# Chapter 11

- 11.1 Gibbs sampler
- 11.2 Metropolis and Metropolis-Hastings
- 11.3 Using Gibbs and Metropolis as building blocks
- 11.4 Inference and assessing convergence
- 11.5 Effective number of simulation draws
- 11.6 Example: hierarchical normal model

# Chapter 11 demos

- demo11_1.m: Gibbs sampling
- demo11_2.m: Metropolis sampling
- demo11_3.m: Convergence of Markov chain
- demo11_4.m: potential scale reduction $\hat{R}$
- demo11_5.m: effective number of samples $n_{\text{eff}}$ and thinning

# Markov chain Monte Carlo (MCMC)

- produce samples $\theta^{(t)}$, from a Markov chain, which has been constructed so that its equilibrium distribution is $p(\theta|y)$.
  - \+ generic
  - \+ chain goes where most of the posterior mass is
  - \- samples 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 $t$

$$p(\theta^t|\theta^1, \ldots, \theta^{(t-1)}) = p(\theta^t|\theta^{(t-1)})$$

- starting point $\theta^0$
- 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

- When using conditionally conjugate (hyper)priors, the sampling from the conditional distributions is easy for wide range of models
  - e.g. hierarchical normal distribution model
  - WinBUGS/OpenBUGS/JAGS
- No algorithm parameters to tune
  (cf. proposal distribution in Metropolis algorithm)

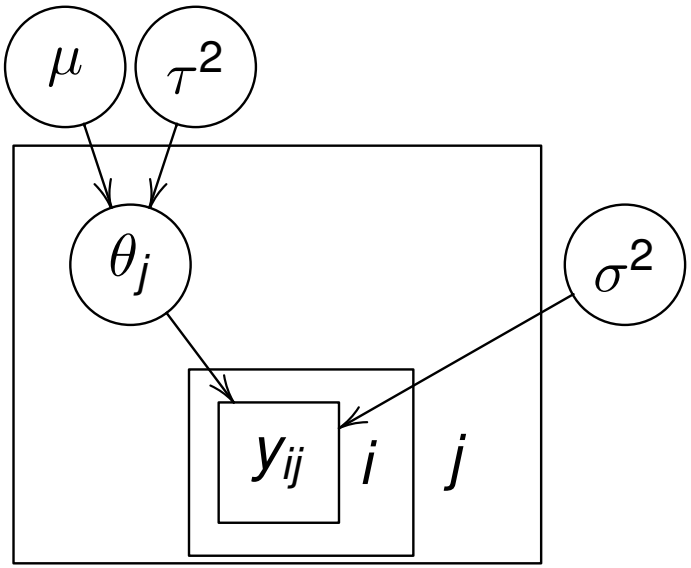
# Gibbs sampling

- When using conditionally conjugate (hyper)priors, the sampling from the conditional distributions is easy for wide range of models
  - e.g. hierarchical normal distribution model
  - WinBUGS/OpenBUGS/JAGS
- No algorithm parameters to tune
  (cf. proposal distribution in Metropolis algorithm)
- For not so easy conditional distributions, trivial to use e.g. grid, Metropolis-Hastings or slice sampling
- Several parameters can be updated in blocks (*blocking*, cf. Metropolis-Hastings)
- Gibbs sampling can be very slow if parameters are highly dependent in the posterior

# Gibbs for hierarchical normal distribution model

$$\theta_j|\mu,\tau^2 \sim \mathrm{N}(\mu,\tau^2)$$

$$y_{ij}|\theta_j \sim \mathrm{N}(\theta_j,\sigma^2)$$




- Simulate from conditional distributions (BDA3 Section 11.6):
  - $\theta_j|\mu,\sigma,\tau,y$
  - $\mu|\theta,\sigma,\tau,y$
  - $\sigma^2|\theta,\mu,\tau,y$
  - $\tau^2|\theta,\mu,\sigma,y$

# Metropolis algorithm

- Metropolis algorithm and its generalizations are basis for all MCMC methods
- Algorithm
  1. starting point $\theta^0$
  2. $t = 1, 2, \ldots$
     (a) pick a proposal $\theta^*$ 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 $\theta_a, \theta_b$

# Metropolis algorithm

- Metropolis algorithm and its generalizations are basis for all MCMC methods
- Algorithm
  1. starting point $\theta^0$
  2. $t = 1, 2, \ldots$
     (a) pick a proposal $\theta^*$ 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 $\theta_a, \theta_b$
     (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

- Metropolis algorithm and its generalizations are basis for all MCMC methods
- Algorithm
  1. starting point $\theta^0$
  2. $t = 1, 2, \ldots$
     (a) pick a proposal $\theta^*$ 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 $\theta_a, \theta_b$

# Metropolis algorithm

- Metropolis algorithm and its generalizations are basis for all MCMC methods
- Algorithm
  1. starting point $\theta^0$
  2. $t = 1, 2, \ldots$
     (a) pick a proposal $\theta^*$ 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 $\theta_a, \theta_b$
     (b) calculate acceptance ratio

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

     (c) set

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

  - instead of $p(\theta|y)$, unnormalized $q(\theta|y)$ can be used
  - step c is executed by generating a random number from U(0, 1)
  - rejection of a proposal increments the time $t$ also by one

# 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 $J_t(\theta^*|\theta^{t-1}) = \mathrm{N}(\theta^*|\theta^{t-1}, 0.8^2)$
- demo11_2.m

# Metropolis-Hastings algorithm

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

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

# Metropolis-Hastings algorithm

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

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

# Metropolis-Hastings algorithm

- More efficient proposal distributions
- e.g. Langevin-Hastings algorithm which uses gradient information to make proposals
  - more likely to propose values with higher density

# Metropolis-Hastings algorithm

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

# Metropolis-Hastings algorithm

- 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

- 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)

# Metropolis-Hastings algorithm

- 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)

# Metropolis-Hastings algorithm

- Update of parameters
  - jointly
  - blocked
  - single-component
- Order of single or blocked updates is free
  - same for all rounds
  - random
  - not all parameters updated each round

# 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 vs. Gibbs

- Metropolis
  - simplest to implement
- Gibbs
  - useful when also discrete parameters
  - maybe fast in specific cases with graphical models

# Advanced MCMC methods

- Chapter 12 presents some more advanced methods
  - Chapter 12 includes HMC/NUTS, which is one of the most efficient methods
  - many algorithms for special cases (like Gaussian processes) are not presented in the book

# Warm-up (burn-in) and convergence diagnostics

- How long it takes to forget the starting point $\theta^0$?
- Need to forget the starting point
  - $\rightarrow$ Warm-up (burn-in) = remove samples from the beginning of the chain
- When the chain has forgot the starting point it has converged
  - $\rightarrow$ convergence diagnostics

# MCMC samples are dependent

- Monte Carlo estimates still valid
- Estimation of Monte Carlo error is more difficult
  - time series analysis
  - thinning
- Evaluation of effective sample size
  - based on time series analysis

# Several chains

- Use of several chains make convergence diagnostics easier
  - start chains from different starting points – preferably overdispersed
  - use different pseudo random generator seed
- Compare samples from the different chains
- Remove samples 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

# Visual convergence diagnostics

- demo11_3.m
- Visual inspection works when a small number of quantities and may indicate where the problem is
- Visual inspection is hard when a large number of quantities

# Comparison within and between variances of the chains (PSRF)

- Examines mixing and stationarity of chains
- To examine stationarity chains are splitted to two parts
  - after splitting $m$ chains, each having $n$ samples
  - scalar samples $\psi_{ij} \quad (i = 1, \ldots, n; j = 1, \ldots, m)$

# Comparison within and between variances of the chains (PSRF)

- BDA3: *potential scale reduction factor* (PSRF)
  - compare means and variances of the chains
  - works best for quantities which are approximately normally distributed
    - good to transform variables, e.g., by taking logarithm of positive quantity

# Comparison within and between variances of the chains (PSRF)

- Between chains variance $B$

$$B = \frac{n}{m-1}\sum_{j=1}^{m}(\bar{\psi}_{.j}-\bar{\psi}_{..})^2, \text{ where } \bar{\psi}_{.j} = \frac{1}{n}\sum_{i=1}^{n}\psi_{ij}, \bar{\psi}_{..} = \frac{1}{m}\sum_{j=1}^{m}\bar{\psi}_{.j}$$

  - $B/n$ is variance of the means of the chains
- Within chains variance $W$

$$W = \frac{1}{m}\sum_{j=1}^{m}s_j^2, \text{ where } s_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}(\psi_{ij}-\bar{\psi}_{.j})^2$$

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

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

# Comparison within and between variances of the chains (PSRF)

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

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

  - 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 $n \to \infty$, $\quad \text{E}(W) \to \text{var}(\psi|y)$
- As $\widehat{\text{var}}^{+}(\psi|y)$ overestimates and $W$ underestimates, compute

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

# Comparison within and between variances of the chains (PSRF)

- Potential scale reduction factor

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

  - estimates how much the scale of $\psi$ could reduce if $n \to \infty$
  - $R \to 1$, when $n \to \infty$
  - if $R$ is big, keep sampling
  - big is, e.g., $R > 1.1$
- demo11_4.m

# Potential scale reduction function (PSRF)

- If $R$ close to 1, it is still possible that chains have not converged
  - if starting points were not overdispersed
  - distribution far from normal
  - just by chance when $n$ is finite

# Convergence diagnostics

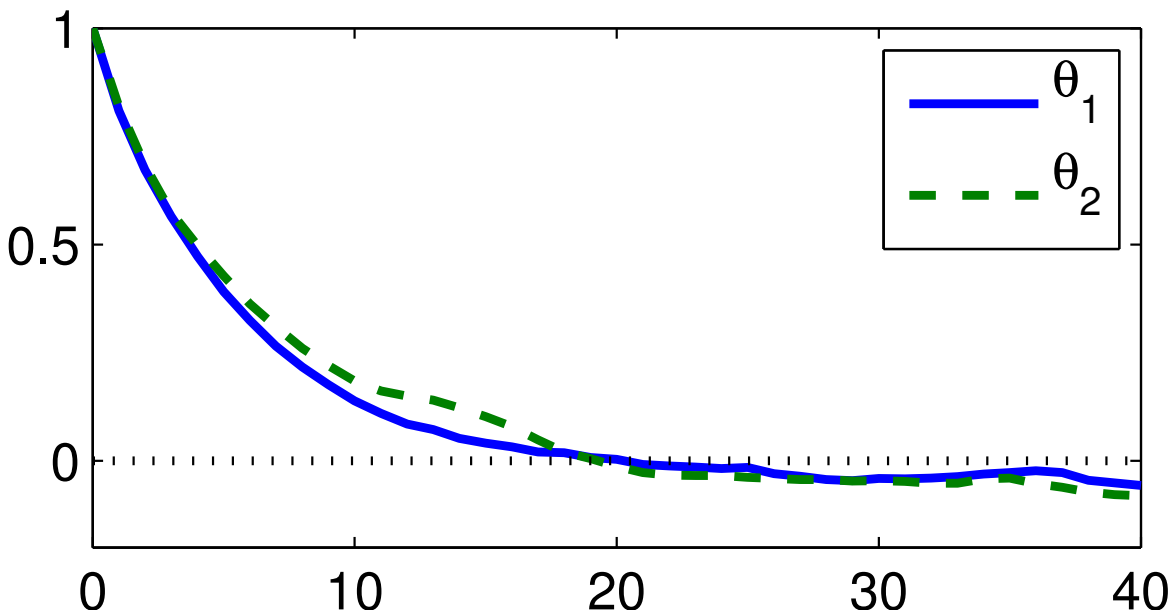
- After how many iterations can we check for convergence
  - it's not possible to no this beforehand
  - Stan uses NUTS which is an efficient sampler and default is to use 4 chains, get 1000 draws from each for adaptation and warmup, 1000 more draws from each and estimate $\hat{R}$ for each parameter

# Problematic distributions

- Nonlinear dependencies
- Funnels
- Multimodal
- Long-tailed with undefined variance and mean

# Time series analysis

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



# Time series analysis

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

$$\text{Var}[\bar{\theta}] = \frac{\sigma_\theta^2}{L/\tau}$$

  where $\tau$ is sum of autocorrelations
  - $\tau$ describes how many dependent samples correspond to one independent sample
  - in BDA3 $L = nm$
  - $n_{\text{eff}} = nm/\tau$
  - BDA3 focuses on $n_{\text{eff}}$ and not the Monte Carlo error directly

- Estimation of the autocorrelation using several chains

$$\hat{\rho}_t = 1 - \frac{V_t}{2\widehat{\text{var}}^+}$$

where $V_t$ is variogram

$$V_t = \frac{1}{m(n-t)} \sum_{j=1}^{m} \sum_{i=t+1}^{n} (\psi_{i,j} - \psi_{i-t,j})^2$$

- Compared to usual method which computes the autocorrelation from a single chain, this estimate has smaller variance, ad $\widehat{\text{var}}^+$ is estimated using several chains

# Time series analysis

- Estimation of $\tau$

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

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

  - empirical autocorrelation function is noisy and thus estimate of $\tau$ is noisy
  - noise is larger for longer lags (less observations)
  - less noisy estimate is obtained by truncating

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

- As $\tau$ is estimated from a finite number of samples, it's expectation is overoptimistic
  - if $\tau > mn/20$ then the estimate is unreliable

# Geyer's adaptive window estimator

- Truncation can be decided adaptively by taking into account some properties of Markov chains
  - for stationary, irreducible, recurrent Markov chain
  - let $\Gamma_m = \rho_{2m} + \rho_{2m+1}$, which is sum of two consequent autocorrelations
  - $\Gamma_m$ 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

# Time series analysis

- Effective number of samples

$$n_{\text{eff}} \approx L/\tau$$

# Thinning

- Not necessary, but used often
  - to save disk space
  - makes post-sampling computations faster
  - makes estimation of Monte Carlo error easier
- Save every $k$th sample
  - if $k > m$, where $m$ from Geyer's method, then samples almost independent
  - information is lost as $m > \tau$
  - demo11_5.m