Posterior Predictive Distributions

STAT 341, SP23

2023-01-25

Contents

Definitions

- A model is **generative** if you can simulate hypothetical data samples from it. Many frequentist models, and basically all Bayesian ones, are generative.
- Obtaining a **posterior predictive distribution** means simulating data for all conjectured values of your parameter(s), then computing an average of all the simulated data distributions weighted by the posterior probability/density. In other words, *sample* parameter values from the posterior and simulate some data for each one, aggregating all the simulated data together at the end.

Question: Discuss why and how the two definitions of "posterior predictive distribution" are equivalent.

> They are both sampling parameter values from the posterior distribution, simulate new data for each sampled values, and combine them together.

Question: The figure below is Figure 3.6 from *Statistical Rethinking,* illustrating how to obtain a posterior predictive distribution. How would you label the y axis of each plot? What process is happening to get from the Posterior Probability to the Sampling Distributions? What process is happening to get from the Sampling Distributions to the Posterior Predictive Distribution?

> Y-axis,
>
> top: posterior probability density
>
> mid: simulated probability for each sample
>
> bottom: combined probability of water from simulated sample
>
> Sample each parameter value of the posterior probability make sampling distributions.
>
> Combine all of the sampling distribution to make the posterior predictive distribution.

Simulate Data Given Parameter

The code below is for the book's "proportion water on the Earth" example. Please change it so it's relevant to the "blood" example OR the "product patent" example.

```r
set.seed(3)
B <- rbinom(n = 10000, size = 9, prob = 0.33)
glimpse(B)
```

```
##  int [1:10000] 2 4 2 2 3 3 1 2 3 3 ...
```

Things to know and consider:

- (Replace the crossed-out text with an explanation that's true for your blood or patent example) Our model says that it's a fair approximation to say that observations of whether a randomly-chosen point on human body is blood or not blood are like draws from a binomial distribution, and we are trying to estimate the probability of any one "trial" or chosen point on human body being blood.
- The **posterior** gives us our "answer" – the estimate of the probability of blood.
- `rbinom()` is a function that produces *random samples* from a binomial distribution. The inputs are `n`, the number of random samples you want; `size`, the number of *trials* making up each sample; and `prob`, the probability of getting a "success."
- So... how did we choose the `n` to be 10,000? Why is the `size` 9? And how was 0.71 picked as the `prob`? *Hint: there's a problem with just using 0.71!)*

  n = 10000 because we want to have a large sample size that is enough for the simulation.
  size = 9 because there are 9 trials.
  0.33 is the probability of success, but we are missing the uncertainty.

- Why was the output called `W` and what might you call it in your (blood or patent) example?

  W is water, but for my example it would be B, blood.

Question: What's the problem with using a single value for `prob`, if our goal is to simulate data that are consistent with our model conclusions?

We will miss the *uncertainty*?

Posterior Predictive Distribution

To get the posterior predictive distribution, we do just as above to simulate data given a *single* parameter value. Except now, we do simulation for a *whole set of parameter values sampled from the posterior.*

The code below is for the book's "proportion water on the Earth" example. Alter it so it's relevant to the "blood" example OR the "product patent" example. Note, you may need to add code to this Rmd so that you have a fitted model to draw posterior samples from.

```
# add code here to: 1) fit your model and 2) generate post_samples, a sample from the posterior, from i
# uncomment this code when ready to run (it will not run until post_samples exists!)

blood_quap <- quap(
  alist(
    obs ~ dbinom(size = n, prob = p),  # binomial likelihood
    p ~ dunif(0, 1)        # **uniform** prior for the probability of success, "p"
  ),
  data = list(obs = 3, n = 9)
)

blood_data <-
  tibble(p_grid = seq(from = 0, to = 1, length.out = 20),       # define grid
         prior  = 1) |>                                        # define prior
  mutate(likelihood = dbinom(3, size = 9, prob = p_grid)) |>   # compute likelihood at each value in gri
  mutate(unstd_posterior = likelihood * prior) |>              # compute product of likelihood and prior
  mutate(posterior = unstd_posterior / sum(unstd_posterior))  # standardize the posterior,

n_sample <- 10000
post_samples <-
  blood_data |>
  slice_sample(n = n_sample,
               replace = TRUE,
               weight_by = posterior)  |>
  select(p_grid) |>
  pull(p_grid)


B_ppd <- rbinom(n = 10000, size = 9, prob = post_samples)
glimpse(B_ppd)
```

```
##  int [1:10000] 2 0 3 3 6 2 4 6 3 1 ...
```

Question: What is the difference between a *sample from the posterior* and a *posterior predictive distribution*? It might help to clearly state what units each one is measured in: probability, counts, etc.? Try including the words "data" and "parameter" and "model" in your answer.

> A *sample from the posterior* is the distribution of the model's parameter from the data.
>
> A *posterior predictive distribution* is the distribution predicting future data using samples from the posterior and model's parameter.

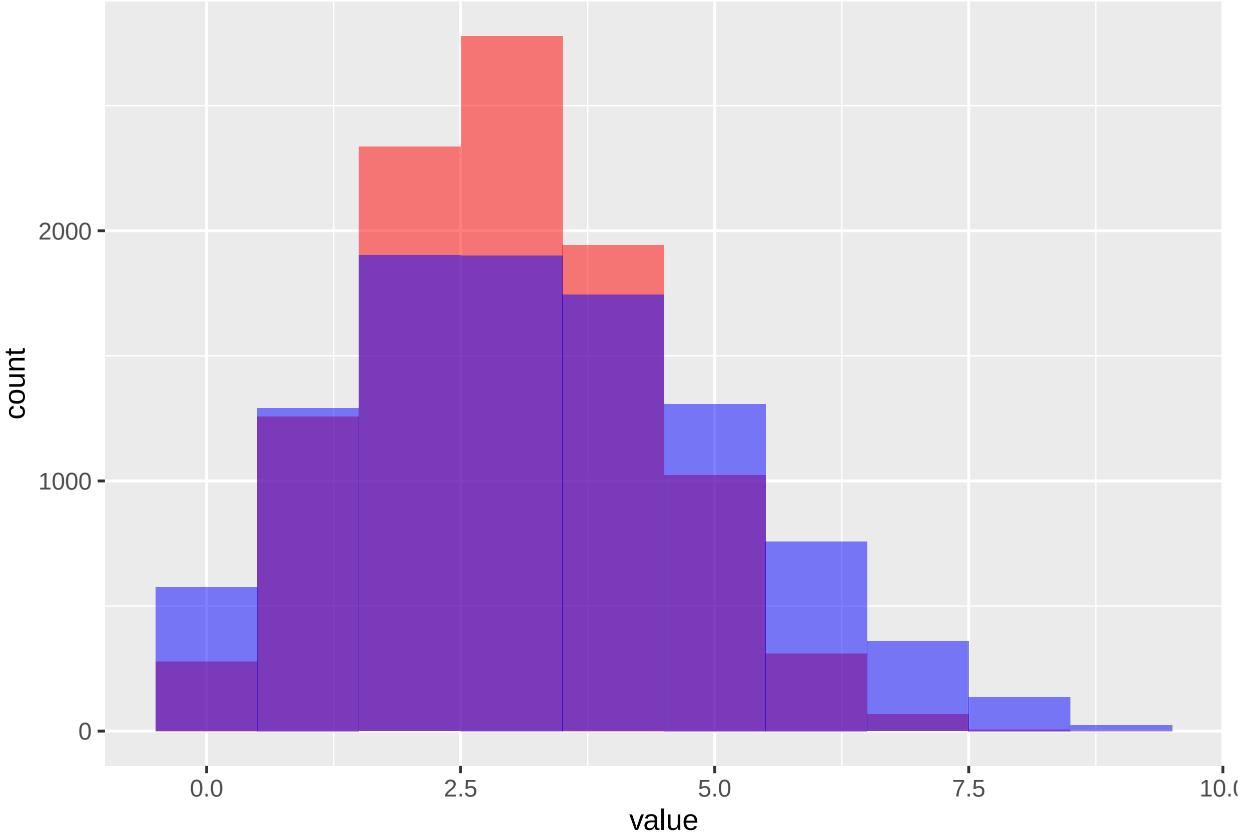
```


B_data <- tibble(value = B)
Bppd_data <- tibble(value = B_ppd)

gf_histogram(~ value, data = B_data, fill = "red", binwidth = 1) |>
  gf_histogram(~ value, data = Bppd_data, fill = "blue", binwidth = 1)
```

Question: How does `B_ppd` (the posterior predictive distribution) differ from the `B` you generated earlier using `rbinom()`? You can just explain based on your understanding so

far, or if you are not sure, try making graphs of `W` and `W_ppd` to help you figure it out.



what is wrong?. . .

Why, why, why?

Why would we want a posterior predictive distribution, anyway? There are a bunch of reasons, such as. . .

1. **Model design** We could simulate data like we did above, but using a sample *from the prior* instead of a sample from the posterior! This is called **prior predictive checking** and is a good way to see whether your choice of prior is realistic and mistake-free. If you think the data simulated from the prior is impossible or ridiculous, then *you* must know something that's not encoded in the prior (yet)!
2. **Model Checking** Does the posterior predictive distribution look more-or-less like real data? If not, you probably made a mistake in coding/fitting your model, choosing the prior, or conceptualizing the model!
3. **Software Validation** Beyond just checking if the posterior predictive distribution "looks okay," to further verify whether your model-fitting is working well, you can *generate* simulated data via the PPD and then *fit* the model to it. You should "recover" the parameter values used as input to the simulation, if everything is working right.
4. **Research Design** You can use simulated data to help plan future research and data collection. You can answer all kinds of practical questions such as, "Given how this system works and how we're modeling it, how big of a sample size would be needed to reliably estimate the difference between . . . and . . . or the effect of . . . on . . . ?"
5. **Forecasting**: You can "simulate new predictions, for new cases and future observations. These forcasts can be useful as . . . prediction, but also [to drive] model criticism and revision."

Question: Which of these items do you think we will do the most? If you had to list the five in order of importance, which ones would be at the top?

Model checking?

The item names on this list, and the quote in #5, were taken from SR Chapter 3, page 61.

Going Further (if extra time)

- Try to generate a **prior predictive distribution** for the model you have been working with.
- How does it compare with the posterior predictive distribution?
- Based on it, do you think the prior you used was a good choice?
- Try repeating this part (getting the prior predictive distribution) for almost-the-same model, but with a *different* prior. Options: try using a *better* prior, or try using a *terrible* one. By "terrible" I mean untrue and inconsistent with what you know about the situation; not one so invalid that it will just cause the model fitting machinery to throw an error!