STAT 447 Assignment 4

Caden Hewlett

2024-02-02

```
# integer for keeping seeds consistent across rmarkdown cells
seed_int = 447
```

Question 1 : Logistic Rocket Improvement

Recall the Rocket data from last week:

```
launches = c(1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1)
```

Recall that we discussed a model where the reliability of the rocket changes in time. This will allow us to incorporate, for example, the fact that engineering teams implement fixes based on past launches and therefore the probability of success should increase.

Part 1

Write a function called `logistic_regression` containing a `simPPLe` probabilistic programming description of the model described in class.

We recall that the model was of the following form, where ℓ is the launch and $\ell \in \{1, 2, \dots 11\}$.

For algebraic conciseness, we let slope $= \beta_1$ and intercept $= \beta_0$.

Therefore, by the reasoning dicsussed in class,

$$\begin{aligned} \beta_1 &\sim N(0,1) \\ \beta_0 &\sim N(0,1) \\ \theta(\ell) = \text{logistic}(\beta_1 \cdot \ell + \beta_0) &= \left(1 + \exp(-(\beta_1 \cdot \ell + \beta_0)\right)^{-1} \end{aligned}$$

Let's begin to design our function.

```
logistic_regression = function() {
  beta_1 = simulate(distr::Norm(0, 1))
  beta_0 = simulate(distr::Norm(0, 1))
  sapply(1:length(launches),
         function(L) {
           observe(launches[L],
```

```
                    Bern(plogis(beta_0 + beta_1 * L)))
        })
  next_p = plogis(beta_0 + beta_1 * 12)
  return(c(beta_0, beta_1, simulate(Bern(next_p))))
}
```

Your function should return a vector containing 3 elements in the following order:

The intercept, $\beta_0 \in \mathbb{R}$, the slope ($\beta_1 \in \mathbb{R}$), a prediction if one more launch would have been successful (1) or a failure (0) ($s \in \{0, 1\}$).

```
logistic_regression()
```

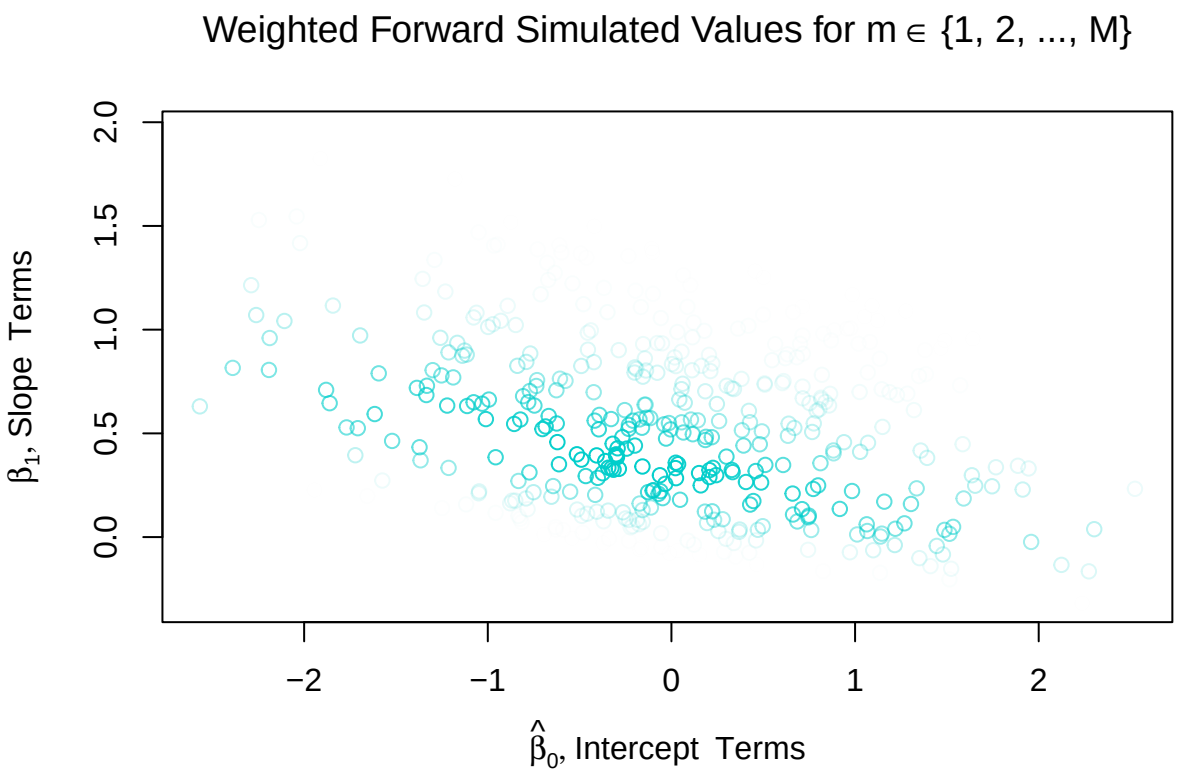
```
## [1] 0.587567 1.021064 1.000000
```

Part 2

Follow the instructions in the appendix below to get some helper functions. Use these functions to reproduce the lecture's bivariate posterior plot over the intercept and slope parameters.

The code to produce the bivariate posterior plot is below:

```
set.seed(seed_int)
post = posterior_particles(logistic_regression, 1000)
weighted_scatter_plot(post,
                      plot_options =
                        list(
    xlab = TeX(r'($\hat{\beta}_0, Intercept \; Terms$)'),
    ylab = TeX(r'($\hat{\beta}_1, Slope \; Terms$)'),
    main = TeX(r'(Weighted Forward Simulated Values for m\in\{1, 2, ..., M\})')
                        ))
```



Part 3

Estimate the probability that the next launch is a success given the data under the **logistic model**.

We know that the launches so far can be thought of as realizations of the random vector $\vec{Y} = Y_{1:11} = \{Y_1, Y_2, Y_3, \ldots, Y_{11}\}$. From these, we have the specific realizations (i.e. "our data") $\vec{y} = \{1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1\}$.

Note that there are $n = 11$ launches so far.

We wish to find:

$$P(Y_{12} = 1 \mid Y_1 = 1, Y_2 = 0, Y_3 = 1, \ldots Y_{11} = 1)$$

Or, in a simpler sense, using the variables defined earlier,

$$P(Y_{12} = 1 \mid Y_{1:12} = \vec{y})$$

In order to find this probability, all we need to do is add this outcome of Y_{12} and use our forward simulator.

Recall the third element of our random vector, predicting if the "next launch" would be a success. We'll denote this as $\mathcal{P}$.

$$\mathcal{P} = \mathbb{1}\big[\text{bern}\big(\text{logistic}(\beta_0 + 12\beta_1)\big) = 1\big]$$

Then, by running our forward simulator, we find the an *approximation* to the *posterior* expectation, i.e. approximating

$$\hat{G}_M \approx \mathbb{E}(\mathcal{P} \mid Y_{1:12} = \vec{y}) = \mathbb{E}\Big(\mathbb{1}\big[\text{bern}\big(\text{logistic}(\beta_0 + 12\beta_1)\big) = 1\big] \mid Y_{1:12} = \vec{y}\Big)$$

Recalling we're using $Y_{1:12}$ because we want to approximate the posterior expectation that the 12th launch is a success, meaning we have added 1 to the "observations" (like we did in the prediction lecture.)

Then, since we are approximating expectation of an indicator the event that the launch is a success, we are in turn computing the probability of the 12th is a success.

Hence, we can interpret the simulated expectation as the probability of the belief of the event that the next launch (Y_{12}) is a 1. In other words, the probability that the next launch is successful.

We do this whole process as follows:

```
set.seed(seed_int)
# add our theorized value for Y_{12}
launches = c(launches, 1)
# run the forward simulator
next_launch_post = posterior(logistic_regression, 1000)
# report our findings
data.frame(
  rbind(c("beta_0", "beta_1", "success"),
        round(next_launch_post, digits = 4))
)
```

```
##        X1     X2      X3
## 1  beta_0 beta_1 success
## 2 -0.1562 0.4399  0.9513
```

So, from this, we estimate $\mathbb{E}(\mathcal{P}) \approx 0.9513$, so the posterior probability of a successful Launch 12 given Launches 1 to 11 is around 95.13%. We can be pretty certain the next launch will be a success.

Part 4

Create a variant of the same model but where the **slope is set to zero**. Estimate the probability that the next launch is a success given the data under this simplified model.

Now, we repeat our methodology, considering the following setting:

$$\beta_0 \sim N(0, 1)$$
$$\theta(\ell) = \text{logistic}(\beta_0)$$

With the following function describing this process:

```
simplified = function() {
  beta_0 = simulate(distr::Norm(0, 1))
  sapply(1:length(launches),
         function(L) {
           observe(launches[L],
                   Bern(plogis(beta_0)))
         })
  next_p = plogis(beta_0)
  return(c(beta_0, simulate(Bern(next_p))))
}
simplified()
```

```
## [1] 0.5041439 1.0000000
```

Then, similar to last time, we let

$$\mathcal{S} = \mathbb{1}\big[\text{bern}\big(\text{logistic}(\beta_0)\big) = 1\big]$$

And use our forward-simulator to approximate:

$$\hat{G}_M \approx \mathbb{E}(\mathcal{S} \mid Y_{1:12} = \vec{y}) = \mathbb{E}\Big(\mathbb{1}\big[\text{bern}\big(\text{logistic}(\beta_0)\big) = 1\big] \mid Y_{1:12} = \vec{y}\Big)$$

Which gives us the posterior probability by the same logic discussed in the last question.

```
set.seed(seed_int)
# re-declare our data
launches = c(1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1)
# add our theorized value for Y_{12}
launches = c(launches, 1)
# run the forward simulator
next_launch_post = posterior(simplified, 1000)
# report our findings
data.frame(
  rbind(c("beta_0", "success"),
        round(next_launch_post, digits = 4))
)
```

```
##       X1      X2
## 1 beta_0 success
## 2 1.1253  0.7089
```

So the posterior probability of a successful Launch 12 given Launches 1 to 11 for the simplified model is around 70.89%. It's a bit more conservative than the full model.

Question 2 : Choosing a Model

You debate with your friend whether the logistic model or the simplified model (with slope equals to zero) should be preferred. To stop that debate, write a **unified** model which gives probability 1/2 to the simplified model, and 1/2 to the logistic model.

Estimate the posterior probability that the logistic model is preferred under the unified model given the same data as in Q.1.

Solution

Now, we will incorporate a new variable, in our model, which I will call ψ. We can think of ψ as the "unifier."

Our new variable takes values in $\{0, 1\}$, dictating whether we choose the Simplified Model ($\psi = 0$) or to include the Slope ($\psi = 1$.)

We begin by assuming each outcome is equally likely. To model this effectively, we let the prior on ψ be Bernoulli, with $p = 1/2$.

$$\begin{aligned}
\beta_1 &\sim N(0, 1) \\
\beta_0 &\sim N(0, 1) \\
\psi &\sim \text{bern}(1/2)
\end{aligned}$$

Then, our regression model is now of the form:

$$\theta(\ell) = \big(1 + \exp[-(\psi\beta_1\,\ell + \beta_0)]\big)^{-1}$$

Or, in a slightly nicer-to-read form:

$$\theta(\ell) = \text{logistic}\big((\beta_1 \cdot \psi \cdot \ell) + \beta_0\big)$$

Noting that $\psi = 0$ eliminates the β_1 term, as we would expect.

Let's recall some of the properties of the Bernoulli distribution. If $X \sim \text{bern}(p)$, where $P(X = 1) = p$, we know that :

$$\mathbb{E}(X) = \sum_{x \in \{0,1\}} xP(X = x) = (1 \cdot p) + 0 \cdot (1 - p) = p$$

In other words, its expectation is its parameter.

This will prove very useful for ψ, since by using an adjusted version of our forward simulator, we can compute its conditional expectation given the launches, $\mathbb{E}(\psi \mid Y_{1:11} = \vec{y})$.

This means, once we've run the simulator and approximate it's expectation this will give us the posterior of the parameter p. This follows the same logic of taking the expectation of an indicator on an event, and receiving the probability of that event. We established earlier that p is $P(\text{Choose Logistic})$ and shows no initial preferences.

So, we can combine our definition of ψ with the properties of Bernoulli random variables to write:

$$\mathbb{E}(\psi \mid Y_{1:11} = \vec{y}) = p_{\,\text{posterior}} = P(\text{ Choose Logistic } \mid Y_{1:11} = \vec{y}\,)$$

Yielding the posterior probability that the logistic model is preferred. *Note:* Alternatively, you can think of this as the posterior probability $P(\psi = 1 \mid Y_{1:11} = \vec{y})$.

Before we run the simulator, we should think a bit about what different values of $p_{\,\text{posterior}}$ represent.

Firstly, recalling $p_{\text{prior}} = 1/2$, we know:

$$\text{Outcomes} = \begin{cases} p_{\,\text{posterior}} \approx p_{\,\text{prior}} & \implies \text{No Preference} \\ p_{\,\text{posterior}} > p_{\,\text{prior}} & \implies \text{Prefer Logistic} \\ p_{\,\text{posterior}} < p_{\,\text{prior}} & \implies \text{Prefer Simplified} \end{cases}$$

In our posterior probability is roughly the same as our prior, the evidence of new launches has not added any significant weight towards one model type or another. If it's greater, our evidence has given support to the Logistic. Logically if it is less, our evidence has given support to the Simplified model.

The code for this unified model in PPL is below.

```
unified = function() {
  choose_log = simulate(Bern(1/2))
  beta_1 = simulate(distr::Norm(0, 1))
  beta_0 = simulate(distr::Norm(0, 1))

  sapply(1:length(launches),
         function(L) {
           observe(launches[L],
                   Bern(plogis(beta_0 + choose_log * beta_1 * L)))
         })
  next_p = plogis(beta_0 + beta_1 * 12)
  return(c(beta_0, beta_1, choose_log))
}
unified()
```

```
## [1] -0.8539497 -0.5796720  1.0000000
```

Now, we run the forward simulator to get the posterior values. It's interesting to note that we technically have a trivariate posterior here. I wonder what the plot of it would look like.

```
set.seed(seed_int)
## simulate the unified model's posterior
unified_post = posterior(unified, 1000)
data.frame(
  rbind(c("beta_0", "beta_1", "psi"),
        round(unified_post, digits = 4))
)
```

```
##       X1     X2     X3
## 1 beta_0 beta_1    psi
## 2 0.3553 0.2374 0.6444
```

So, the posterior probability that the logistic model is preferred under the unified model given the launch data is approximately 0.64444, or approximately 64.44%. This tells us that $p_{\text{posterior}} > p_{\text{prior}}$, implying that the complete logistic model is somewhat preferred to the simplified one.