Markov chain Monte Carlo I

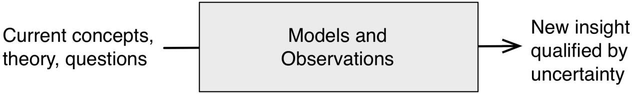
Models for Socio-Environmental Data

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

May 23, 2019

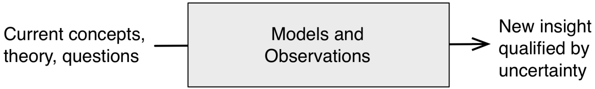
SESYNC

What is this course about?

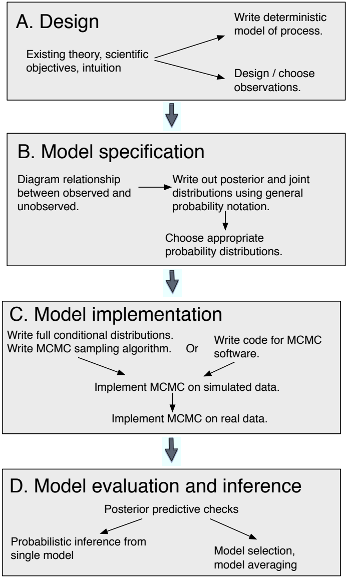


You can understand it.

- Rules of probability
  - Conditioning and independence
    - Law of total probability
    - Factoring joint probabilities
- Distribution theory
- Markov chain Monte Carlo



The Bayesian method



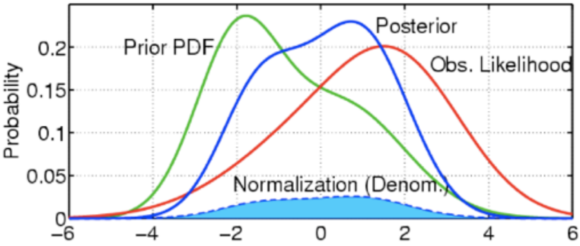
The MCMC algorithm

- Why MCMC?
- Some intuition about how it works for a single parameter model
- MCMC for multiple parameter models
  - Full-conditional distributions (today)
  - Gibbs sampling (today and lab next week)
  - Metropolis-Hastings algorithm (Thursday)
  - MCMC software (JAGS, week after next)

MCMC learning outcomes

1. Develop a big picture understanding of how MCMC allows us to approximate the marginal posterior distributions of parameters and latent quantities.
2. Understand and be able to code a simple MCMC algorithm.
3. Appreciate the different methods that can be used within MCMC algorithms to make draws from the posterior distribution.
   3.1 Metropolis
   3.2 Metropolis-Hastings
   3.3 Gibbs
4. Understand concepts of burn-in and convergence.
5. Understand and be able to write full-conditional distributions.

Remember the marginal distribution of the data



We have simple solutions for the posterior for simple models:

$$[\phi|y] = \text{beta}\left(\underbrace{\overbrace{\alpha}^{\text{The prior } \alpha} + y}_{\text{The new } \alpha}, \underbrace{\overbrace{\beta}^{\text{The prior}\beta} + n - y}_{\text{The new } \beta}\right)$$

Problems of high dimension do not have simple solutions:

$$[\theta_1, \theta_2, \theta_3, \theta_4, \mathbf{z} \mid \mathbf{y}, \mathbf{u}] = \frac{\prod_{i=1}^{n} [y_i|\boldsymbol{\theta}_1 z_i][u_i|\boldsymbol{\theta}_2, z_i][z_i|\boldsymbol{\theta}_3, \boldsymbol{\theta}_4][\boldsymbol{\theta}_1][\boldsymbol{\theta}_2][\boldsymbol{\theta}_3][\boldsymbol{\theta}_4]}{\int \ldots \int \prod_{i=1}^{n} [y_i|\boldsymbol{\theta}_1 z_i][u_i|\boldsymbol{\theta}_2, z_i][z_i|\boldsymbol{\theta}_3, \boldsymbol{\theta}_4][\boldsymbol{\theta}_1][\boldsymbol{\theta}_2][\boldsymbol{\theta}_3][\boldsymbol{\theta}_4]\, dz_i\, d\boldsymbol{\theta}_1\, d\boldsymbol{\theta}_2\, d\boldsymbol{\theta}_3\, d\boldsymbol{\theta}_4}$$

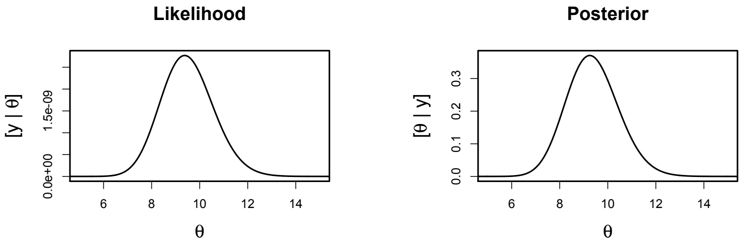
What we are doing in MCMC?

Recall that the posterior distribution is proportional to the joint:

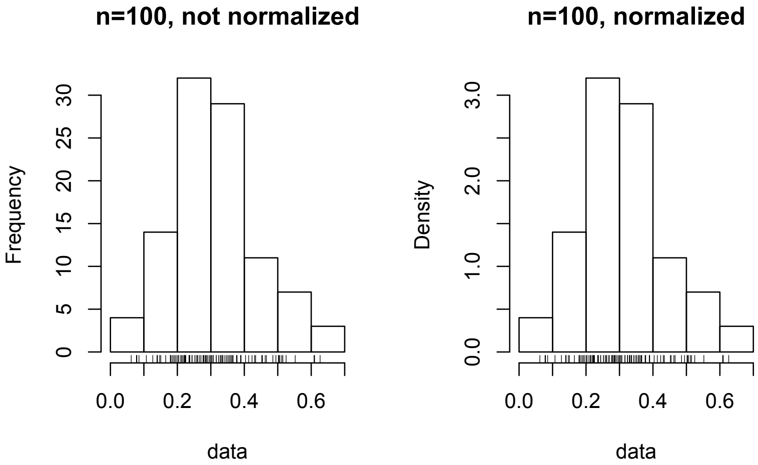
$$[\boldsymbol{\theta}|y] \propto [y|\boldsymbol{\theta}][\boldsymbol{\theta}], \tag{1}$$

because the marginal distribution of the data $\int [y|\boldsymbol{\theta}][\boldsymbol{\theta}]d\boldsymbol{\theta}$ is a constant after the data have been observed.

What we are doing in MCMC?



What we are doing in MCMC?

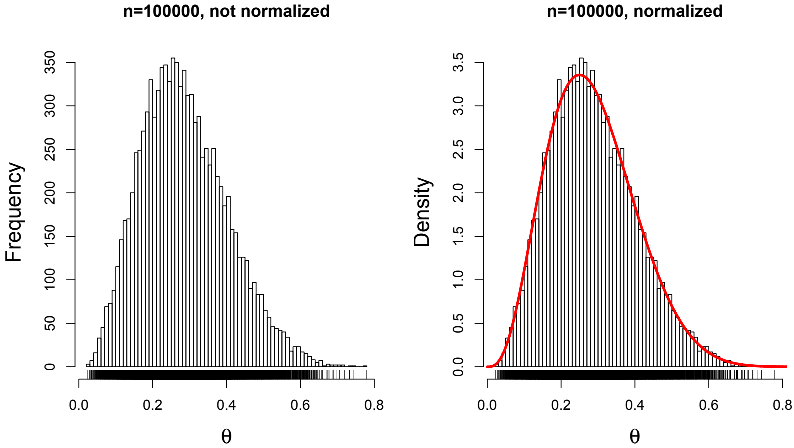


What are we doing in MCMC?

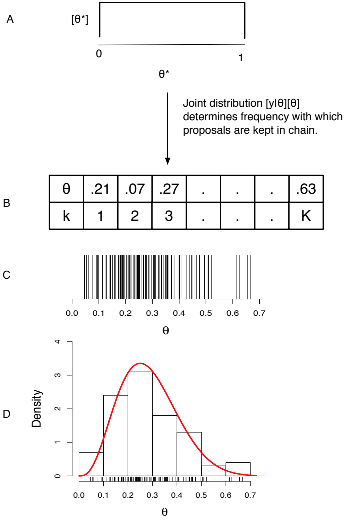
- The posterior distribution is unknown, but the likelihood is known as a likelihood profile and we know the priors.
- We want to accumulate many, many values that represent random samples proportionate to their density in the *marginal* posterior distribution.
- MCMC generates these samples using the likelihood and the priors to decide which samples to keep and which to throw away.
- We can then use these samples to calculate statistics describing the distribution: means, medians, variances, credible intervals etc.

What are we doing in MCMC?

The marginal posterior distribution of each unobserved quantity is approximated by samples accumulated in the chain.



What are we doing in MCMC?



Metropolis updates

We keep the more probable members of the posterior distribution by comparing a proposal with the current value in the chain.

$$\begin{array}{ccc}
k & 1 & 2 \\
\text{Proposal}\,\boldsymbol{\theta}^{*\,k+1} & & \boldsymbol{\theta}^{*\,2} \\
\text{Test} & & P(\boldsymbol{\theta}^{*\,2}) > P\left(\boldsymbol{\theta}^{1}\right) \\
\text{Chain}(\boldsymbol{\theta}^{k}) & \boldsymbol{\theta}^{1} & \boldsymbol{\theta}^{2} = \boldsymbol{\theta}^{*\,2}
\end{array}$$

Metropolis updates

We keep the more probable members of the posterior distribution by comparing a proposal with the current value in the chain.

$$\begin{array}{cccc}
k & 1 & 2 & 3 \\
\text{Proposal}\,\boldsymbol{\theta}^{*\,k+1} & & \boldsymbol{\theta}^{*\,2} & \boldsymbol{\theta}^{*\,3} \\
\text{Test} & & P(\boldsymbol{\theta}^{*\,2}) > P\left(\boldsymbol{\theta}^{1}\right) & P(\boldsymbol{\theta}^{2}) > P\left(\boldsymbol{\theta}^{*\,3}\right) \\
\text{Chain}(\boldsymbol{\theta}^{k}) & \boldsymbol{\theta}^{1} & \boldsymbol{\theta}^{2} = \boldsymbol{\theta}^{*\,2} & \boldsymbol{\theta}^{3} = \boldsymbol{\theta}^{2}
\end{array}$$

Metropolis updates

We keep the more probable members of the posterior distribution by comparing a proposal with the current value in the chain.

k	1	2	3	4				K
Proposal $\theta^{*\,k+1}$		$\theta^{*\,2}$	$\theta^{*\,3}$	$\theta^{*\,4}$	.	.	.	.
Test		$P(\theta^{*\,2}) > P\left(\theta^{1}\right)$	$P(\theta^{2}) > P\left(\theta^{*\,3}\right)$	$P(\theta^{3}) > P\left(\theta^{*\,4}\right)$	.	.	.	.
Chain(θ^{k})	θ^{1}	$\theta^{2} = \theta^{*\,2}$	$\theta^{3} = \theta^{2}$	$\theta_{4} = \theta_{3}$				

Metropolis updates

$$[\boldsymbol{\theta}^{*k+1}|y] = \frac{\overbrace{[y|\boldsymbol{\theta}^{*k+1}]}^{\text{likelihood}}\overbrace{[\boldsymbol{\theta}^{*k+1}]}^{\text{prior}}}{\int [y|\boldsymbol{\theta}][\boldsymbol{\theta}]d\boldsymbol{\theta}}$$

$$[\boldsymbol{\theta}^{k}|y] = \frac{\overbrace{[y|\boldsymbol{\theta}^{k}]}^{\text{likelihood}}\overbrace{[\boldsymbol{\theta}^{k}]}^{\text{prior}}}{\int [y|\boldsymbol{\theta}][\boldsymbol{\theta}]d\boldsymbol{\theta}}$$

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

The cunning idea behind Metropolis updates

$$[\boldsymbol{\theta}^{*k+1}|y] = \frac{\overbrace{[y|\boldsymbol{\theta}^{*k+1}]}^{\text{likelihood}}\overbrace{[\boldsymbol{\theta}^{*k+1}]}^{\text{prior}}}{\cancel{\int [y|\boldsymbol{\theta}][\boldsymbol{\theta}]d\boldsymbol{\theta}}}$$

$$[\boldsymbol{\theta}^{k}|y] = \frac{\overbrace{[y|\boldsymbol{\theta}^{k}]}^{\text{likelihood}}\overbrace{[\boldsymbol{\theta}^{k}]}^{\text{prior}}}{\cancel{\int [y|\boldsymbol{\theta}][\boldsymbol{\theta}]d\boldsymbol{\theta}}}$$

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

When do we keep the proposal?

$$P_R = \min(1, R)$$

Keep θ^{*k+1} as the next value in the chain with probability P_R and keep θ^k with probability $1 - P_R$.

When do we keep the proposal?

1. Calculate R based on likelihoods and priors.
2. Draw a random number, U from uniform distribution 0,1 If $R > U$, we keep the proposal θ^{*k+1} as the next value in the chain.
3. Otherwise, we retain θ^k as the next value.

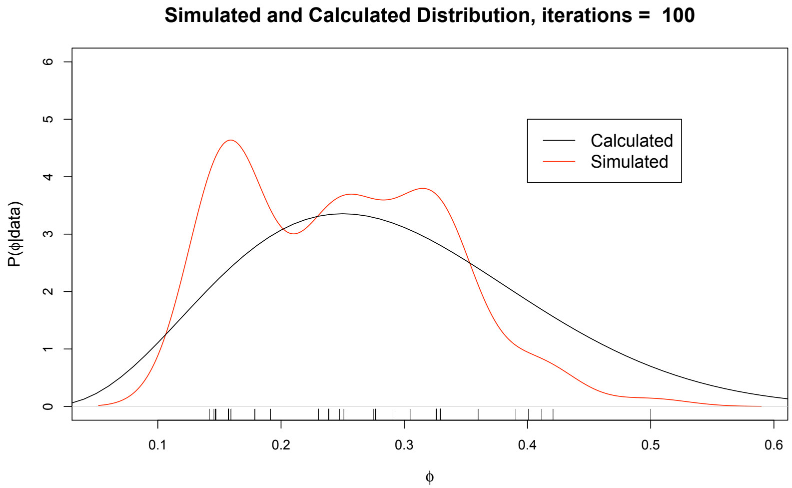
A simple example for one parameter

- Tawni is interested in estimating the prevalence of bacterial kidney disease in a population of trout in Colorado.
- She is a bit lazy, so she only samples 12 fish, 3 of which have the disease.
- How could she calculate the parameters of the posterior distribution of prevalence on the back of a cocktail napkin?

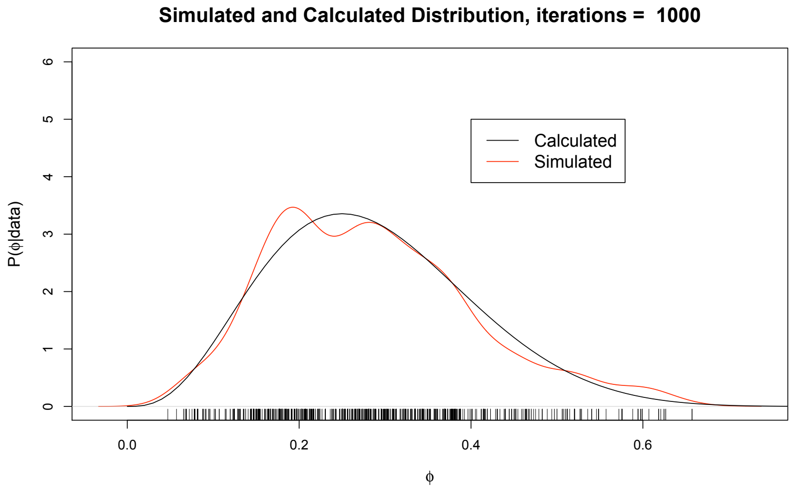
The model

$$[\phi|y] \propto \text{binomial}(y|n,\phi)\text{beta}(\phi|1,1)$$

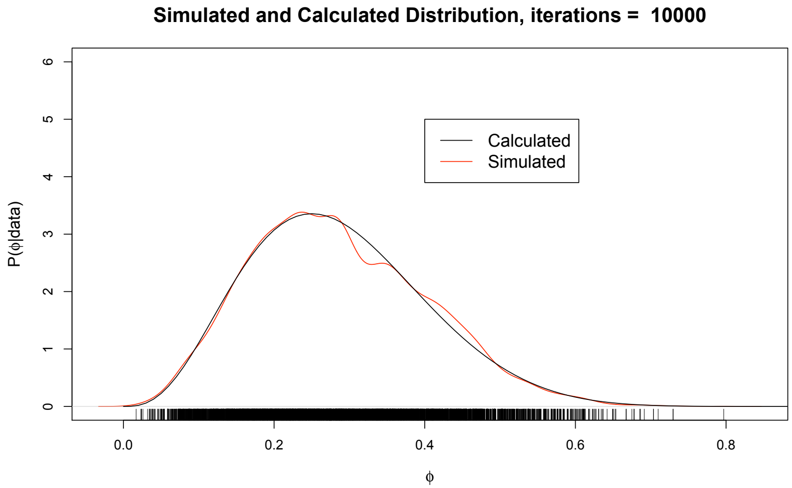
Sampling from the posterior



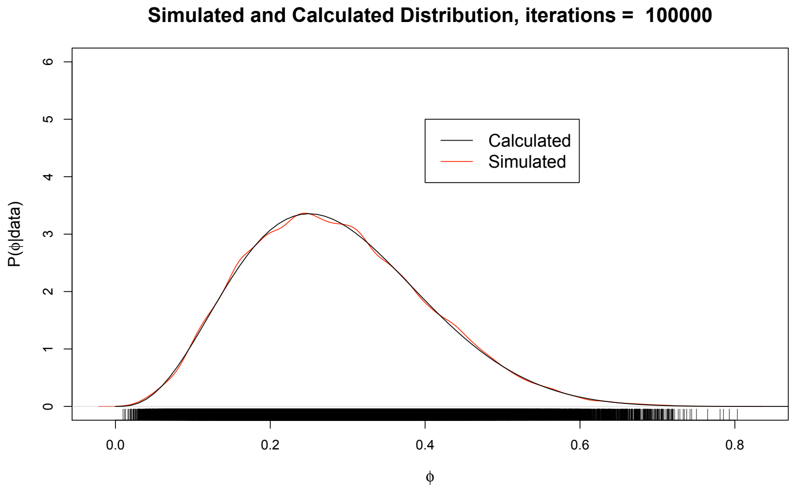
Sampling from the posterior



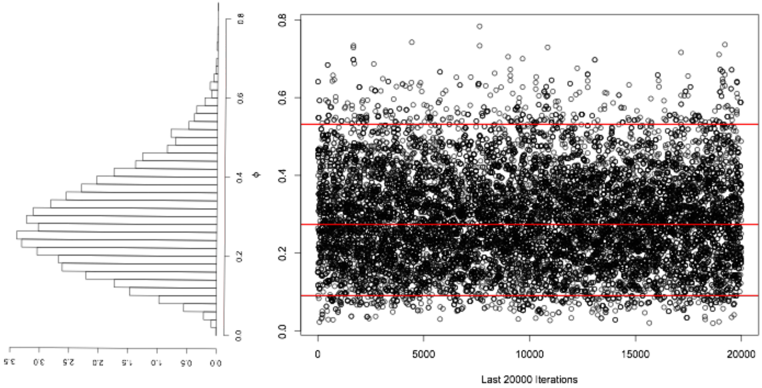
Sampling from the posterior



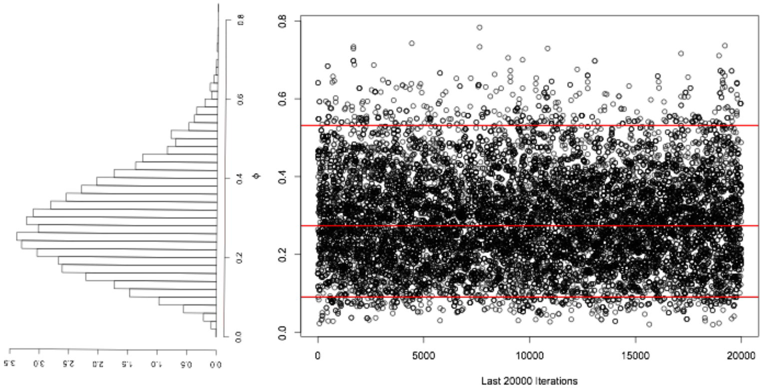
Sampling from the posterior



Sampling from the posterior



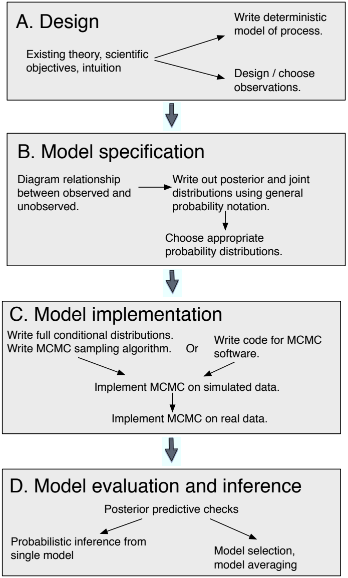
Sampling from the posterior



The chain has *converged* when adding more samples does not change the shape of the posterior distribution. We throw away samples that are accumulated before convergence (burn-in).

Start Tuesday Lecture here with some review.

The Bayesian method



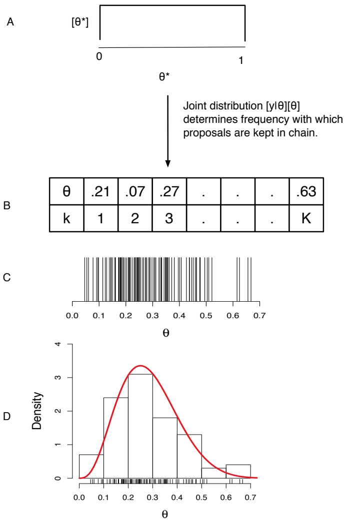
The MCMC algorithm

- Why MCMC?
- Some intuition about how it works for a single parameter model
- MCMC for multiple parameter models
  - Full-conditional distributions (today)
  - Gibbs sampling (today and lab next week)
  - Metropolis-Hastings algorithm (Thursday)
  - MCMC software (JAGS, week after next)

MCMC learning outcomes

1. Develop a big picture understanding of how MCMC allows us to approximate the marginal posterior distributions of parameters and latent quantities.
2. Understand and be able to code a simple MCMC algorithm.
3. Appreciate the different methods that can be used within MCMC algorithms to make draws from the posterior distribution.
   - 3.1 Metropolis
   - 3.2 Metropolis-Hastings
   - 3.3 Gibbs
4. Understand concepts of burn-in and convergence.
5. Understand and be able to write full-conditional distributions.

What are we doing in MCMC?



Metropolis updates

$$[\boldsymbol{\theta}^{*k+1}|y] = \frac{\overbrace{[y|\boldsymbol{\theta}^{*k+1}]}^{\text{likelihood}}\overbrace{[\boldsymbol{\theta}^{*k+1}]}^{\text{prior}}}{\int [y|\boldsymbol{\theta}][\boldsymbol{\theta}]d\boldsymbol{\theta}}$$

$$[\boldsymbol{\theta}^{k}|y] = \frac{\overbrace{[y|\boldsymbol{\theta}^{k}]}^{\text{likelihood}}\overbrace{[\boldsymbol{\theta}^{k}]}^{\text{prior}}}{\int [y|\boldsymbol{\theta}][\boldsymbol{\theta}]d\boldsymbol{\theta}}$$

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

The cunning idea behind Metropolis updates

$$[\boldsymbol{\theta}^{*k+1}|y] = \frac{\overbrace{[y|\boldsymbol{\theta}^{*k+1}]}^{\text{likelihood}}\overbrace{[\boldsymbol{\theta}^{*k+1}]}^{\text{prior}}}{\cancel{\int [y|\boldsymbol{\theta}][\boldsymbol{\theta}]d\boldsymbol{\theta}}}$$

$$[\boldsymbol{\theta}^{k}|y] = \frac{\overbrace{[y|\boldsymbol{\theta}^{k}]}^{\text{likelihood}}\overbrace{[\boldsymbol{\theta}^{k}]}^{\text{prior}}}{\cancel{\int [y|\boldsymbol{\theta}][\boldsymbol{\theta}]d\boldsymbol{\theta}}}$$

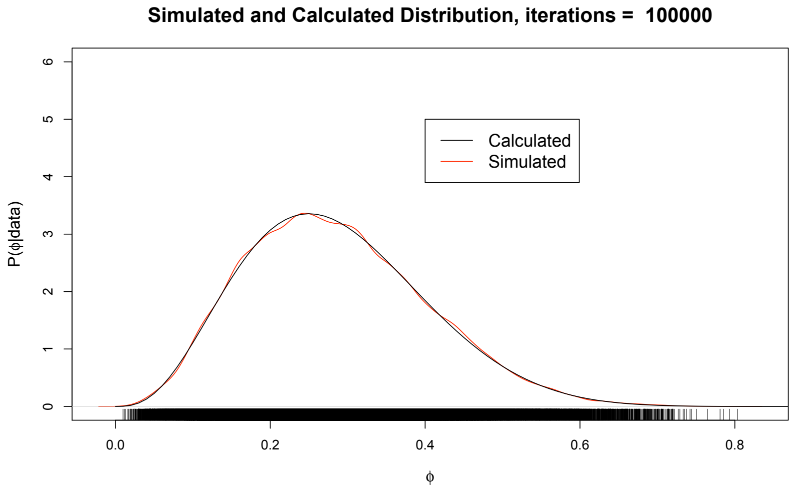
$$R = \frac{[\boldsymbol{\theta}^{*k+1}|y]}{[\boldsymbol{\theta}^{k}|y]}$$

When do we keep the proposal?

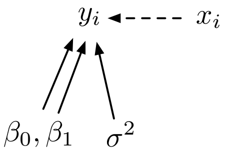
$$P_R = \min(1, R)$$

Keep θ^{*k+1} as the next value in the chain with probability P_R and keep θ^k with probability $1 - P_R$.

Sampling from the posterior



Intuition for MCMC for multi-parameter models



$$g(\beta_0, \beta_1, x_i) = \beta_0 + \beta_1 x_i$$

$$[\beta_0, \beta_1, \sigma^2 \mid y_i] \propto [\beta_0, \beta_1, \sigma^2, y_i]$$

factoring rhs using DAG:

$$[\beta_0, \beta_1, \sigma^2 \mid y_i] \propto [y_i \mid g(\beta_0, \beta_1, x_i), \sigma^2][\beta_0], [\beta_1][\sigma^2]$$

joint for all data :

$$[\beta_0, \beta_1, \sigma^2 \mid \boldsymbol{y}] \propto \prod_{i=1}^{n} [y_i \mid g(\beta_0, \beta_1, x_i), \sigma^2][\beta_0][\beta_1][\sigma^2]$$

choose specific distributions:

$$\begin{aligned}[\beta_0, \beta_1, \sigma^2 \mid \boldsymbol{y}] \propto & \prod_{i=1}^{n} \text{normal}(y_i \mid g(\beta_0, \beta_1, x_i), \sigma^2) \\ & \times \text{normal}(\beta_0 \mid 0, 10000)\text{normal}(\beta_1 \mid 0, 10000) \\ & \times \text{uniform}(\sigma^2 \mid 0, 500)\end{aligned}$$

Intuition for MCMC for multi-parameter models

$$\begin{aligned}[\beta_0, \beta_1, \sigma^2 \mid \mathbf{y}] &\propto \prod_{i=1}^{n} \text{normal}(y_i | g(\beta_0, \beta_1, x_i), \sigma^2) \\ &\times \text{normal}(\beta_0 \mid 0, 10000)\text{normal}(\beta_1 \mid 0, 10000)\text{uniform}(\sigma^2 \mid 0, 10\end{aligned}$$

1. Set initial values for $\beta_0, \beta_1, \sigma^2$
2. Assume β_1, σ^2 are known and constant. Make a draw for β_0. Store the draw.
3. Assume β_0, σ^2 are known and constant. Make a draw for β_1. Store the draw.
4. Assume β_0, β_1 are known and constant. Make a draw for σ^2. Store the draw.
5. Do this many times. The stored values for each parameter approximate its marginal posterior distribution after convergence.

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 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.

Definition of full-conditional distribution

Let $\boldsymbol{\theta}$ be a vector of length k containing all of the unobserved quantities we seek to understand. Let $\boldsymbol{\theta}_{-j}$ be a vector of length $k-1$ that contains all of the unobserved quantities *except* θ_j. The full-conditional distribution of θ_j is

$$[\theta_j | y, \boldsymbol{\theta}_{-j}],$$

which we notate as

$$[\theta_j | \cdot].$$

It is the posterior distribution of θ_j conditional on all of the other parameters and the data, which we assume are *known*.

Writing full-conditional distributions

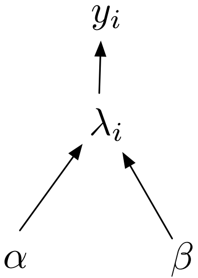
- You will have one full-conditional for each unobserved quantity in the posterior.
- For each unobserved quantity, write the distributions where it appears.
- Ignore the other distributions.
- Simple.

Example

- Clark 2003 considered the problem of modeling fecundity of spotted owls and the implication of individual variation in fecundity for population growth rate.
- Data were number of offspring produced by per pair of owls with sample size $n = 197$.

Clark, J. S. 2003. Uncertainty and variability in demography and population growth: A hierarchical approach. Ecology 84:1370-1381.

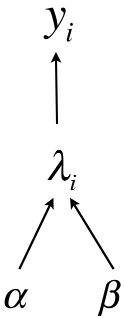
Example



$$
\begin{aligned}
[\boldsymbol{\lambda}, \alpha, \beta | \mathbf{y}] &\propto \prod_{i=1}^{n} \text{Poisson}\,(y_i | \lambda_i)\, \text{gamma}\,(\lambda_i | \alpha, \beta) \\
&\times \text{gamma}\,(\alpha | .001, .001)\, \text{gamma}\,(\beta | .001, .001)
\end{aligned}
$$

Full-conditionals

$$[\boldsymbol{\lambda}, \alpha, \beta | \mathbf{y}] \propto \prod_{i=1}^{n} \text{Poisson}\left(y_i | \lambda_i\right) \text{gamma}\left(\lambda_i | \alpha, \beta\right) \text{gamma}\left(\beta | .001, .001\right) \ \text{gamma}\left(\alpha | .001, .001\right)$$



We use the multivariate joint distribution to find univariate full-conditional distributions for all unobserved quantities.

How many full conditionals are there?

Writing full-conditional distributions

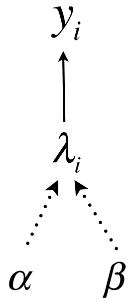
- You will have one full-conditional for each unobserved quantity in the posterior.
- For each unobserved quantity, write the distributions (including products) where it appears.
- Ignore the other distributions.
- Simple.

Full-conditional for each λ_i

$$[\boldsymbol{\lambda}, \alpha, \beta | \mathbf{y}] \propto \prod_{i=1}^{n} \boxed{\text{Poisson}\,(y_i|\lambda_i)\,\text{gamma}\,(\lambda_i|\alpha,\beta)}\,\text{gamma}\,(\beta|.001,.001)\;\;\text{gamma}\,(\alpha|.001,.001)$$

Writing the full-conditional distribution for λ_i:

$$[\lambda_i \mid .] \propto \text{Poisson}\left(y_i \mid \lambda_i\right)\text{gamma}\left(\lambda_i \mid \alpha,\beta\right)$$

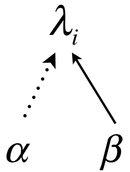


Full-conditional for β

$$[\boldsymbol{\lambda}, \alpha, \beta|\mathbf{y}] \propto \prod_{i=1}^{n} \text{Poisson}\left(y_i|\lambda_i\right) \boxed{\text{gamma}\left(\lambda_i|\alpha, \beta\right) \text{gamma}\left(\beta|.001, .001\right)} \text{gamma}\left(\alpha|.001, .001\right)$$

Writing the full-conditional distribution for β:

$$[\beta|.] \propto \prod_{i=1}^{n} \text{gamma}\left(\lambda_i|\alpha, \beta\right) \text{gamma}\left(\beta|.001, .001\right)$$

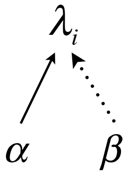


Full-conditional for α

$$[\boldsymbol{\lambda}, \alpha, \beta | \mathbf{y}] \propto \prod_{i=1}^{n} \text{Poisson}\left(y_i | \lambda_i\right) \boxed{\text{gamma}\left(\lambda_i | \alpha, \beta\right)} \text{gamma}\left(\beta | .001, .001\right) \boxed{\text{gamma}\left(\alpha | .001, .001\right)}$$

Writing the full-conditional distribution for α:

$$[\alpha \mid \cdot] \propto \prod_{i=1}^{n} \text{gamma}(\lambda_i \mid \alpha, \beta) \text{gamma}(\alpha \mid .001, .001)$$



Full-conditionals for the model

Posterior and joint:

$$
\begin{aligned}
[\boldsymbol{\lambda}, \alpha, \beta | \mathbf{y}] &\propto \prod_{i=1}^{n} \text{Poisson}\,(y_i | \lambda_i)\,\text{gamma}\,(\lambda_i | \alpha, \beta) \\
&\times \quad \text{gamma}\,(\alpha | .001, .001)\,\text{gamma}\,(\beta | .001, .001)
\end{aligned}
$$

Full conditionals:

$[\lambda_i | .] \propto \text{Poisson}\,(y_i | \lambda_i)\,\text{gamma}\,(\lambda_i | \alpha, \beta)$

$$[\beta | .] \propto \prod_{i=1}^{n} \text{gamma}\,(\lambda_i | \alpha, \beta)\,\text{gamma}\,(\beta | .001, .001)$$

$$[\alpha | .] \propto \prod_{i=1}^{n} \text{gamma}\,(\lambda_i | \alpha, \beta)\,\text{gamma}\,(\alpha | .001, .001)$$

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 expression for posterior and joint 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 the its full-conditional distribution, treating the others as if they were known and constant.
  - 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: requires a symmetric proposal distribution (e.g., normal, uniform). This is what we used above in the example for one parameter.
   1.2 Metropolis-Hastings: allows asymmetric proposal distributions (e.g., beta, gamma, lognormal). Thursday.
2. Gibbs: accepts all proposals because they are especially well chosen. Now.

Why do you need to understand conjugate priors?

- A easy way to find parameters of posterior distributions for simple problems.
- Critical to understanding Gibbs updates in Markov chain Monte Carlo as you are about to learn.

What are conjugate priors?

Assume we have a likelihood and a prior:

$$\overbrace{[\boldsymbol{\theta}|y]}^{\text{posterior}} = \frac{\overbrace{[y|\boldsymbol{\theta}]}^{\text{likelihood}} \overbrace{[\boldsymbol{\theta}]}^{\text{prior}}}{[y]}.$$

If the form of the distribution of the posterior

$[\boldsymbol{\theta}|y]$

is the same as the form of the distribution of the prior,

$[\boldsymbol{\theta}]$

then the likelihood and the prior are said to be conjugates

$\underbrace{[y|\boldsymbol{\theta}][\boldsymbol{\theta}]}_{\text{congugates}}$

and the prior is called a conjugate prior for θ.

Gibbs updates

When priors and likelihoods are conjugate, we *know* all but one of the parameters of the full-conditional because they are *assumed to be known* at each iteration. We make a draw of the single unknown *directly* from its posterior distribution as if the other parameters were fixed.

Wickedly clever.

Gamma-Poisson conjugate relationship for λ

The conjugate prior distribution for a Poisson likelihood is gamma$(\lambda|\alpha,\beta)$. Given n observations y_i of new data, the posterior distribution of λ is

$$[\lambda|\mathbf{y}] = \text{gamma}\left(\underbrace{\overbrace{\alpha_0}^{\text{The prior }\alpha} + \sum_{i=1}^{n} y_i}_{\text{The new }\alpha}, \underbrace{\overbrace{\beta_0}^{\text{The prior }\beta} + n}_{\text{The new }\beta}\right). \qquad (2)$$

Sampling from the Poisson-gamma conjugate:

Full conditional:

$$[\lambda_i \mid .] \propto \text{Poisson}(y_i \mid \lambda_i)\text{gamma}(\lambda_i \mid \alpha, \beta) \tag{3}$$

Gibbs sample:

$$[\lambda_i^k | y_i] = \text{gamma}\left(\overbrace{\alpha^{k-1}}^{\text{The current}\alpha} + y_i, \overbrace{\beta^{k-1}}^{\text{The current } \beta} + 1 \right). \tag{4}$$

In R, this would be:

```
shape[k] = alpha[k-1] + y_i
rate[k] = beta[k-1] + 1
lambda[k,i] = rgamma(1, shape[k], rate[k])
```

Gamma-gamma conjugate relationship

The conjugate prior distribution for the β parameter (rate) in a gamma likelihood gamma$(y_i|\alpha,\beta)$ is a gamma distribution gamma$\{\beta \mid \alpha_0, \beta_0)$. Given n observations y_i of new data, the posterior distribution of β (assuming that α (shape) is known) is given by:

$$[\beta|\mathbf{y}] = \text{gamma}\left(\underbrace{\overbrace{\alpha_0}^{\text{The prior } \alpha} + n\alpha}_{\text{The new } \alpha}, \underbrace{\overbrace{\beta_o}^{\text{The prior } \beta} + \sum_{i=1}^{n} y_i}_{\text{The new } \beta}\right). \quad (5)$$

We can substitute any "known" quantity for $\mathbf{y}$, e.g., $\boldsymbol{\lambda}$ in MCMC.

Sampling from the gamma-gamma conjugate:

The full conditional:

$$[\beta|.] \propto \prod_{i=1}^{n} \text{gamma}\,(\lambda_i|\alpha,\beta)\,\text{gamma}\,(\beta|.001,.001)$$

Gibbs sample:

$$\beta^k \sim \text{gamma}\left(.001 + n\alpha^{k-1},\, .001 + \sum_{i=1}^{n} \lambda_i^k\right)$$

In R this would be:

```
shape[k] = .001 + length(y) * alpha[k-1]
rate[k] = .001 + sum(lambda[,k])
beta[k] = rgamma(1, shape[k], rate[k])
```

MCMC algorithm

1. Iterate over $i = 1...197$
2. At each i, make a draw from

$$\lambda_i^k \sim \underbrace{\text{gamma}\left(\alpha^{k-1} + y_i, \beta^{k-1} + 1\right)}_{\text{Gibbs update using gamma - Poisson conjugate for } each \lambda_i} \quad (6)$$

$$\beta^k \sim \underbrace{\text{gamma}\left(.001 + \alpha^{k-1} n, .001 + \sum_{i=1}^{n} \lambda_i^k\right)}_{\text{Gibbs update using gamma - gamma conjugate for } \beta} \quad (7)$$

$$\alpha^k \propto \underbrace{\prod_{i=1}^{n} \text{gamma}\left(\lambda_i^k | \alpha^{k-1}, \beta^k\right) \text{gamma}\left(\alpha^{k-1} | .001, .001\right)}_{\text{No conguate for } \alpha \text{. Use Metropolis - Hastings update}} \quad (8)$$

3. Repeat for $k = 1...K$ iterations, storing λ_i^k, α^k and β^k. Store the value of each parameter at each iteration in a vector.

Inference from MCMC

Make inference on each unobserved quantity using the elements of their vectors stored after convergence. These post-convergence vectors, (i.e., the "rug" described above) approximate the marginal posterior distributions of unobserved quantities.

Why use Gibbs updates?

We exploit conjugate relationships to sample from the posterior because they are easier to code and because they are faster than accept-reject methods like like Metropolis or Metropolis-Hastings. However, accept-reject methods will produce the same result.