

Markov chain Monte Carlo II

Models for Socio-Environmental Data

Chris Che-Castaldo, Mary B. Collins, and N. Thompson Hobbs

June 1, 2018



The MCMC algorithm

- ▶ Some intuition
- ▶ Accept-reject sampling with Metropolis algorithm
- ▶ Introduction to full-conditional distributions
- ▶ Gibbs sampling
- ▶ Metropolis-Hastings algorithm
- ▶ Implementing accept-reject sampling

Implementing MCMC for multiple parameters and latent quantities

- ▶ Write an expression for the posterior and joint distribution using a DAG as a guide. Always.
- ▶ If you are using MCMC software (e.g. JAGS) use the expression for the posterior and joint distribution as template for writing code.
- ▶ If you are writing your own MCMC sampler:
 - ▶ Decompose the expression of the multivariate joint distribution into a series of univariate distributions called *full-conditional distributions*.
 - ▶ Choose a sampling method for each full-conditional distribution.
 - ▶ Cycle through each unobserved quantity, sampling from its full-conditional distribution, treating the others as if they were known and constant.
 - ▶ The accumulated samples approximate the marginal posterior distribution of each unobserved quantity.
 - ▶ Note that this takes a complex, multivariate problem and turns it into a series of simple, univariate problems that we solve, as in the example above, one at a time.

Choosing a sampling method

1. Accept-reject:
 - 1.1 Metropolis
 - 1.2 Metropolis-Hastings
2. Gibbs: accepts all proposals because they are especially well chosen.

When is accept-reject update mandatory?

We need to use Metropolis, Metropolis-Hastings or some other accept reject methods whenever

1. A conjugate relationship does not exist for the full-conditional distribution of a parameter, for example, for the shape parameter in the gamma distribution.
2. The deterministic model is non-linear and can't be linearized by transformation.

When is a model linear?

- ▶ A model is linear if it can be written as the sum of products of coefficients and predictor variables, i.e.

$\mu_i = \beta_0 + \beta_1 x_{1,i} + \dots + \beta_n x_{n,i}$ or in matrix form $\mu_i = \mathbf{x}_i \boldsymbol{\beta}$. We can take powers and products of the x and the model remains linear. We often transform models to linearize them using link functions (i.e., log, logit, probit).

- ▶ A model is non-linear if it cannot be written this way.

Metropolis Updates

$$\begin{aligned} [\theta^{*k+1} | y] &= \frac{\overbrace{[y | \theta^{*k+1}]}^{\text{likelihood}} \overbrace{[\theta^{*k+1}]}^{\text{prior}}}{\int [y | \theta] [\theta] d\theta} \\ [\theta^k | y] &= \frac{\overbrace{[y | \theta^k]}^{\text{likelihood}} \overbrace{[\theta^k]}^{\text{prior}}}{\int [y | \theta] [\theta] d\theta} \\ R &= \frac{[\theta^{*k+1} | y]}{[\theta^k | y]} \end{aligned}$$

Metropolis Updates

$$\begin{aligned} [\theta^{*k+1} | y] &= \frac{\overbrace{[y | \theta^{*k+1}]}^{\text{likelihood}} \overbrace{[\theta^{*k+1}]}^{\text{prior}}}{\cancel{f[y|\theta]} \cancel{[\theta]} d\theta} \\ [\theta^k | y] &= \frac{\overbrace{[y | \theta^k]}^{\text{likelihood}} \overbrace{[\theta^k]}^{\text{prior}}}{\cancel{f[y|\theta]} \cancel{[\theta]} d\theta} \\ R &= \frac{[\theta^{*k+1} | y]}{[\theta^k | y]} \end{aligned}$$

Proposal distributions

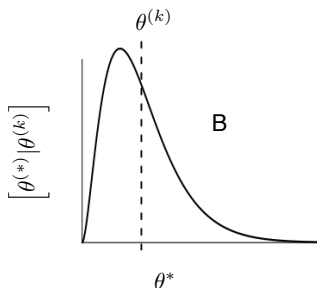
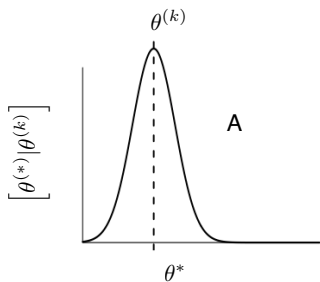
- ▶ Independent chains have proposal distributions that do not depend on the current value (θ^k) in the chain. This is what we used in the *Chytrid* example.
- ▶ Dependent chains, as you might expect, have proposal distributions that *do* depend on the current value of the chain (θ^k). In this case we draw from

$$[\theta^{*k+1} | \theta^k, \sigma] \quad (1)$$

where σ is a *tuning* parameter that we specify to obtain an acceptance rate of about 40%. Note that my notation and notation of others simplifies this distribution to $[\theta^{*k+1} | \theta^k]$. The σ is implicit because it is a constant, not a random variable. We specify σ .

- ▶ Why are dependent chains usually more efficient than independent chains?

Proposal distributions for dependent chains



Metropolis-Hastings updates

- ▶ Metropolis updates require symmetric proposal distributions (e.g., uniform, normal).
- ▶ Metropolis-Hastings updates allow use of asymmetric (e.g., beta, gamma, lognormal).

Definition of symmetry

A proposal distribution is symmetric if and only if

$$[\theta^{*k+1} | \theta^k] = [\theta^k | \theta^{*k+1}]. \quad (2)$$

Normal and uniform are symmetric. Gamma, beta, lognormal are not.

Illustrating with code

```
#symmetric example
sigma=1
x = .8
z=rnorm(1,mean=x,sd=sigma);z
#[z|x]
dnorm(z,mean=x,sd=sigma)
#[x|z]
dnorm(x,mean=z,sd=sigma)
#asymmetric example
sigma=1
x = .8
a.x=x^2/sigma^2; b.x=x/sigma^2
z=rgamma(1,shape=a.x,rate=b.x);z
a.z=z^2/sigma^2; b.z=z/sigma^2
#[z|x]
dgamma(z,shape=a.x,rate=b.x)
#[x|z]
dgamma(x,shape=a.z,rate=b.z)
```

Metropolis-Hastings updates

Metropolis R:

$$R = \frac{[\boldsymbol{\theta}^{*k+1} | y]}{[\boldsymbol{\theta}^k | y]} \quad (3)$$

Metropolis-Hastings R:

$$R = \frac{[\boldsymbol{\theta}^{*k+1} | y]}{[\boldsymbol{\theta}^k | y]} \frac{\overbrace{[\boldsymbol{\theta}^k | \boldsymbol{\theta}^{*k+1}]}^{\text{Proposal distribution}}}{\underbrace{[\boldsymbol{\theta}^{*k+1} | \boldsymbol{\theta}^k]}_{\text{Proposal distribution}}}, \quad (4)$$

which is the same as:

$$R = \frac{\overbrace{[y | \boldsymbol{\theta}^{*k+1}]}^{\text{Likelihood}} \overbrace{[\boldsymbol{\theta}^{*k+1}]}^{\text{Prior}} \overbrace{[\boldsymbol{\theta}^k | \boldsymbol{\theta}^{*k+1}]}^{\text{Proposal distribution}}}{\underbrace{[y | \boldsymbol{\theta}^k]}_{\text{Likelihood}} \underbrace{[\boldsymbol{\theta}^k]}_{\text{Prior}} \underbrace{[\boldsymbol{\theta}^{*k+1} | \boldsymbol{\theta}^k]}_{\text{Proposal distribution}}} \quad (5)$$

Example using beta proposal distribution

1. Current value of parameter, $\theta^k = .42$, tuning parameter set at $\sigma = .10$
2. Make a draw from $\theta^{*k+1} \sim \text{beta}(m(.42, .10))$, where m is moment matching function.

3. Calculate $R = \frac{[\theta^{*k+1}|y] \overbrace{[.42|m(\theta^{*k+1}, .10)]}^{\text{beta}}}{[\theta^k|y] \underbrace{[\theta^{*k+1}|m(.42, .10)]}_{\text{beta}}}$.

4. Choose proposed or current value based on R as we did with Metropolis.

MCMC

- ▶ Methods based on the Markov chain Monte Carlo algorithm allow us to approximate marginal posterior distributions of unobserved quantities without analytical integration.
- ▶ This makes it possible to estimate models that have many parameters, have multiple sources of uncertainty, and include latent quantities.
- ▶ We will learn a tool, JAGS, that simplifies the implementation of MCMC methods.
- ▶ Will will put this tool to use in building models that include nested levels in space, errors in the observations, differences among groups and processes that unfold over time.