Introduction to OpenBUGS Tutorial

Kevin M. Esterling (UC – Riverside)

Presentation to GradQuant

March 1, 2023

Traditional statistics relies on “frequentist inference”

Philosophy of Frequentist Inference

Probability is an **objective** property of the external/natural world

- The parameters of the distribution governing a random variable cannot be observed
- Inferring distribution parameters instead requires repeated observations; e.g., 1,000 coin flips
- The notion of repeated sampling motivates frequentist hypothesis testing

Frequentist hypothesis testing

- The null hypothesis significance test (NHST) procedure can tell you, if the true parameter is zero, what is the (p-value) probability of observing a point estimate 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 or makes direct statements about the parameters of interest; only to accept or reject the null

What is Bayesian inference?

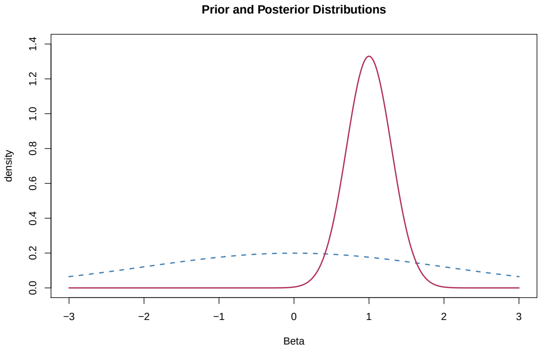
Philosophy of Bayesian Inference

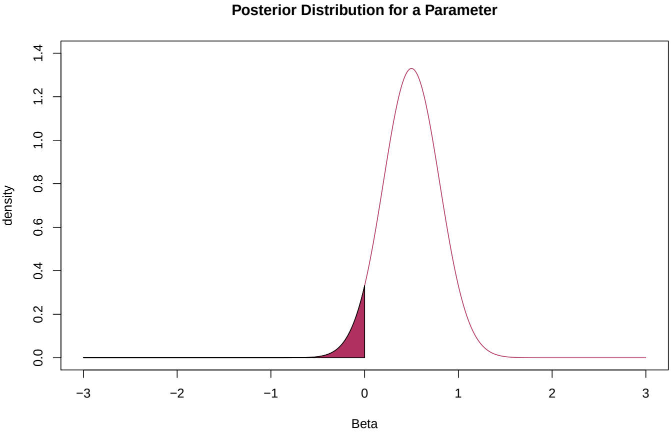
Bayesian inference posits that probability is best conceived of as a **subjective belief**

- The goal of scientific research is to change others' **beliefs** about properties of the world
- Bayesian analysis is a way to inform your audience how they rationally should change their beliefs after observing data
- And beliefs are what really matter for decision making – for example, it was my belief that the Covid vaccine works that led me to take it

Bayesian Inference

The *prior* distribution (blue curve) characterizes one's beliefs before observing any data. The *posterior* distribution (maroon curve) characterizes one's rational beliefs about the world, given the model and data.





That is, we can say the *probability* that the treatment is effective.

Why is Bayesian inference useful?

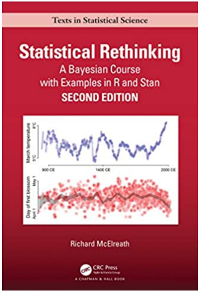
If you agree with the subjective view, cool things happen:

- Reporting results is intuitive; e.g., probability of a heads
- A single observation can be quite meaningful
- We often hold informed beliefs about probabilities/distributions before we observe data, so that information should not be discarded

Practical reasons to be Bayesian, given computational methods:

- Can approximate frequentist results
- But Bayesian methods are much more flexible and/or computationally faster
- Easier to state uncertainty about arbitrary functions of parameters
- Natural way to multiply impute missing data

This workshop will not make you a Bayesian statistician! But it will get you started. The best place to start is the McElreath book:



You might also look at these excellent lecture notes from Mark Lai, who is a USC Psyc professor:
https://bookdown.org/marklhc/notes_bookdown/

OpenBUGS vs. Stan

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

OpenBUGS is easier to teach, but once you understand the principles, you might consider switching to Stan:

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

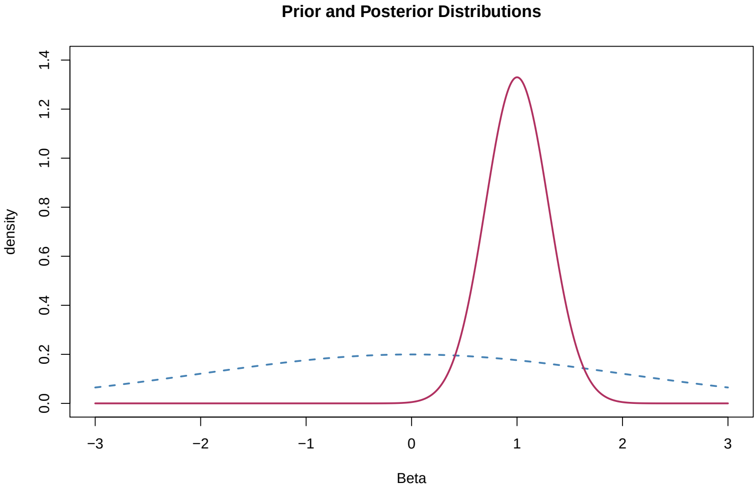
$$p(\beta|y) \propto p(\beta)p(y|\beta)$$

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

Analytical Bayesian statistical analysis

The old fashioned way was to derive the posterior *analytically*, often using "conjugate" priors

- A conjugate prior for a likelihood yields a posterior in the same form as the prior
- Example, conjugate prior for a binomial distribution is the beta distribution . . .
- . . . or the conjugate prior for a normal distribution is the normal distribution

Analytical solution for conjugate normal prior

Let $y_i \overset{iid}{\sim} N(\mu, \sigma^2), i = 1, \ldots, n$, with σ^2 known, and $\mathbf{y} = (y_1, \ldots, y_n)'$. If $\mu \sim N(\mu_0, \sigma_0^2)$ is the prior density for μ, then μ has posterior density,

$$\mu|\mathbf{y} \sim N\left(\frac{\mu_0 \frac{1}{\sigma_0} + \bar{y}\frac{n}{\sigma^2}}{\frac{1}{\sigma_0} + \frac{n}{\sigma^2}}, \left(\frac{1}{\sigma_0} + \frac{n}{\sigma^2}\right)^{-1}\right)$$

Results:

- *Posterior mean* is a variance-weighted average of the prior and data
- *Posterior variance* is the weighted sum of the prior variance and the data variance
- Note: Bayesian and MLE converge with diffuse priors and/or lots of data

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.

$$
\begin{aligned}
O_{ij}^m \sim & \text{OverdispersedPoisson}(\widetilde{\lambda_{ij}^m}) & \text{(1a)}\\
& ln\widetilde{\lambda_{ij}^m} = \lambda_{ij}^m = \beta_0^m + \beta_1^m d(L_{l_j}, L_{a_i}) + \beta_2^m R_i + \beta_3^m d(L_{l_j}, L_{a_i}) R_i + \boldsymbol{\eta}_{ij}^m \\
L_{l_j} \sim & \text{Normal}(\mu_{l_j}, \sigma^{L_l}) & \text{(1b)}\\
& \mu_{l_j} = \alpha_0 + \alpha_1 \psi_{l_j} \\
L_{a_i} \sim & \text{Normal}(\mu_{a_i}, \sigma^{L_a}) & \text{(1c)}\\
& \mu_{a_i} = (\alpha_0 + \alpha_2) + (\alpha_1 + \alpha_3)\psi_{a_i}.
\end{aligned}
$$

Simulated Data Example

$$
\begin{aligned}
Y_i &\sim \phi(\mu_i, \tau) \\
\mu_i &= \beta_0 + \beta_1 Site_{2i} + \beta_2 Site_{3i}
\end{aligned}
$$

Where,
$\beta_0 = 1.793$
$\beta_1 = -0.690$
$\beta_2 = 0.352$
$\tau = 0.379$ is the "precision," i.e., the inverse of the variance

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:

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

Flat priors ("uninformative")

Flat positive prior

OLS regression model

Likelihood:

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

IID assumption

Priors:

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

Flat priors ("uninformative")

Flat positive prior

OLS regression model

Likelihood:

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

Priors:

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

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:

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

Flat priors ("uninformative")

Flat positive prior

Computational Bayesian statistics: MCMC

"Bayesian estimation using Gibbs sampling" (`OpenBUGS`) uses *simulation* to approximate the posterior distribution

- Gibbs sampling: sample an estimate from a candidate posterior distribution for each parameter, conditional on the current estimate of all other parameters
- MCMC = "Markov Chain Monte Carlo" = run the Gibbs sampler repeatedly until the parameters estimates converge to 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

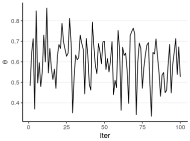
	β_0	β_1	β_2
Burn-in Period			
t_0	20	-200	12
t_1	17	-105	15
t_2	2	-2	2
t_3	0.9	1.5	0.0
...			
t_{9997}	0.7	1.7	0.87
t_{9998}	0.8	1.8	0.89
t_{9999}	0.6	1.7	0.95
t_{10000}	0.6	2.0	0.91

Computational Bayesian statistics: MCMC

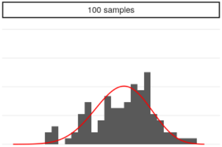
Table: Simulated Posterior Distribution

	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

We can plot the draws,

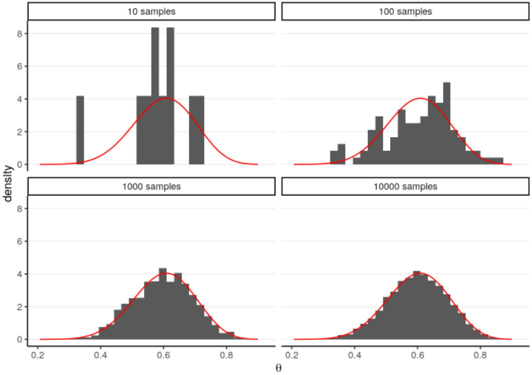


and then visualize them in a histogram to approximate the posterior distribution

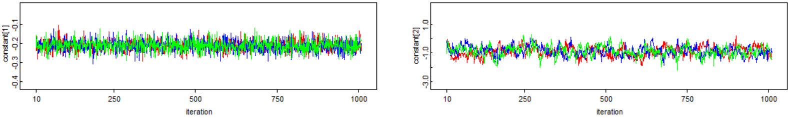


These figures are from the Lai notes

As the number of samples increases, the histogram will converge to the posterior distribution



This figure is from the Lai notes



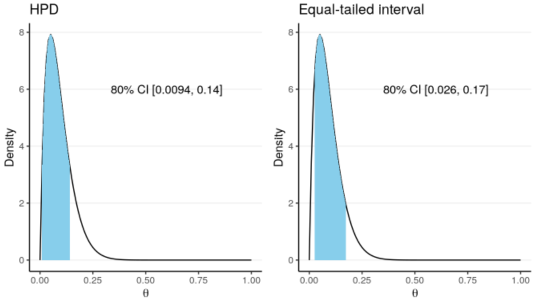
- Normally we use multiple chains to test for stationarity/convergence
- Note: in practice, because the samples are a Markov chain, the draws are correlated – which reduces the number of effective observations

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 (for example, say you need to know $\widehat{\frac{ln(\beta_0)}{1+sin(\beta_1)}}$)
- Normative way to impute missing data for correct standard errors using data augmentation

Bayesian uncertainty is generally reported using “credible intervals” – using either highest posterior density or equal tails



This figure is from the Lai notes

Evaluating model quality

Bayesian methods allow you to make better assessments of the quality of your models:

- **Posterior predictive check** – how well the estimated model can simulate important features of the data
- **Information criteria - AIC, DIC, WAIC** – how well the estimated model will predict out of sample
- **Bayesian model averaging** – instead of presenting a single model, average the results across plausible models weighted by their information
- **Regularization** – use stronger priors that shrink parameters to prevent overfitting, reducing the effective number of parameters

Computational Bayesian statistics: MCMC (even still cont.)

MCMC workflow

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. Read results back into `R` and summarize marginal distributions, plots, statistical tests, posterior evaluation

Hands on example...

I’ll do this example now, and later you can do it yourself
https://bit.ly/mcmclinks

Common problems and some advice

- Always run multiple chains (usually three or four) in order to test convergence
  - BGR diagnostic assesses within-to-between chain variance (assumes overdispersed/random initial values)
  - Consider both mathematical and empirical identification (just because you can write it down does not mean you should estimate it)
  - Best to start with simple model and build up complexity
- Assess burnin period, mixing carefully, use BGR diagnostic
- Be sure there are no missing data on RHS
- 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
- (the R package for Stan is RStan)

Installation

Install OpenBUGS following the instructions at https://bit.ly/MCMCworkshop

- Note that OpenBUGS has versions for Windows and Linux but not Mac. I am not a Mac user so I cannot provide support for that. But I did a search and found a website that seems to give correct instructions which you can follow in that same link
- Stan has version native to Mac
- Also at that link is the .ODC file for today's practice example