University of Nevada, Reno

Lecture 14: Markov chain Monte Carlo II

Perry Williams, PhD

NRES 779

Bayesian Hierarchical Modeling in Natural Resources

Learning Objectives: 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

- Write the posterior and joint distribution.
- If you are using MCMC software (e.g., JAGS) use expression for the posterior and joint distribution as a template for code.
- If you are writing your own MCMC sampler:
  - Decompose the expression of the multivariate joint distribution into *full-conditional distributions*
  - Choose a sampling method.
  - 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.
  - A complex, multivariate problem is turned into a series of simple, univariate problems.

Choosing a Sampling Method

- Accept-reject:
    - Metropolis
    - Metropolis-Hastings
- 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:

- A conjugate relationship does not exist for the full-conditional distribution of a parameter, for example, for the shape parameter in the gamma distribution.
- The deterministic model is non-linear, which almost always means a conjugate doesn't exist for its parameters.

When is a Model Linear?

- A model is linear if it can be written as the sum of products of coefficients and predictor variables
  - $\mu_i = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_{p,i}$

  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). These are called *generalized linear models*.
- A model is non-linear if it cannot be written this way.

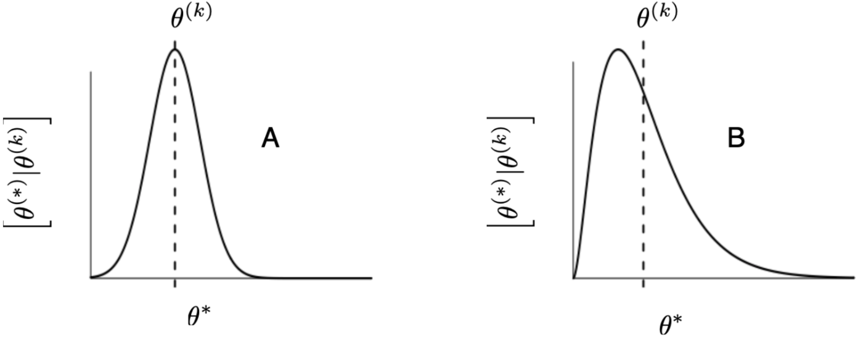
Proposal Distributions

- Independent chains have proposal distributions that do not depend on the current value (θ^k) in the chain.
- 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

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

  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.
- 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 distributions (e.g., beta, gamma, lognormal).

Definition of Symmetry

A proposal distribution is symmetric if and only if

$$[\theta^{*,k+1}|\theta^{k}] = [[\theta^{k+1}|\theta^{*,k+1}]$$

. 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:

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

Metropolis-Hastings R:

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

which is the same as:

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

Example Using Beta Proposal Distribution

1. Current value of parameter, $\theta^k = .42$, tuning parameter set at $\boldsymbol{\sigma} = .10$
2. Make a draw from $\theta{*}^{k+1} \sim$ beta$(m(.42, .10))$, where m is moment matching function.
3. Calculate $R = \dfrac{\overbrace{[y \mid \boldsymbol{\theta}^{*k+1}]}^{\text{Likelihood}}\overbrace{[\boldsymbol{\theta}^{*k+1}]}^{\text{Prior}}\overbrace{[.42| \mid m(\boldsymbol{\theta}^{*k+1}, .10)]}^{\text{beta proposal}}}{\underbrace{[y \mid \boldsymbol{\theta}^{k}]}_{\text{Likelihood}}\underbrace{[\boldsymbol{\theta}^{k}]}_{\text{Prior}}\underbrace{[\boldsymbol{\theta}^{*k+1} \mid m(.42, .10)]}_{\text{beta proposal}}}.$
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 (but limits) the implementation of MCMC.
- 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.