Bayesian Thinking: Fundamentals, Regression and Multilevel Modeling

Jim Albert and Jingchen (Monika) Hu

November 2020

General Overview

This webinar series is divided into three parts.

- Part 1 (November 3) Bayesian Fundamentals
- Part 2 (November 10) Bayesian Regression
- Part 3 (November 17) Multilevel Modeling

Structure

- Each part will consist of two 50-minute presentations, followed by a 10 minute time for questions.
- We encourage you to submit questions during the presentations.
- All presentations and R code will be posted on our Github site.

Our Backgrounds

- Jim Albert has taught a Bayesian graduate course at Bowling Green State University for many years.
- Monika Hu has taught a Bayesian class at the undergraduate level at Vassar College.
- Recently we coauthored the text *Probability and Bayesian Modeling*.
- Web version of the book available at

http://bitly.com/ProbBayes

Files

All of the webinar files (markdown, pdf, data files) can be found at

https://github.com/bayesball/JSM2020

Any Questions?

- Can ask questions by use of the Zoom chat window.
- We will try to address all questions during the Webinar or afterwards

Computation

- Our text focuses on the use of JAGS for simulating from general Bayesian models.
- Here we are going to focus on the use of Stan which implements Hamiltonian MCMC sampling.
- Stan is well-supported and is especially efficient in fitting multilevel models.

Using Stan

- Stan modeling language
- Interfaces with popular computing environments
- Package `rstan` is the R interface to Stan
- Write a script defining the Bayesian model

Higher Level Interface

- Packages `brms` and `rstanarm` provide R formula type interfaces to Stan.
- We will illustrate the use of `brms` for the regression and multilevel modeling examples.

Example: Sleeping Patterns of Students

- Recent StatCrunch survey of high school students
- Each student asked “What is an average hours of sleep for you per night?”
- Interested in the outcome “average is at least 8 hours”
- Observed data $y_1, ..., y_n$ where $y_i = 1$ (student averages at least 8 hours of sleep) or $y_i = 0$

A Bayesian Model

- (Sampling) $y_1, ..., y_N$ are independent Bernoulli(p)
- p is the proportion of all students who average at least 8 hours of sleep
- (Prior) p is random, assign it a prior density $g(p)$
- Prior represents one's subjective beliefs about the location of p

Choice of Prior?

- Convenient to let p have a beta density

$$g(p) \propto p^{\alpha-1}(1-p)^{\beta-1}, 0 < p < 1$$

- Choose shape parameters α and β to reflect beliefs about p

Specifying a Beta Prior

- Hard to specify values of the shape parameters directly.
- Indirectly specify shape parameters by specifying quantiles of p
- Specify a median (best guess at p)
- Specify a 90th percentile (indicates sureness of your guess)
- Find values of α, β that match values of median and 90th percentile

Shiny App

- I wrote a Shiny app to help one specify a subjective beta prior for a proportion

https://bayesball.shinyapps.io/ChooseBetaPrior_3/

- Use a slider to specify two percentiles
- Graph shows the matching beta prior

Predictive density

- Bayesian model specifies joint density of (p, y):

$$f(p, y) = g(p) f(y|p)$$

- (Prior) predictive density is marginal density of y

$$f(y) = \int g(p) f(y|p) dp$$

- This represents what one predicts in a future sample of a particular size.

Choosing a Prior

- I think relatively few students average 8 or more hours of sleep
- My best guess at p is 0.15
- Pretty sure (with probability 0.90) that p is smaller than 0.25
- This information is matched up with a beta prior with $\alpha = 4.42$ and $\beta = 23.51$

Checking My Prior

- This prior says that my 90% interval estimate is

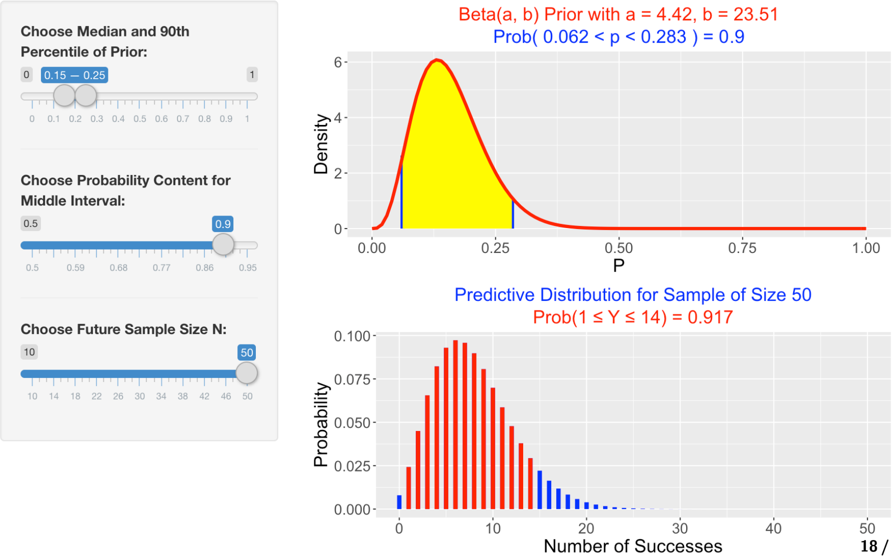
$$P(0.062 < p < 0.283) = 0.90$$

- Suppose I think about a future sample of 50 where Y is the number of students who average 8+ hours of sleep. My prior implies

$$P(1 \leq Y \leq 14) = 0.917$$

- If these bounds don't seem right, adjust your statements about the median and 90th percentile

Constructing a Beta(a, b) Prior From Two Quantiles



Updating Beliefs

- Sample $N = 44$ students, observe $Y = 7$ who average more than 8 hours of sleep.
- [Y | p] is binomial(N, p)
- Posterior density is product of likelihood and prior

$$g(p|y) \propto p^{Y}(1-p)^{N-Y} \times p^{\alpha-1}(1-p)^{\beta-1}$$

- Here we get

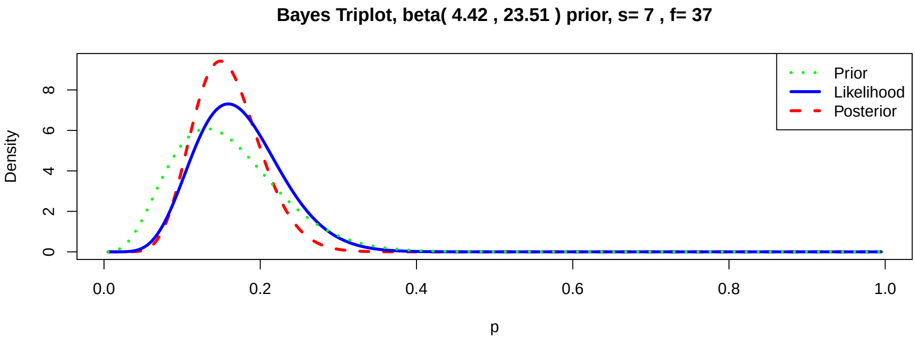
$$g(p|y) \propto p^{\alpha+Y-1}(1-p)^{\beta+N-Y-1}, 0 < p < 1$$

- Posterior is also beta with parameters $\alpha + Y$ and $\beta + N - Y$

Example

- Prior is Beta(4.42, 23.51)
- Observe $Y = 7$ in a sample of $N = 44$
- Posterior is Beta(4.42 + 7, 23.51 + 37) = Beta(11.42, 60.51)

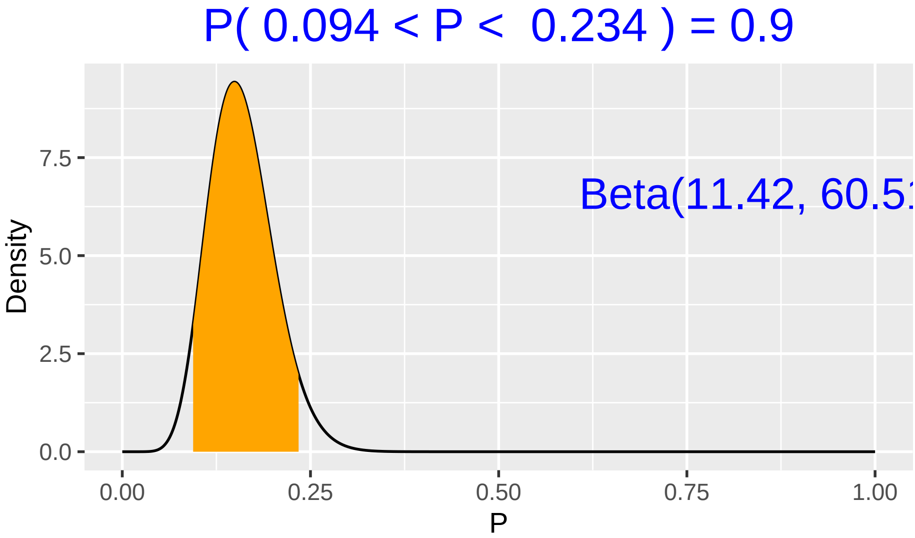
Bayesian Triplot - Show Prior, Likelihood and Posterior



Bayesian Inferences

- All inferences about p are summaries of this posterior density
- For example, a 90% interval estimate is an interval that covers 90% of the posterior probability

90% Interval Estimate



Simulation-Based Inference

- Here we can find the exact posterior distribution
- But in most situations, we cannot, but it is possible to simulate from the posterior
- Simulate many draws from the posterior and implement inference by summarizing the simulated sample

Simulation for Our Example

- Simulate 1000 draws from the beta posterior.

```
p_sim <- rbeta(1000, 11.42, 60.51)
```

- Find 90% interval estimate by computing quantiles of the simulated draws.

```
quantile(p_sim, c(0.05, 0.95))
```

```
##         5%        95%
## 0.09197345 0.23765737
```

Prediction

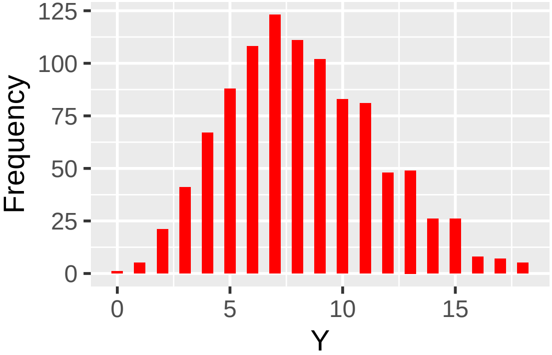
- Suppose one wishes to predict the number y^* of 8+ hours of sleep in future sample of 50 students
- Interested in posterior predictive (PP) density.

$$f(y^*|y) = \int f(y^*|p)g(p|y)dp$$

- Simulate draws from PP density by (1) simulating p from posterior and (2) simulating $y|p$ from the sampling density

```
p_sim <- rbeta(1000, 11.42, 60.51)
ys <- rbinom(1000, size = 50, prob = p_sim)
```

Prediction in Our Example




```r
quantile(ys, c(0.05, 0.95))
```

```
##  5% 95%
##   3  14
```

Using Stan

- Write a Stan script defining the Bayesian model.

```
data {
  int<lower=0> N;
  int<lower=0,upper=1> y[N];
}
parameters {
  real<lower=0,upper=1> theta;
}
model {
  theta ~ beta(4.42, 23.51);
  for (i in 1:N) {
      y[i] ~ bernoulli(theta);
  }
}
```

Enter data by a list

```
library(readr)
d <- read_csv("Happiness_vs_Sleep_Exercise.csv")
d$y <- ifelse(d$Exercise >= 8, 1, 0)
my_data <- list(N = 44, y = d$y)
```

Run Stan Using the `rstan` package

- Inputs are Stan model file and the data list.

```
library(rstan)
fit_bern <- stan(file = "bern_beta.stan",
                 data = my_data,
                 refresh = 0)
```

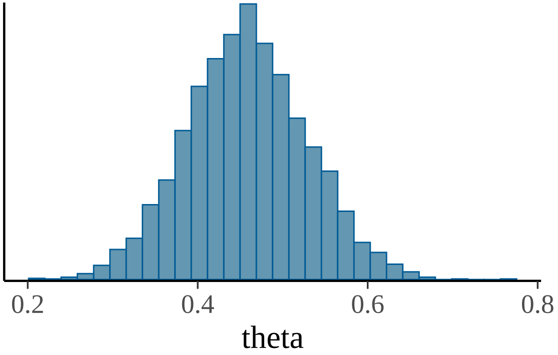
Extract the posterior draws

```
draws <- as.data.frame(fit_bern)
head(draws)
```

```
##       theta      lp__
## 1 0.5552438 -32.61103
## 2 0.4756824 -31.74454
## 3 0.3155454 -33.70083
## 4 0.3303363 -33.28502
## 5 0.4909249 -31.81974
## 6 0.4806227 -31.76424
```

Histogram of the simulated draws of p

```
library(bayesplot)
mcmc_hist(draws, pars='theta')
```



Posterior summaries

```
summary(fit_bern)
```

```
## $summary
##                mean     se_mean         sd        2.5%
## theta     0.4576733 0.00189036 0.07101391   0.3169804    0.40
## lp__    -32.1929326 0.01765811 0.70325701 -34.1575695 -32.34
##                 75%        97.5%   n_eff       Rhat
## theta     0.5042715   0.5990542 1411.229 1.0007915
## lp__    -31.7562443 -31.7109801 1586.134 0.9996238
##
## $c_summary
## , , chains = chain:1
##
##          stats
## parameter        mean          sd        2.5%         25%
##     theta   0.4565847 0.07008792   0.3176341   0.4123798
##     lp__   -32.1817956 0.73031214 -34.2578436 -32.2992104 -3
```

Wrap-Up: Some Attractive Features of Bayes

- One recipe (Bayes' rule) for implementing inference
- Conditional inference
- Allows input of prior opinion

More Attractive Features

- Intuitive conclusions
- The probability that p is in $(0.23, 0.45)$ is 90 percent.
- If you have hypothesis $p \leq 0.7$, you can compute the probability a hypothesis is true.
- Prediction and inference: in both cases you are learning about unobserved quantities given observations

Some More Attractive Features of Bayes

- Flexibility in modeling
- Advances in Bayesian computation
- Attractive way to implement multilevel modeling
- Can handle sparse data (say, many 0's in categorical response data)

Some Issues with Bayes

- "QUESTION: What if I use the wrong prior?"
- "QUESTION: Aren't I introducing errors by simulating from the posterior?"