Bayesian Computing in the Statistics and Data Science Curriculum

Abstract

This Supplementary Material contains four learning activities introduced in the main text, and R code for Section 2.3 Normal Approximation.

1 Learning Activities

1.1 Discrete Bayes Activity: "Learning About a Proportion"

Introduction

In this activity, you will gain experience in constructing a discrete prior distribution for a proportion that reflects your beliefs about the location of the proportion.

Helpful Apps

The "Learning About a Proportion Using Bayes' Rule" app at
`https://bayesball.github.io/nsf_web/jscript/p_discrete/prior2a.htm`
is helpful in computing posterior probabilities using a discrete prior.

Story

Suppose you are interested in the proportion p of students from your campus who need corrective vision. **Note:** This activity can be adjusted to learn about any proportion that might be of interest to your students.

Part 1: Choosing a Prior

1. Make a short list of plausible values for p. In the event that you can't easily construct this list, use the eleven values $p = 0$, 0.1, 0.2, ..., 0.9, 1. Write down the following table with columns p, Weights and Prior and place your values of p in the p column.

p	Weight	Prior

2. Assign a weight of 10 to the value of the proportion p that is most likely. Put this weight value in the table.

3. Assign weights of 5 to those values of the proportion p that you believe are half as likely as the value of p that you selected in part 2. In a similar manner, assign integer weight values (say 1, 2, 3, 4, 5, 6, 7, 8, or 9) to the other values of p. Place all of these weight values in your table.

4. Compute the sum of the weight values and put this sum at the bottom of the Weight column.

5. Find the probabilities for your prior by dividing each weight values by the sum of the weights.

6. Using your prior, find the probability that the proportion of students needing corrective vision is over 0.5.

7. Find the probability that the proportion of students needing corrective vision is at most 0.3.

Part 2: The Posterior

1. Suppose you collect data from a sample of 20 students. Of this sample, 13 students need some type of corrective vision. Using Bayes' rule find the posterior of the proportion p.

2. Display your prior and posterior probabilities on the same scale. Describe how your prior opinions have changed in the light of this new information.

3. Using your posterior, compute the probability that the proportion is over 0.5, and the probability that the proportion is at most 0.3. Describe how these probability computations have changed from the prior to the posterior.

1.2 Conjugate Prior Activity "Did Shakespeare Use Long Words?"

Helpful Apps

The "Constructing a Beta(a, b) Prior From Two Quantiles" at
`https://bayesball.shinyapps.io/ChooseBetaPrior_3/`
is helpful in constructing a conjugate beta prior for a proportion.

Introduction

One way to measure the complexity of some written text is to look at the frequency of long words, where we will define a "long word" as one that has 7 or more characters. Actually, we are interested in the fraction of all words that are long. For example, consider this sentence (from Moby Dick):

"These **reflections** just here are **occasioned** by the **circumstance** that after we were all seated at the table, and I was **preparing** to hear some good **stories** about **whaling**; to my no small **surprise**, nearly every man **maintained** a **profound silence**."

There are a total of 41 words of which 10 (the ones in bold type) are long, so the fraction of long words is $10/41 = 0.24$.

Let P denote the proportion of long words among all of the plays written by William Shakespeare.

Part 1: Choosing a Prior

1. Without looking at any Shakespeare text, make an educated guess at the value of P. You will be specifying the median, the value M such that it is equally likely that P is smaller or larger than M (that is, $Prob(P < M) = 0.5$) .

2. Without looking at any Shakespeare text, find the 90th percentile P_{90} such that your prior probability that P is smaller than P_{90} is equal to 0.90 (that is, $Prob(P < P_{90}) = 0.90$).

3. Based on your answers to questions 1 and 2, use the app at

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

   to find the shape parameters of your beta prior that match your statements about the values of M and P_{90}.

4. Using the app, find the values of P that bracket the middle 50% of the prior probability, and the values of P that bracket the middle 90% of the prior probability. Put these values below:

   50% interval: ____________

   90% interval: ____________

5. Reflecting on the 50% and 90% intervals, are you interested in changing your statements about the values of M and P_{90}? If so, adjust your values of M and P_{90} and find your new values of the shape parameters of your beta prior.

Part 2: The Posterior Analysis

1. Now collect some data. Going to

   `http://shakespeare.mit.edu/`

   choose one play and select approximately 100 words from your chosen play. Paste your selection of words to the site

   `https://wordcounttools.com/`

   This site will count the number of words in your text and also give you the count of long words. Record the number of words N and number of long words Y you find.

2. Find the shape parameters of the beta posterior for P that combines your prior found above with the data information.

   Using the `qbeta()` and `pbeta()` functions in R or the app at `https://homepage.divms.uiowa.edu/~mbognar/applets/beta.html`

to answer questions 8 through 10.

3. Find the posterior median.
4. Find the posterior probability that P is larger than 0.20.
5. Find a 90 percent interval estimate for P.

1.3 Activity: Normal Approximation to Posterior

Introduction

This activity explores the accuracy of the normal approximation in a situation where the actual posterior density is not normal in shape.

Description

Suppose you are interested in learning about the standard deviation in the time that it takes to commute to work. You collect the the following times (in minutes) for 8 trips:

11 2 10 7 8 5 9 5 6 9

Assume that your commuting time y is normally distributed with known mean of 10 seconds and standard deviation σ. Assuming that your commuting times are independent, the likelihood function of σ is equal to

$$L(\sigma) = \prod_{j=1}^{8} \frac{1}{\sigma} \exp\left(-\frac{1}{2\sigma^2}(y_j - 10)^2\right)$$

If you place a uniform prior on σ, then the posterior density of σ is proportional to:

$$g(\sigma|y) = \frac{1}{\sigma^8} \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{8}(y_j - 10)^2\right), \ \sigma > 0$$

For these data, $\sum_{j=1}^{8}(y_j - 10)^2 = 146$, so the posterior density is proportional to:

$$g(\sigma|y) = \frac{1}{\sigma^8} \exp\left(-\frac{1}{2\sigma^2}146\right), \ \sigma > 0$$

1. Graph this posterior density over the interval (0, 14). Describe the shape of this curve. Would it be appropriate to approximate this density with a normal curve?
2. Compute the posterior density on the grid of values of 0.1, 0.2, ..., 13.9, 14. Using these values, compute the posterior mean and posterior standard deviation. These will be good approximations to the actual posterior mean and posterior standard deviation.

3. Find a normal approximation to this posterior density. One way is to write a short function in R defining the logarithm of the posterior and using a function such as "'laplace()"' in the ProbBayes package to find the mean and standard deviation of the normal approximation.
4. Redraw the exact posterior density from part 1 and overlay the normal approximation curve. Comment on the accuracy of the normal approximation.
5. Compare the "exact" posterior mean and posterior standard deviation with the values found from the normal approximation.
6. Suppose you are interested in the posterior probability $P(\sigma > 8)$. Compute this probability two ways, one using the grid of values from part 2 and one using the normal approximation. Comment on the accuracy of the normal approximation.
7. Suppose you are interested in computing a 90% interval estimate of σ. Compute this interval two ways, one using the grid of values from part 2 and one using the normal approximation. Comment on the accuracy of the normal approximation.

1.4 Metropolis Activity "Random Walk on a Number Line"

Introduction

This activity illustrates the Metropolis algorithm for sampling from a probability mass function defined on a number line.

Helpful Apps

The "Metropolis Random Walk" app at `https://bayesball.shinyapps.io/Metropolis/` is used to show a Metropolis random walk for a probability distribution on a small set of integer values.

Description

Suppose we define the following probability mass function on the values 1, 2, 3, 4, 5. A graph of this probability mass function is displayed.

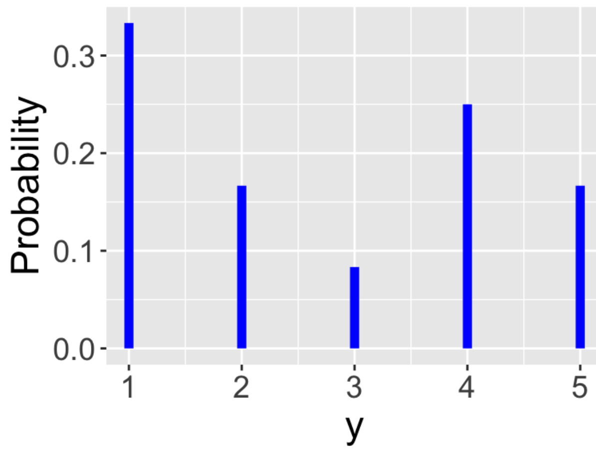

Figure 1: A probability mass function defined on five discrete values.

Suppose we sample from this distribution in the following way.

- Step 1. Start at any possible value of Y, say $Y = 2$. This is the Current location.
- Step 2. Flip a coin – if it lands heads, then the Candidate is the value one step to the left; if the coin lands tails the Candidate is the value one step to the right.
- Step 3. Compute the ratio

  $$R = \frac{weight(Candidate)}{weight(Current)}.$$

  (Note for ease of calculation, I am using the weights in the table rather than the actual probabilities to compute this ratio.)
- Step 4. Generate a random number U from 0 to 1. If the value of U is less than the ratio R, move to the Candidate location. Otherwise, remain at the Current location.

Repeat steps 2 through 4 many times. After many iterations, the relative frequencies of the visits to the five different values of y will approximate the probability distribution.

1. If you have a coin and a method for simulating a random number from 0 to 1, you can physically implement this random walk simulation. Starting at location $y = 2$, take 10 steps of this simulation, recording all of your visits.

2. A Shiny app for illustrating this random walk simulation can be found at `https://bayesball.shinyapps.io/Metropolis/`

   Using this app ...

   (a) Set the weights of your probability distribution to 4, 2, 1, 3, 2.

   (b) Set the number of iterations of the simulation to 100.

(c) This app will display a graph of the simulated locations as a function of the iteration number. Looking at this graph, what are some interesting features?

(d) The bottom graph in this app shows a histogram of the simulated locations. Set the number of iterations to 500. Is the shape of this histogram similar to the shape of the probability density? Is this what you would expect? Explain.

2 R Code for Section 2.3 Normal Approximation

The calculations for the two-group logistic model example are facilitated by use of the `LearnBayes` package (Albert, 2018). First we write a short function `logistic_posterior()` that computes the logarithm of the posterior density of $\boldsymbol{\beta} = (\beta_0, \beta_1)$ in Equation (**??**).

```
logistic_posterior <- function(theta, df){
  beta0 <- theta[1]
  beta1 <- theta[2]
  lp <- beta0 + beta1 * df$female
  p <- exp(lp) / (1 + exp(lp))
  sum(df$s * log(p) + df$f * log(1 - p)) +
    dcauchy(beta1, 0, 0.5, log = TRUE) +
    dnorm(beta0, 0, sqrt(1 / 0.0001), log = TRUE)
}
```

Suppose we observe $y_M = 8$ Facebook users in a sample of $n_M = 30$ men, and $y_W = 15$ Facebook users in a sample of $n_W = 30$ women. A data frame `ldata` is constructed that contains the numbers of successes (i.e. users), the numbers of failures (i.e. non-users), and the female indicator variables for the two groups. We find the normal approximation by use of the `laplace()` function[1]:

```
fit <- laplace(logistic_posterior, c(0, 0), ldata)
```

The object `fit` from `laplace()` function has several outputs. The vector `fit$mode` gives the posterior mode of the posterior and `fit$var` provides the estimate at the posterior variance-covariance matrix of the parameter vector.

For most of the Bayesian models considered at the undergraduate level including regression, the Laplace approximation is applicable and provides reasonable suitable approximations to the posterior distribution.

References

Albert, J. (2018) *LearnBayes: Functions for Learning Bayesian Inference*. URL: `https://CRAN.R-project.org/package=LearnBayes`. R package version 2.15.1.

[1]The `laplace()` uses the general-purpose `optim()` function from the `stats` package in R.