Introduction to Bayesian Statistics

Environmental Data Analytics: Part 2

Dr. Alix I. Gitelman

Statistics Department
gitelman@science.oregonstate.edu

July 2015

Overview

In Part 1, we covered some of the fundamental principals of Bayesian modeling, but there's a lot more to talk about:

- There's more to be said about prior distributions.
- The real beauty of the Bayesian approach is evidenced in hierarchical modeling.
- To fit these and other complicated models, we'll turn to Markov Chain Monte Carlo (MCMC) methods.

A Hierarchical Model

So far, most of the models that we've looked at are fairly low dimensional—they don't have very many parameters.

- As we look at more complicated models with more parameters, we're going to need to know how to sample from high-dimensional posterior distributions.
- To set the stage for discussing Markov Chain Monte Carlo methods for sampling from high-dimensional posterior distributions, let's consider a hierarchical model.

Tumors in Rats

Consider a series of 71 independent experiments on female rats, in which tumor incidence in each experiment is recorded (example from Gelman et al. 2004)

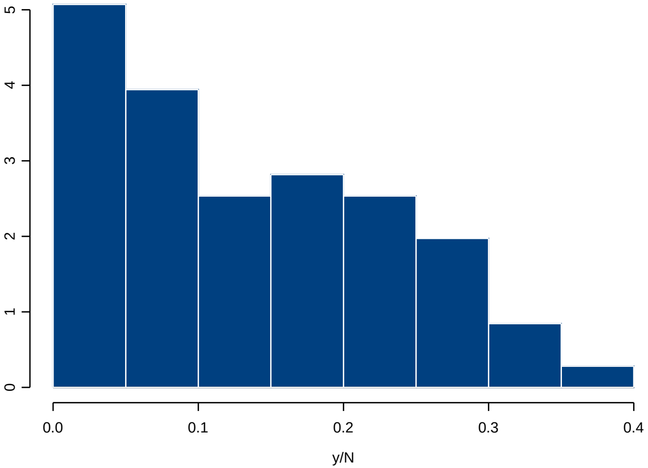
- That is, for each experiment, we have a sample of n_j rats and a count, y_j, of the number of those n_j rats that develop tumors. So that

  $$y_j|n_j, \theta_j \overset{ind}{\sim} Bin(n_j, \theta_j)$$

  for each experiment $j = 1, \ldots, 71$, where θ_j is the probability of tumors in experiment j.

- Why not just assume that $\theta_j = \theta$ for all j?

Rat Tumor Data



There's likely to be extra-binomial variation.

Rat Tumor Example (cont'd)

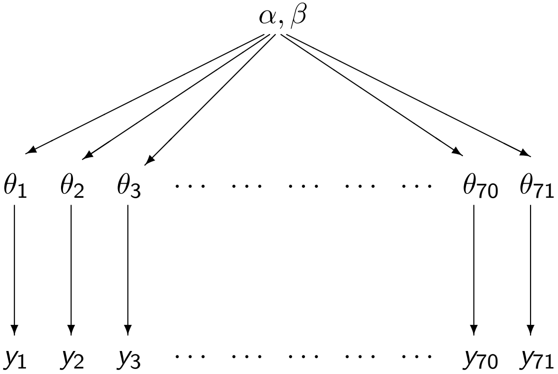
In the case where we had only one Binomial observation, we used a Beta prior for θ, and we can do the same thing here, **but we'll use the same Beta prior for all of the θ_j's**:

$$\begin{aligned} y_j | n_j, \theta_j &\overset{ind}{\sim} \text{Bin}(n_j, \theta_j) \\ \theta_j | \alpha, \beta &\overset{ind}{\sim} \text{Beta}(\alpha, \beta) \end{aligned}$$

for all $j = 1, \ldots, 71$.

- It's important to recognize that this assumes that the θ_j are a random sample from a common distribution.
- Also, α and β are NOT known here, but are to be estimated from the data—therefore, we're going to have to specify a prior (sometimes called a hyper-prior) for them.

The Beta-Binomial Hierarchical Model


α, β
θ_1 θ_2 θ_3 $\cdots$ $\cdots$ $\cdots$ $\cdots$ $\cdots$ θ_{70} θ_{71}
y_1 y_2 y_3 $\cdots$ $\cdots$ $\cdots$ $\cdots$ $\cdots$ y_{70} y_{71}

The Beta-Binomial Model

The likelihood function is just

$$f(\mathbf{y}|\boldsymbol{\theta}) = \prod_{j=1}^{71} \binom{n_j}{y_j} \theta_j^{y_j}(1-\theta_j)^{n_j-y_j} \propto \prod_{j=1}^{71} \theta_j^{y_j}(1-\theta_j)^{n_j-y_j}$$

The prior distribution for the θ_j's is

$$f(\boldsymbol{\theta}|\alpha,\beta) = \prod_{j=1}^{71} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta_j^{\alpha-1}(1-\theta_j)^{\beta-1} = \left[\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\right]^{71} \prod_{j=1}^{71} \theta_j^{\alpha-1}(1-\theta_j)^{\beta-1}$$

And for now, we'll leave the prior for α and β as $f(\alpha,\beta)$

The Beta-Binomial Model

Therefore, the joint posterior distribution of all parameters is

$$\begin{aligned} f(\boldsymbol{\theta}, \alpha, \beta | \mathbf{y}) &\propto f(\alpha, \beta) \left[\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \right]^{71} \prod_{j=1}^{71} \theta_j^{\alpha-1}(1-\theta_j)^{\beta-1} \\ &\quad \times \left[\prod_{j=1}^{71} \theta_j^{y_j}(1-\theta_j)^{n_j-y_j} \right] \\ &= f(\alpha, \beta) \left[\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \right]^{71} \prod_{j=1}^{71} \theta_j^{y_j+\alpha-1}(1-\theta_j)^{n_j-y_j+\beta-1} \end{aligned}$$

What about the Prior?

If we don't have any prior information, we'll use a non-informative prior.

- It turns out that the improper prior, $f(\alpha, \beta) \propto 1$, results in an improper posterior.
- This is a case in which it's advisable to make a transformation to the parameters α and β and then put prior distributions on those transformed values.
  - We'll let $\lambda = \alpha/(\alpha + \beta)$—this is the prior mean of the $\theta_j s$
  - And let $\kappa = \alpha + \beta$—this is a prior sample size
  - And then, we'll use uniform prior distributiosn for λ and κ.

Making Inference

To make inferences about α, β and the θ_j's:

1. Draw α and β from their joint marginal posterior distribution, $f(\alpha, \beta|\mathbf{y})$. This can be done numerically, using, for example, inverse CDF sampling or rejection sampling.

2. Draw the θ_j's from their conditional posterior distributions, given the draws of α and β (since the θ_j are conditionally independent given α and β, they can be drawn independently).

3. OR...use MCMC!

Monte Carlo Methods

Monte Carlo methods are based on simulations of random variables.

- That is, one might use Monte Carlo methods to represent the random fluctuations of some kind of natural or physical system.
- Or, one might use Monte Carlo methods to perform mathematical calculations that are impossible to solve analytically.
- Or, Monte Carlo simulations are just simulations of random variables—this is the sense in which we'll use the term.

Markov Chains

The Monte Carlo simulations that we'll use for posterior inferences are based on Markov chains (thus the name, Markov Chain Monte Carlo, or MCMC).

- A **stochastic process** is a family of random variables, X_t, where t runs over some index set, T.
- Commonly, t corresponds to discrete units of time, and the index set is just the non-negative integers. Some examples:
  - Outcomes in successive tosses of a coin
  - Repeated responses of a subject in a learning experiment
  - Stream temperatures measured through time

Markov Chains

Stochastic processes are distinguished by their **state spaces**, or the range of possible values for X_t; by their index set, T; and by the dependence relations among the X_t.

- A **Markov process** is a stochastic process with the following property: given the value of X_t, the value of X_{t+1} does not depend on the values of X_u for $u < t$.
- That is: the probability of the future behavior of a Markov process—when its current state is known exactly—is not changed by any knowledge concerning its past behavior.
- Notationally:

$$Pr(X_{t+1} = x_{t+1} | X_t = x_t, \ldots, X_1 = x_1) = Pr(X_{t+1} = x_{t+1} | X_t = x_t).$$

Markov Chains

A **discrete state-space Markov Chain** is a Markov process whose state space is a finite or countable set; and a **continuous state-space Markov Chain** is a Markov process whose state space is uncountable (continuous!).

- We're going to deal with continuous state-space Markov Chains—the state spaces for the chains we'll build are the support spaces of the parameters we're trying to estimate.
- In the Beta-Binomial hierarchical model, the state-space is:

$$[0,1] \times [0,1] \times \cdots \times [0,1] \times (0,\infty) \times (0,\infty)$$

Transition Kernels

Markov chains have **transition kernels**, which describe the probabilities of moving from one state to another state of the process.

- For *discrete-space* Markov chains, these probabilities are captured by a *transition probability matrix* with elements:

$$P_{xy} = Pr(X_{t+1} = y | X_t = x).$$

- Here's an example:

$$\begin{array}{c||cccc||}
 & 0 & 1 & 2 & 3 \\
0 & 1 & 0 & 0 & 0 \\
1 & .3 & 0 & .7 & 0 \\
2 & 0 & .3 & 0 & .7 \\
3 & 0 & 0 & 0 & 1
\end{array}$$

Transition Kernels

For *continuous-space* Markov chains, the transition kernel is a conditional probability density function, where the conditioning is on the current state of the chain:

$$f(\boldsymbol{\theta}^{(t+1)}|\boldsymbol{\theta}^{(t)}).$$

- A univariate example: suppose that the current state of a chain is $\theta^{(t)}$.
- Then, to move to the next state of the chain, we might take:

  $$\theta^{(t+1)} \sim N(\theta^{(t)}, \tau^2);$$

  that is, $\theta^{(t+1)}$ is a random draw from a Normal distribution centered at the current state, $\theta^{(t)}$.

Stationary Distributions

Roughly, a **stationary distribution** for a Markov chain is the limiting distribution of the chain. That is, it is a density, π such that

$$X_t \sim \pi \Longrightarrow X_{t+1} \sim \pi$$

as $t \longrightarrow \infty$.

- The idea here is that after the chain has run for a long time, the conditional probabilities of finding the chain in any state are stationary in that it they don't even depend on the previous state of the chain.

Markov Chain Monte Carlo Simulation

Suppose that $\boldsymbol{\theta}$ is the parameter vector of interest.

The general idea of MCMC: build a continuous state-space Markov chain whose state-space is the support space of $\boldsymbol{\theta}$, and whose stationary distribution *is* the joint posterior distribution, $f(\boldsymbol{\theta}|\mathbf{y})$.

- That is, if we run an appropriately constructed Markov chain long enough, we will draw samples from the ppdf, $f(\boldsymbol{\theta}|\mathbf{y})$.
- This is a very clever idea, and surprisingly, it turns out not to be hard to create Markov chains whose stationary distribution is $f(\boldsymbol{\theta}|\mathbf{y})$.
- The key is in the construction of the transition kernels.

Constructing Transition Kernels

To construct a transition kernel is to construct a conditional probability distribution for moving from one state to another.

- First, we need to choose a state that the chain will (possibly) move to; and for this we need a *proposal distribution*, where the proposal distribution should depend on the current state of the chain.
- Suppose that at time $t-1$, the chain is in state $\boldsymbol{\theta}^{(t-1)}$. Then let:

  $$J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$$

  denote the **proposal distribution** (also called the jumping distribution).

Constructing Transition Kernels

The Markov chain transition kernel is a mixture of the proposal distribution, $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$, and a point mass at $\boldsymbol{\theta}^{t-1)}$.

- That is, at each step in the Markov chain, we'll either accept the value proposed by the proposal distribution, or we'll stay at the current value.
- The clever idea here is that we construct the transition kernel using the posterior distribution, so that in the long run, we accept a proposed value as a draw from the posterior distribution of interest.

How Does it Work?

There are two things to understand/verify:

1. The Markov chain that we generate *has* a stationary distribution.
2. The stationary distribution *is* the posterior distribution of interest.

The existence of a stationary distribution is proved if the Markov chain is **positive recurrent** and **aperiodic** (this is the **ergodic theorem**).

Positive recurrence and aperiodicity are properties of a Markov chain that describe how it jumps from state to state.

How Does it Work?

So, using a transition kernel that is a mixture of a point mass at the current state and a carefully constructed proposal distribution ensures that the Markov chain the we build has a stationary distribution.

To see that $f(\boldsymbol{\theta}|\mathbf{y})$ is the stationary distribution of the Markov chain we construct, we have to talk about a specific algorithm for constructing the Markov chain.

- This algorithm is called the Metropolis algorithm, after Metropolis et al. (1953).
- There is also a Metropolis-Hastings algorithm, Hastings (1970)
- And now, lots of others.

Initial Values

As with most computational algorithms, a Markov Chain used in MCMC needs to be assigned an initial state. That is, we have to specify $\boldsymbol{\theta}_0$.

- Sometimes, stationarity is achieved faster depending on the starting values.
- Mostly, stationarity is achieved *regardless* of the starting values, **provided that the posterior is proper!**
- We'll typically start chains at multiple (random) initial values.

Metropolis Algorithm

Suppose that our target density is $f(\boldsymbol{\theta}|\mathbf{y})$, a posterior density. Start with $\boldsymbol{\theta}^{(0)}$, an initial value for the chain. Then for $t = 1, 2, \ldots$

(1) Sample a candidate point, $\boldsymbol{\theta}^*$, from a *jumping distribution* at time t, $J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$. The jumping distribution must be *symmetric*.

This simply means that

$$J_t(\boldsymbol{\theta}_a|\boldsymbol{\theta}_b) = J_t(\boldsymbol{\theta}_b|\boldsymbol{\theta}_a)$$

So, symmetric distributions like the Normal and the Uniform are often used as proposal distributions.

Metropolis Algorithm

(2) Calculate the ratio:

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

Notice that here is where the posterior (i.e., target distribution) is involved, and notice that since we are computing the **ratio**:

$$\frac{\text{posterior at proposed value}}{\text{posterior at current value}}$$

we don't need to know the normalizing constant (denominator of Bayes Theorem).

We **do** still need to be sure that the normalizing constant exists, however.

Metropolis Algorithm

(3) Set

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

This is the transition kernel at iteration t, where

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

Metropolis-Hastings Algorithm

The M-H algorithm generalizes the Metropolis algorithm in the following ways:

1. The jumping or proposal distribution is not required to be symmetric.
2. The ratio r is modified accordingly so that at iteration t:

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

It turns out that relaxing the symmetry requirement in the jumping distribution tends to speed up the algorithm.

The Gibbs Sampler

Let

$$f(\theta_j|\boldsymbol{\theta}_{-j}^{(t-1)}, y)$$

denote the **full** or **complete conditional** distribution of the j^{th} component of $\boldsymbol{\theta}$, conditional on all other *most recently sampled* components of $\boldsymbol{\theta}$.

That is,

$$\boldsymbol{\theta}_{-j}^{(t-1)} = (\theta_1^{(t)}, \ldots, \theta_{j-1}^{(t)}, \theta_{j+1}^{(t-1)}, \ldots, \theta_d^{(t-1)}).$$

In words, each θ_j is updated conditional on the latest values of all the other components of $\boldsymbol{\theta}$.

The Gibbs Sampler

The Gibbs sampler is a special case of the M-H algorithm in which we move to a new state at every step.

- Essentially, if the complete conditional distribution of a particular parameter (or set of parameters) has a closed form distribution, we can sample that parameter using a Gibbs step in the Markov Chain.
- If the complete conditional distribution of a parameter (or parameters) does not have a closed form, then we use a M-H (or other) step to sample it.

Rat Tumor Example

Recall that:

$$f(\alpha, \beta, \boldsymbol{\theta} | \mathbf{y}) \propto f(\alpha, \beta) \left[\frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \right]^{71} \prod_{j=1}^{71} \theta_j^{y_j + \alpha - 1} (1 - \theta_j)^{n_j - y_j + \beta - 1}$$

Tomorrow, we'll see how to use Stan to fit this model.

But for now, we'll look at the complete conditional distributions for each parameter.

Rat Tumor Example

First write the complete conditional for α:

$$f(\alpha|\beta, \boldsymbol{\theta}, \mathbf{y}) \propto f(\alpha) \left[\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)} \right]^{71} \prod_{j=1}^{71} \theta_j^{\alpha-1}$$

The complete conditional for β:

$$f(\beta|\alpha, \boldsymbol{\theta}, \mathbf{y}) \propto f(\beta) \left[\frac{\Gamma(\alpha+\beta)}{\Gamma(\beta)} \right]^{71} \prod_{j=1}^{71} (1-\theta_j)^{\beta-1}$$

Neither of these are recognizable as density functions, so we'll use Metropolis steps to sample from them.

Rat Tumor Example

The complete conditionals for each of the θ_j:

$$f(\theta_j|\alpha,\beta,\boldsymbol{\theta}_{-j},\mathbf{y}) \propto \theta_j^{y_j+\alpha-1}(1-\theta_j)^{n_j-y_j+\beta-1}$$

But this is the kernel of a Beta$(y_j+\alpha, n_j-y_j+\beta)$ density.

So, we'll have a MCMC with 71 Gibbs steps, one for each of the θ_j and two Metropolis steps—one for α and one for β.

MCMC: Monitoring Convergence

There are a few ways to evaluate Markov Chain convergence:

- Use conjugate priors whenever possible—this doesn't ensure convergence but it helps.
- Examine the trace (or history) plots of ALL parameters in the chain: you're looking for stability in the means and the variances.
- Run multiple chains, starting at different starting values and check to see whether the posterior results are the same (within MC-error) across the different chains

MCMC: Multiple Chains

Suppose that you decide to run J Markov Chains for a particular model. You run each of them for a burn-in of s iterations, followed by T iterations that you use for inference.

- Let $\{\theta_j^{(s+1)}, \theta_j^{(s+2)}, \ldots, \theta_j^{(s+T)}\}$ denote the T iterations of the jth Markov chain.
- Then the within-chain variability is just:

$$V_j = \frac{1}{T-1} \sum_{t=s+1}^{T} (\theta_j^{(t)} - \overline{\theta}_j)^2,$$

where $\overline{\theta}_j$ is the sample average computed from the jth chain.

The Gelman-Rubin Statistic

An overall assessment of within-chain variability takes the average of these V_j's across all the chains:

$$V_W = \frac{1}{J} \sum_{j=1}^{J} V_j.$$

The idea behind the Gelman-Rubin statistic is to compare this average within-chain variability to a between-chain variability calculated as

$$V_B = \frac{T}{J-1} \sum_{j=1}^{J} (\overline{\theta}_j - \overline{\theta}.)^2,$$

where $\overline{\theta}.$ is the average of the $\overline{\theta}_j$'s.

The Gelman-Rubin Statistic

The scale reduction factor (SRF) compares a pooled estimate of the two components of variance:

$$V_p = V_B/T + TV_W/(T-1)$$

with the sample (within-chain) variance, V_W. Specifically, the SRF or Gelman-Rubin statistic is:

$$\sqrt{V_P/V_W}.$$

Values of this statistic under 1.2 indicate convergence of the Markov Chain.