - when $N \to \infty, \quad \text{E}(W) \to \text{var}(\theta|y)$

$\widehat{R}$

- 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
  - but is unbiased under stationarity (that is, if the starting distribution equals the target distribution), or in the limit $N \to \infty$

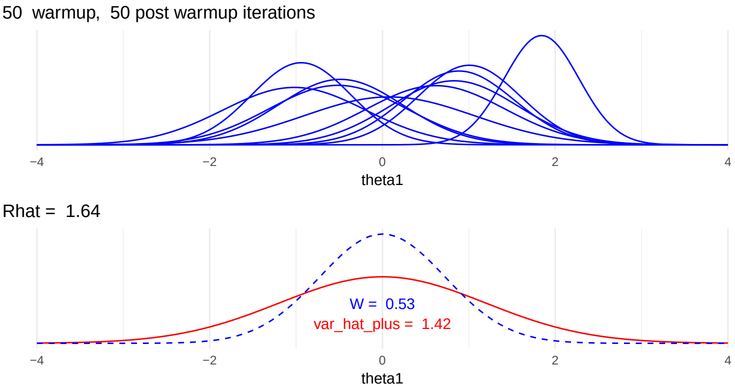
- Given finite N, W underestimates marginal posterior variance
  - single chains have not yet visited all points in the distribution
  - when $N \to \infty, \quad \mathrm{E}(W) \to \mathrm{var}(\theta|y)$

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

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

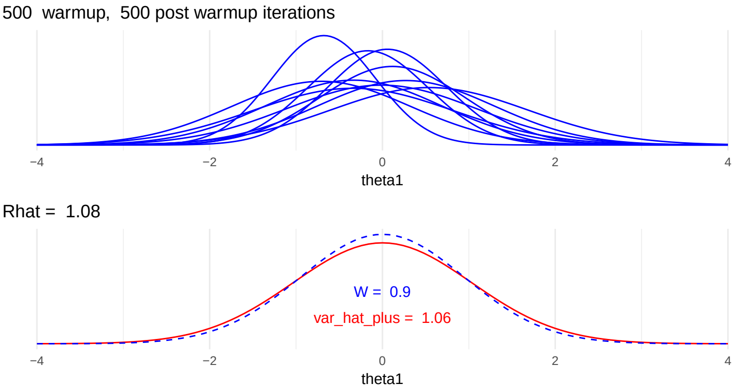
$\widehat{R}$

- BDA3: $\widehat{R}$ aka *potential scale reduction factor* (PSRF)
- Compare means and variances of the chains
  W = within chain variance estimate
  var_hat_plus = total variance estimate



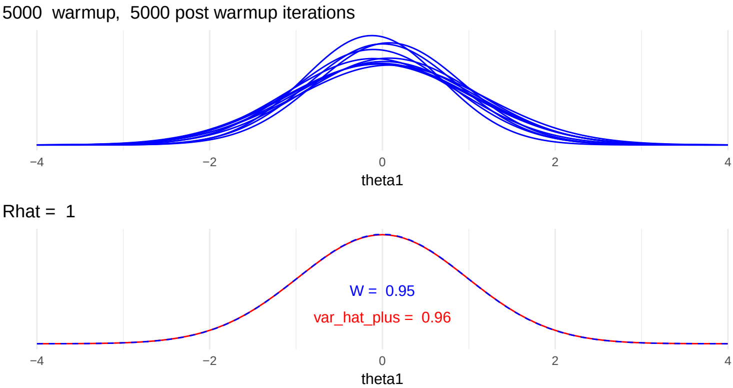
$\widehat{R}$

- BDA3: $\widehat{R}$ aka *potential scale reduction factor* (PSRF)
- Compare means and variances of the chains
  W = within chain variance estimate
  var_hat_plus = total variance estimate



$\widehat{R}$

- BDA3: $\widehat{R}$ aka *potential scale reduction factor* (PSRF)
- Compare means and variances of the chains
  W = within chain variance estimate
  var_hat_plus = total variance estimate



$\widehat{R}$

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

- Estimates how much the scale of a scalar estimand θ could reduce if $N \to \infty$
- $\widehat{R} \to 1$, when $N \to \infty$
- if $\widehat{R}$ is big (e.g., $R > 1.10$), keep sampling

$\widehat{R}$

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

- Estimates how much the scale of a scalar estimand θ could reduce if $N \to \infty$
- $\widehat{R} \to 1$, when $N \to \infty$
- if $\widehat{R}$ is big (e.g., $R > 1.10$), keep sampling
- If $\widehat{R}$ close to 1, it is still possible that chains have not converged
  - if starting points were not overdispersed
  - distribution far from normal (especially if infinite variance)
  - just by chance when n is finite

Split-$\widehat{R}$

- BDA3: split-$\widehat{R}$
- Examines *mixing* and *stationarity* of chains
- To examine stationarity chains are split to two parts
    - after splitting, we have M chains, each having N draws
    - scalar draws $\theta_{nm} \quad (n = 1, \ldots, N; m = 1, \ldots, M)$
    - compare means and variances of the split chains

Rank normalized $\widehat{R}$

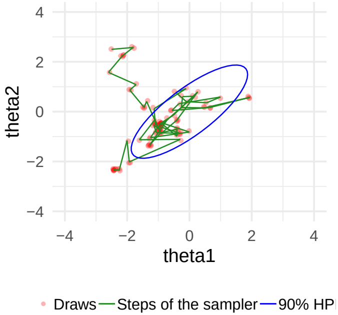
- Original $\widehat{R}$ requires that the target distribution has finite mean and variance
- Rank normalization fixes this and is also more robust given finite but high variance
- Folding improves detecting scale differences between chains
- Paper proposes also local convergence diagnostics and practical MCSE estimates for quantiles
- Notation updated compared to BDA3

Vehtari, Gelman, Simpson, Carpenter, Bürkner (2020). Rank-normalization, folding, and localization: An improved R-hat for assessing convergence of MCMC. Bayesian Analysis, doi:10.1214/20-BA1221. https://projecteuclid.org/euclid.ba/1593828229.

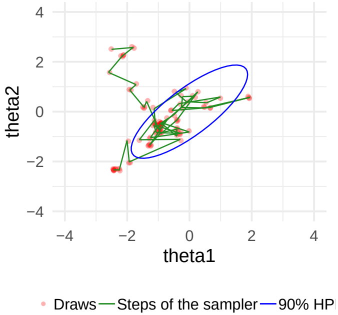
Time series analysis

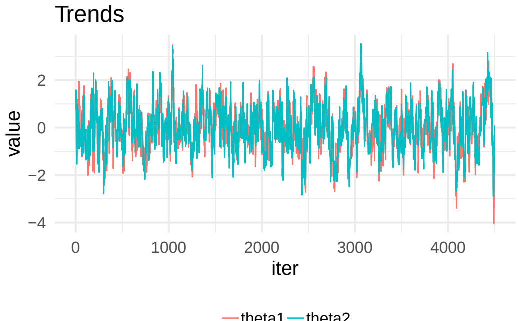
- Auto correlation function
  - describes the correlation given a certain lag
  - can be used to compare efficiency of MCMC algorithms and parameterizations

Auto correlation

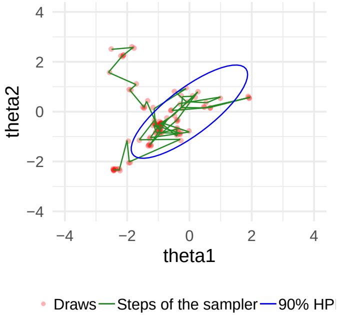


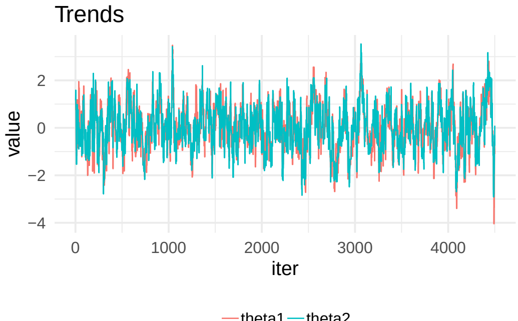
Auto correlation



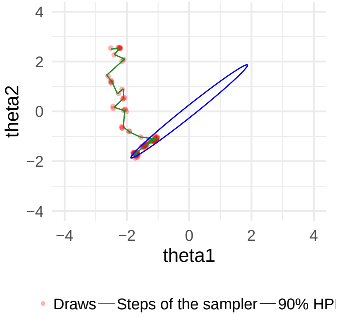


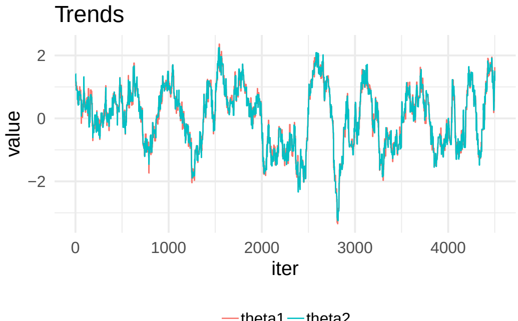
Auto correlation



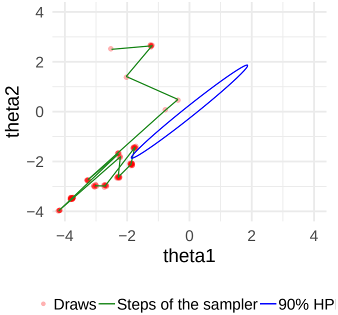


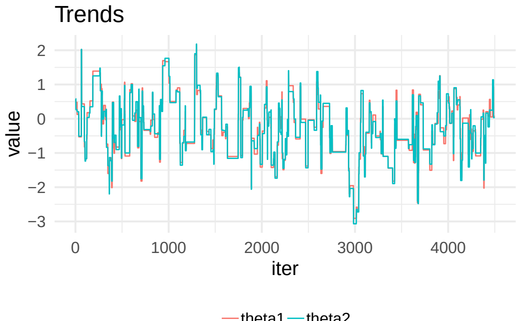
Auto correlation





Auto correlation





Time series analysis

- Time series analysis 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

Time series analysis

- Time series analysis 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
  - τ describes how many dependent draws correspond to one independent sample

Time series analysis

- Time series analysis 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 sample
  - new R-hat paper $S = NM$ (in BDA3 $N = nm$ and $n_{\text{eff}} = N/\tau$)

Time series analysis

- Time series analysis 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
  - τ describes how many dependent draws correspond to one independent sample
  - new R-hat paper $S = NM$ (in BDA3 $N = nm$ and $n_{\mathrm{eff}} = N/\tau$)
  - BDA3 focuses on S_{eff} and not the Monte Carlo error directly new R-hat paper discusses more about MCSEs for different quantities

Time series analysis

- 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

Time series analysis

- Estimation of the autocorrelation using several chains

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

  where $\hat{\rho}_{n,m}$ is autocorrelation at lag n for chain m
- This combines $\widehat{R}$ and autocorrelation estimates
  - takes into account if the chains are not mixing (the chains have not converged)

Time series analysis

- Estimation of the autocorrelation using several chains

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

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

- This combines $\widehat{R}$ and autocorrelation estimates
  - takes into account if the chains are not mixing (the chains have not converged)
- BDA3 has slightly different and less accurate equation. The above equation is used in Stan 2.18+

Time series analysis

- Estimation of the autocorrelation using several chains

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

  where $\hat{\rho}_{n,m}$ is autocorrelation at lag n for chain m
- This combines $\widehat{R}$ and autocorrelation estimates
  - takes into account if the chains are not mixing (the chains have not converged)
- 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

Time series analysis

- Estimation of τ

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

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

Time series analysis

- 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 and thus estimate of τ is noisy
  - noise is larger for longer lags (less observations)

Time series analysis

- 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 and thus estimate of τ is noisy
  - noise is larger for longer lags (less observations)
  - less noisy estimate is obtained by truncating

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

Time series analysis

- 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 and thus estimate of τ is noisy
  - noise is larger for longer lags (less observations)
  - less noisy estimate is obtained by truncating

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

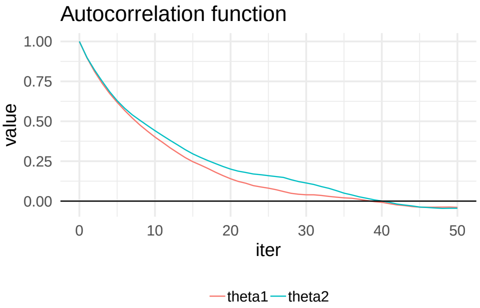
- As τ is estimated from a finite number of draws, it's expectation is overoptimistic
  - if $\hat{\tau} > MN/20$ then the estimate is unreliable

Geyer's adaptive window estimator

- Truncation 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

- Truncation 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

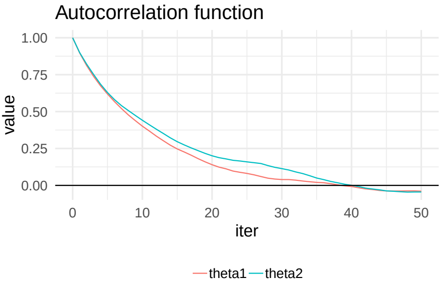


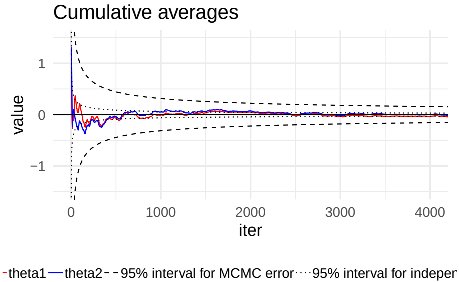
Effective sample size

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

Effective sample size

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



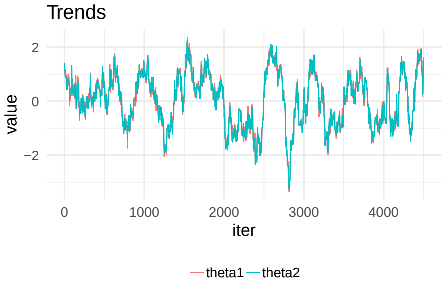


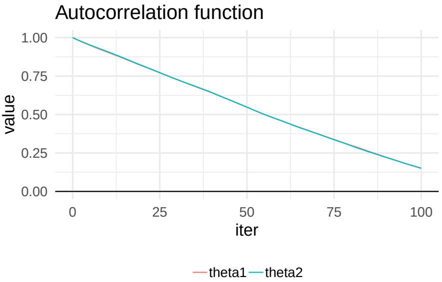
$$\hat{\tau} = 1 + 2\sum_{t=1}^{T} \hat{\rho}_t$$

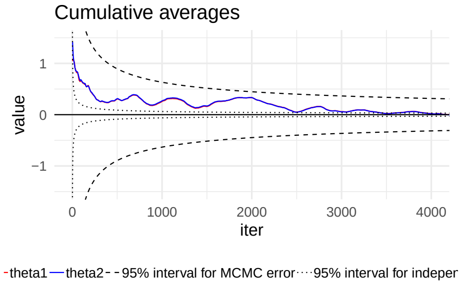
$$\approx 24$$

Effective sample size

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



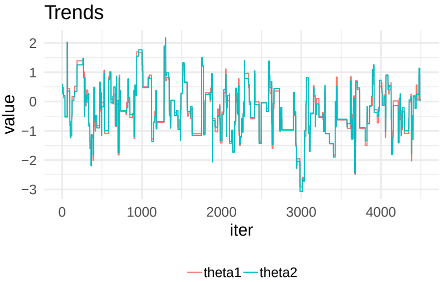


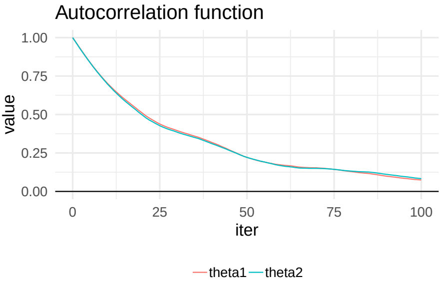


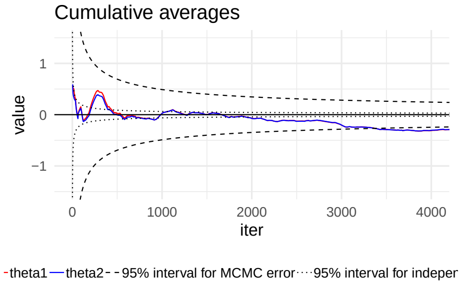
$$\hat{\tau} = 1 + 2\sum_{t=1}^{T} \hat{\rho}_t$$
$$\approx 104$$

Effective sample size

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







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

$$\approx 63$$

Problematic distributions

- Nonlinear dependencies
    - optimal proposal depends on location

Problematic distributions

- Nonlinear dependencies
  - optimal proposal depends on location
- Funnels
  - optimal proposal depends on location

Problematic distributions

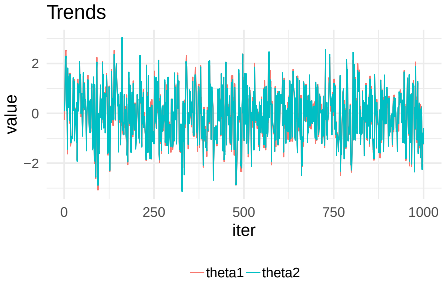
- Nonlinear dependencies
  - optimal proposal depends on location
- Funnels
  - optimal proposal depends on location
- Multimodal
  - difficult to move from one mode to another

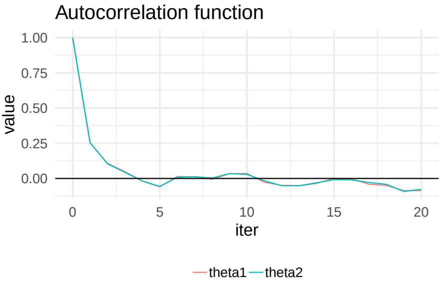
Problematic distributions

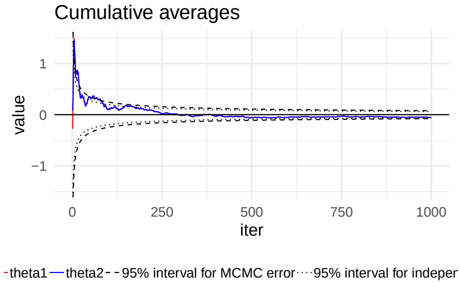
- Nonlinear dependencies
  - optimal proposal depends on location
- Funnels
  - optimal proposal depends on location
- Multimodal
  - difficult to move from one mode to another
- Long-tailed with non-finite variance and mean
  - central limit theorem for expectations does not hold

Ch. 12: HMC, NUTS, and dynamic HMC

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







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

$$\approx 1.6$$

Further diagnostics

- Dynamic HMC/NUTS has additional diagnostics
  - divergences
  - tree depth exceedences
  - fraction of missing information

Note

- Basics of Markov chains: http://www.netlab.tkk.fi/opetus/s38143/luennot/english.shtml, see Lecture 4.
  To prove that Metropolis algorithm works, it is sufficient to show that chain is irreducible, aperiodic and not transient.
- Nice animations with discussion http://elevanth.org/blog/2017/11/28/build-a-better-markov-chain/
  And just the animations with more options to experiment https://chi-feng.github.io/mcmc-demo/

- Metropolis algorithm
  Don't confuse rejection in the rejection sampling and in Metropolis algorithm.
  In the rejection sampling, the rejected samples are thrown away.
  In Metropolis algorithm the rejected proposals are thrown away, but time moves on and the previous sample $x(t)$ is also the sample $x(t+1)$.
    - When rejecting a proposal, the previous sample is repeated in the chain, they have to be included and they are valid samples from the distribution.
    - For basic Metropolis, it can be shown that optimal rejection rate is 55–77%, so that on even the optimal case quite many of the samples are repeated samples.
      However, high number of rejections is acceptable as then the accepted proposals are on average further away from the previous point. It is better to jump further away 23–45% of time than more often to jump really close.

- Transition distribution vs. proposal distribution Transition distribution is a property of Markov chain. In Metropolis algorithm the transition distribution is a mixture of a proposal distribution and a point mass in the current point. BDA uses also term jumping distribution to refer to proposal distribution.
- Convergence
  Theoretical convergence in an infinite time is different than practical convergence in a finite time. There is no exact moment when chain has converged and thus it is not possible to detect when the chain has converged (except for rare perfect sampling methods not discussed in BDA3).

  The convergence diagnostics can help to find out if the chain is unlikely to be representative of the target distribution. Furthermore, even if would be able to start from a independent sample from the posterior so that chain starts from the convergence, the mixing can be so slow that we may require very large number of samples before the samples are representative of the target distribution.

If starting point is selected at or near the mode, less time is needed to reach the area of essential mass, but still the samples in the beginning of the chain are not presentative of the true distribution unless the starting point was somehow samples directly from the target distribution.

Laplace, VB, EP: motivation

- MCMC methods could be expensive to compute, especially for large sample sizes n.
- Moreover, many MCMC algorithms require a rough estimate of some key posterior quantities, such as the posterior variance.
- These issues motivates the development of **deterministic approximations** of the posterior distribution
- Compared to MCMC methods, the accuracy of this class of approximations can not be reduced by running the algorithm longer.
- On the other hand, deterministic approximations are typically very fast to compute and sufficiently reliable in several applied contexts.

Laplace approximation

- Let $\pi(\theta|\mathbf{X})$ be a continuous and differentiable posterior density in $\Theta \subseteq \Re^p$
- The Laplace approximation is one of the first approximation methods that has been proposed. It was known even before the advent of MCMC.
- The key idea is approximating the log-posterior density $\log \pi(\theta|\mathbf{X})$ using a **Taylor expansion** around the mode $\hat{\theta}_{MAP}$ yielding

$$\log \pi(\theta|\mathbf{X}) \approx \log \pi(\hat{\theta}_{MAP}|\mathbf{X}) - \frac{1}{2}(\theta - \hat{\theta}_{MAP})^T \hat{M}(\theta - \hat{\theta}_{MAP}) + \text{const}$$

where M is the negative Hessian of $\log \pi(\theta|\mathbf{X})$ evaluated at $\hat{\theta}_{MAP}$, ie,

$$\hat{M} = -\left.\frac{\partial^2}{\partial\theta\partial\theta^T} \log \pi(\theta|\mathbf{X})\right|_{\theta=\hat{\theta}_{MAP}}$$

- This expansion leads to the following multivariate Gaussian approximate posterior

$$\pi(\theta|\mathbf{X}) \approx N_p(\hat{\theta}_{MAP}, \hat{M}^{-1})$$

- A fairly strong asymptotic justification of the Laplace approximation is based on the Bernstein–von Mises theorem.
- Here we are also assuming that $\hat{\theta}_{MAP}$ and $n\hat{M}^{-1}$ are consistent estimators for the "true" parameter value θ_0 and for the inverse Fisher information matrix, respectively.
- Hence, in several cases and for n large enough, the law $\pi(\theta|\mathbf{X})$ is roughly a Gaussian centered at the mode and with variance depending on the Fisher information.

+ The Laplace approximation is an old and simple method that has appealing asymptotic guarantees. Moreover, it only requires the computation of the Hessian and the MAP.

- Refined higher order improvements of expected posterior functionals can be obtained.
  - On the other hand, especially when the sample size n is relatively small, the quadratic approximation of $\log \pi(\theta|\mathbf{X})$ may perform poorly.
  - For example, if the posterior is not symmetric and unimodal, the MAP is not a good estimate for the posterior mean, thus leading to inaccurate Gaussian approximations.
  - Furthermore, if the parameter space Θ is bounded, a Gaussian approximation could be quite problematic $\Rightarrow$ a reparametrization should be considered.
  - Finally, it is unclear how to handle discrete parameter spaces.

Approximation methods

- Let $\pi(\theta|\mathbf{X})$ be the intractable posterior distribution and let $q(\theta)$ be a density belonging to $\mathcal{Q}$, where $\mathcal{Q}$ is a general class of tractable densities.
- An optimal approximation $\hat{q}(\theta) \in \mathcal{Q}$ of the posterior distribution is defined as

$$\hat{q}(\theta) = \arg\min_{q \in \mathcal{Q}} \mathcal{D}\{q(\theta), \pi(\theta|\mathbf{X})\}$$

  where $\mathcal{D}$ is some divergence or metric over the space of probability distributions.
- An example is the Kullback-Leibler divergence $\mathcal{D}(\cdot, \cdot) = KL(\cdot||\cdot)$
  - $KL(q(\theta||\pi(\theta|\mathbf{X}))$ divergence leads to the variational Bayes method
  - $KL(\pi(\theta|\mathbf{X})||q(\theta)$ divergence leads to the expectation propagation method