Introduction to OpenBUGS Tutorial

Kevin M. Esterling (UC – Riverside)

What is Bayesian inference?

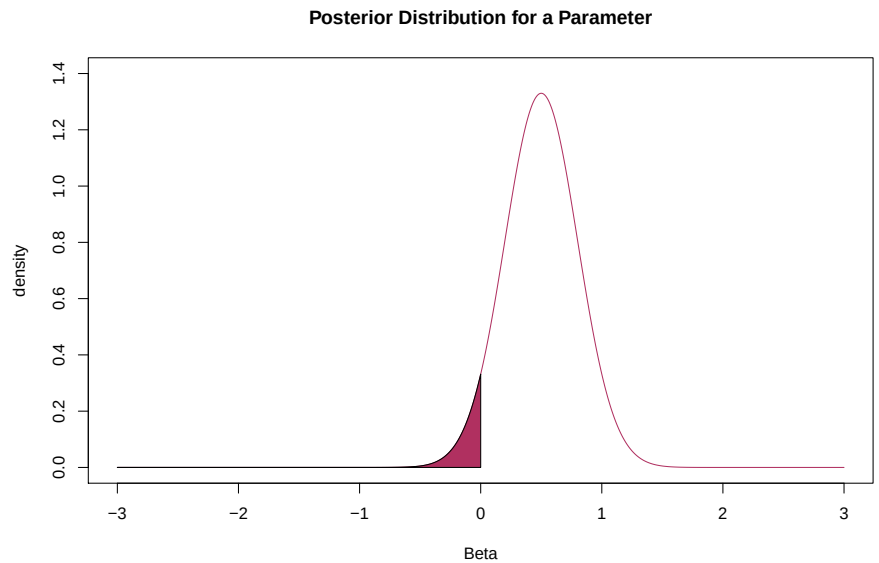
You are probably already familiar with frequentist inference ...

- Null hypothesis test procedure can tell you, if the true parameter is zero, what is the probability of observing a point estimate result of a given magnitude?
- Or that a confidence interval will cover the true parameter 95 times out of 100
- Neither of these interpretations is easy to communicate

What is Bayesian inference?

Bayesian Inference

Recover the *posterior* distribution of estimated parameters, given the model and data.



The goal of this tutorial is to show you how to estimate posterior distributions computationally using OpenBUGS

Doing Bayesian statistical analysis

General form of Bayes rule for statistical modeling:

$$p(\beta|y) = \frac{p(\beta)p(y|\beta)}{p(y)}$$

In words, the posterior density (beliefs after seeing the data) is proportional to the prior density (beliefs before seeing the data) times the likelihood of observing the data given those prior beliefs, divided by a normalizing constant.

We can drop the normalizing constant that makes the posterior a true probability density

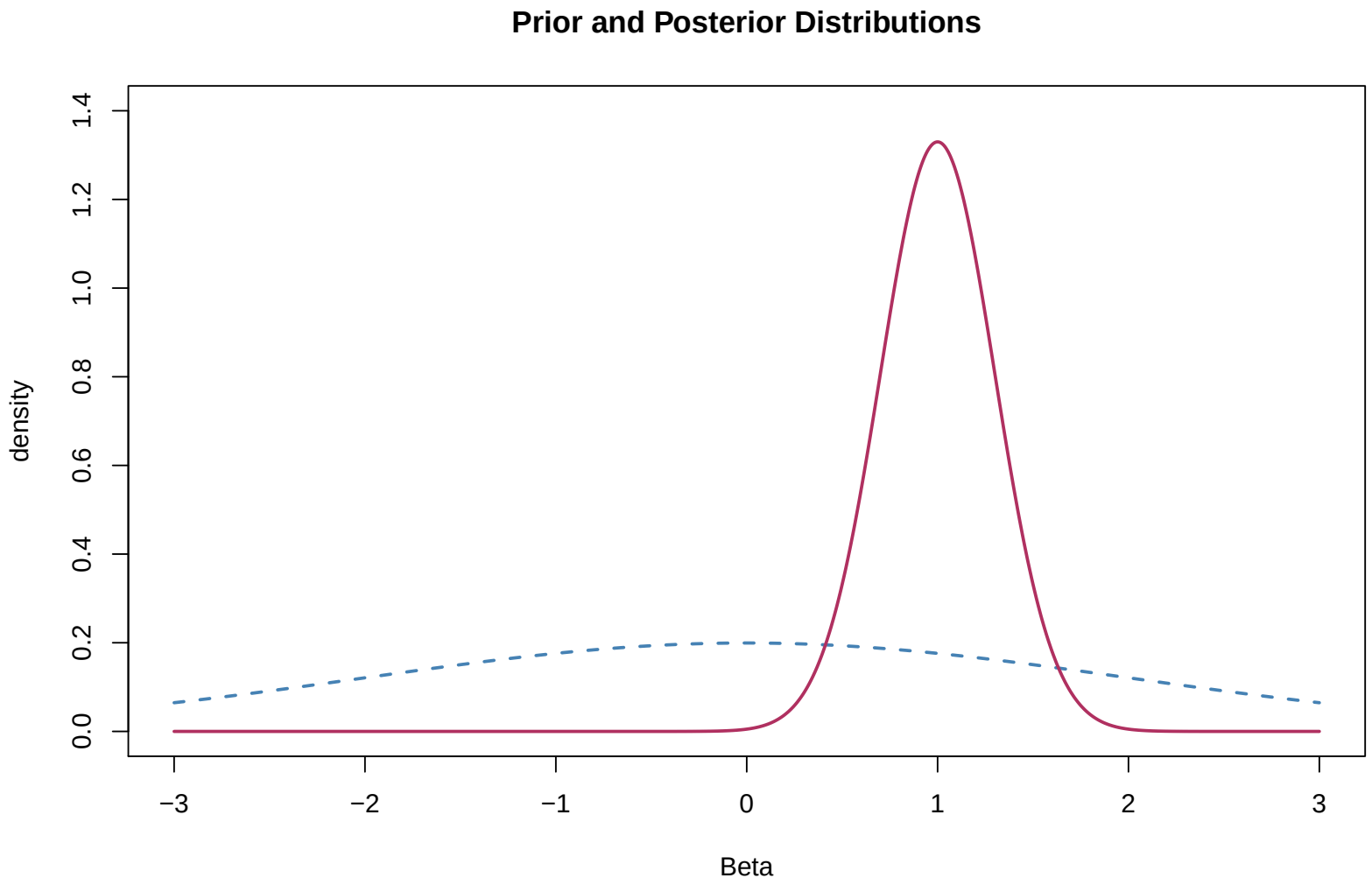
$$p(\beta|y) \propto p(\beta)p(y|\beta)$$

Doing Bayesian statistical analysis

Bayes Rule:

$$\text{Posterior Beliefs} = \frac{\text{Prior Beliefs} \times \text{Data Likelihood}}{\text{Probabilty of the Data}} \tag{1a}$$

$$\propto \text{Prior Beliefs} \times \text{Data Likelihood} \tag{1b}$$


Prior and Posterior Distributions
density
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
−3
−2
−1
0
1
2
3
Beta

Doing Bayesian statistical analysis in the real world

Bayes rule can sometimes be solved analytically, but Bayesian computational methods allow you to solve arbitrary, complex models much more flexibly and/or computationally faster. The goal of this tutorial is to show you how to estimate posterior distributions computationally using OpenBUGS.

Simulated Data Example

$$\begin{aligned} \text{xpostO1}_i &\sim \phi(\mu_i, \tau) \\ \mu_i &= \beta_0 + \beta_1 \mathit{Site}_{2i} + \beta_2 \mathit{Site}_{3i} \end{aligned}$$

Where,
$\beta_0 = 1.793$
$\beta_1 = -0.690$
$\beta_2 = 0.352$
$\tau = 0.379$ is the mean square precision

OLS regression model

Likelihood:

$$\left.\begin{array}{rcl} Y_i & \sim & \phi(\mu_i, \tau) \\ \mu_i & = & \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} \end{array}\right\} 1 \leq i \leq \text{n.obs}$$

IID assumption

Priors:

$$\begin{array}{rcll} \beta_0 & \sim & \phi(0, 0.0001) & \text{Flat priors ("uninformative")} \\ \beta_1 & \sim & \phi(0, 0.0001) & \\ \beta_2 & \sim & \phi(0, 0.0001) & \\ \tau & \sim & U(0, 1000) & \text{Flat positive prior} \end{array}$$

OLS regression model

Likelihood:

$$\left.\begin{aligned} Y_i &\sim \phi(\mu_i, \tau) \\ \mu_i &= \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} \end{aligned}\right\} 1 \leq i \leq \text{n.obs}$$

IID assumption

Priors:

$$\begin{aligned} \beta_0 &\sim \phi(0, 0.0001) \\ \beta_1 &\sim \phi(0, 0.0001) \\ \beta_2 &\sim \phi(0, 0.0001) \\ \tau &\sim U(0, 1000) \end{aligned}$$

Flat priors ("uninformative")

Flat positive prior

OLS regression model

Likelihood:

$$\left.\begin{array}{rcl} Y_i & \sim & \phi(\mu_i, \tau) \\ \mu_i & = & \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} \end{array}\right\} 1 \leq i \leq \text{n.obs} \quad \text{IID assumption}$$

Priors:

$$\begin{array}{rcll} \beta_0 & \sim & \phi(0, 0.0001) & \text{Flat priors ("uninformative")} \\ \beta_1 & \sim & \phi(0, 0.0001) & \\ \beta_2 & \sim & \phi(0, 0.0001) & \\ \tau & \sim & U(0, 1000) & \text{Flat positive prior} \end{array}$$

OLS regression model

Likelihood:

$$\left.\begin{array}{rcl} Y_i & \sim & \phi(\mu_i, \tau) \\ \mu_i & = & \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} \end{array}\right\} 1 \leq i \leq \text{n.obs} \quad \text{IID assumption}$$

Priors:

$$\begin{array}{rcll} \beta_0 & \sim & \phi(0, 0.0001) & \text{Flat priors ("uninformative")} \\ \beta_1 & \sim & \phi(0, 0.0001) & \\ \beta_2 & \sim & \phi(0, 0.0001) & \\ \tau & \sim & U(0, 1000) & \text{Flat positive prior} \end{array}$$

Computational Bayesian statistics: MCMC

- "Bayesian estimation using Gibbs sampling" (`OpenBUGS`) uses *simulation* to approximate the posterior distribution
- You give the software the **model** (priors and likelihood only), **data** and **starting values**, and the software will draw a sequence of realizations from the posterior distribution to create an empirical approximation of the posterior parameter distribution
- Basic procedure to run the simulation:
  - Start at an arbitrary set of initial values
  - Discard "burn-in period" draws
  - Save and analyze "stationary period" draws

Computational Bayesian statistics: MCMC

Table: Simulated Posterior Distribution

	β_0	β_1	β_2
Burn-in Period			
t_0	20	-200	12
t_1	17	-105	15
t_2	2	-2	2
t_3	0.9	1.5	0.0
...			
t_{9997}	0.7	1.7	0.87
t_{9998}	0.8	1.8	0.89
t_{9999}	0.6	1.7	0.95
t_{10000}	0.6	2.0	0.91

Computational Bayesian statistics: MCMC

Table: Simulated Posterior Distribution

	β_0	β_1	β_2
Stationary Period			
t_{10001}	0.7	1.9	0.89
t_{10002}	0.6	1.5	0.87
t_{10003}	0.8	1.6	0.89
t_{10004}	0.5	1.8	0.83
t_{10005}	0.7	2.1	0.99
...			
t_{10997}	0.4	2.2	0.87
t_{10998}	0.7	1.7	0.97
t_{10999}	0.6	1.9	0.99
t_{11000}	0.9	1.5	1.02

Computational Bayesian statistics: MCMC (cont.)

Result is a simulated posterior distribution: computational approximation of the posterior

- The vector of draws post-convergence for each parameter is the marginal posterior distribution
- Summarize (mean, SD, 95% intervals) and plot densities
- Trivial to create sampling distributions of functions of parameters $(\widehat{\frac{ln(\beta_0)}{1+sin(\beta_1)}})$

Computational Bayesian statistics: MCMC (even still cont.)

MCMC procedure

1. Specify model (likelihood and priors) with `OpenBUGS` code
2. Create files with data and initial values (using RStudio)
3. In OpenBUGS:
   - Check model, load data, and compile model
   - Provide initial values for parameters
   - Run model for an initial "burn-in" period until MCMC converges on the posterior distribution
   - Save a sample of draws from the posterior for parameters of interest
4. Summarize marginal distributions, plots, statistical tests

Common problems and some advice

- Always run multiple chains (usually three) in order to test convergence
- Assess burnin period, mixing carefully, use BGR diagnostic
- Be sure there are no missing data on RHS
- Read the manual; and Gelman and Hill (2006) is a great resource for multilevel modeling
- Learn scripting language

Using OpenBUGS with R

In practice, you want to store your data and analyze/graph results within R (or Stata or SAS etc.)

- Once you know how to use OpenBUGS you can read documentation to these R packages:
  - R2OpenBUGS, BRugs = Interact with OpenBUGS within R
  - CODA = Suite of tools to assess convergence and describe results
  - BRugs installs/loads all three
- Call OpenBUGS from R for automating Bayesian analysis