Logistic Regression - Solutions

Setup

Exercise A - (3 min)

> One in a hundred women has breast cancer (B). If you have breast cancer, there is a 95% chance that you will test positive (T); if you do not have breast cancer (B^c), there is a 2% chance that you will nonetheless test positive (T). We know nothing about Alice other than the fact that she tested positive. How likely is it that she has breast cancer?

Solution

Let $T \equiv \{\text{Tests Positive}\},\ B \equiv \{\text{Has Breast Cancer}\}$. By Bayes' Theorem:

$$\begin{aligned} P(B|T) &= \frac{P(T|B)P(B)}{P(T)} = \frac{P(T|B)P(B)}{P(T|B)P(B) + P(T|B^C)P(B^C)} \\ &= \frac{0.95 \times 0.01}{0.95 \times 0.01 + 0.02 \times 0.99} \approx 0.32. \end{aligned}$$

Exercise B - (8 min)

1. Probabilities are between 0 and 1. What about odds? What about log odds?
2. If the probability of an event is $1/2$, what are the odds? What about the log odds? What does this tell us about the qualitative interpretation of odds and log odds?
3. I haven't given you a closed-form expression for `plogis()`. Use the log-odds representation of logit regression to work it out, then create a function called `myplogis()`. Compare to the "real" `plogis()` function on a grid of 400 points between -4 and 4.

Solutions

Part 1

Odds are between 0 and $+\infty$. Since $p \in [0,1]$ the ratio $p/(1-p)$ cannot be negative, but we can make it arbitrarily large by taking p close to one. Log odds are the log of something that is between zero and positive infinity, so they are between $-\infty$ and $+\infty$.

Part 2

A probability of 1/2 corresponds to odds of $(1/2)/(1-1/2) = 1$ and log odds of $\log(1) = 0$. Thus, odds of one mean "as likely as not," odds greater than one mean "more likely than not," and odds less than one mean "less likely than not." Log odds of zero mean "as likely as not," positive log odds mean "more likely than not," and negative log odds mean "less likely than not."

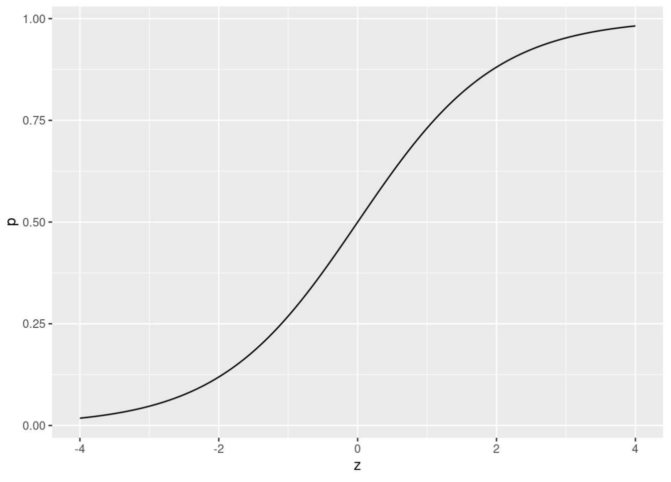
Part 3

Let $p \equiv P(Y=1|X)$. We are given $p = \texttt{plogis}(X'\beta)$ and $\log[p/(1-p)] = X'\beta$. Taking the exponential of both sides and re-arranging: $\exp(-X'\beta) = (1-p)/p$. Solving for p, we obtain $p = 1/[1+\exp(-X'\beta)]$. But since $p = \texttt{plogis}(X'\beta)$, $\texttt{plogis}(z) = 1/[1+\exp(-z)]$. Equivalently, we could multiply both the numerator and denominator by $\exp(z)$ to write $\texttt{plogis}(z) = \exp(z)/[1+\exp(z)]$.

```r
library(tidyverse)

myplogis <- function(z) {
  1 / (1 + exp(-z))
}

z_seq <- seq(from = -4, to = 4, length.out = 400)

all.equal(myplogis(z_seq), plogis(z_seq))
```

```
[1] TRUE
```

```r
plot_me <- tibble(z = z_seq, p = myplogis(z_seq))

ggplot(plot_me, aes(x = z, y = p)) +
  geom_line()
```



Exercise C - (8 min)

1. Consider two people: one has $X = x_2$ and the other has $X = x_1$. Under the simple logistic regression model from above, what is the *odds ratio* for these two people? In other words, what is the ratio of the odds that $Y = 1$ for person two relative to those for person one?
2. Perhaps you've heard of the "divide by four rule." It goes like this: if you divide a logistic regression coefficient by four, you'll obtain the *maximum possible* effect of changing the associated regressor by one unit on the predicted probability that $Y = 1$. Where does this rule come from?
3. The quantity $\beta \times \texttt{dlogis}(\alpha + \beta x)$ is usually called the partial effect of x. I write x to denote a realization of the random variable X, so x is constant whereas X is random. In contrast, the *average partial effect* is defined as $E[\beta \times \texttt{dlogis}(\alpha + \beta X)]$. and the *partial effect at the average* is $\beta \times \texttt{dlogis}(\alpha + \beta E[X])$. Do these two coincide in general? Why or why not? What question does each of them answer?

Solutions

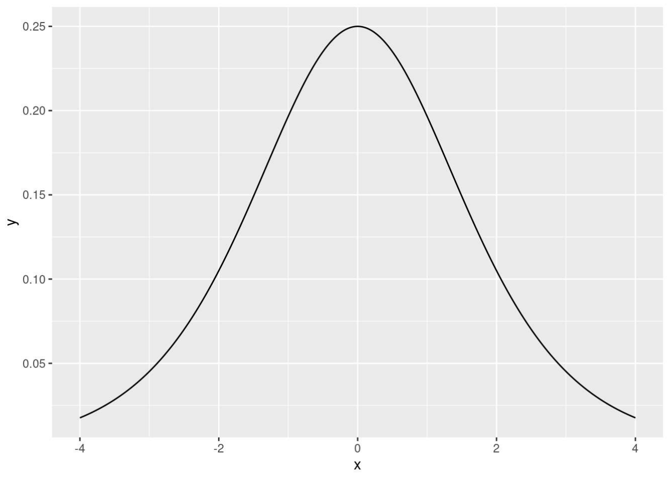
Part 1

The odds for person two are $\exp(\alpha + \beta x_2)$ while those for person one are $\exp(\alpha + \beta x_1)$. Taking the ratio of these gives $\exp\{\beta(x_2 - x_1)\}$.

Part 2

Above we found that the derivative of $\texttt{plogis}(\alpha + \beta x)$ with respect to x was $\beta \times \texttt{dlogis}(\alpha + \beta x)$. We also noted that `dlogis()` is the standard logistic density, so it can't be negative. This means that the sign of the effect equals the sign of β. The magnitude, however, depends on the value of $\texttt{dlogis}(\alpha + \beta x)$. To get the largest possible magnitude, we need to make `dlogis()` as large as possible. This density is symmetric with a mode at zero:

```r
x_seq <- seq(-4, 4, length.out = 400)
tibble(x = x_seq, y = dlogis(x_seq)) |>
  ggplot(aes(x, y)) +
  geom_line()
```



The "divide by four" rule arises because the height of the logistic density at zero is 1/4

```r
dlogis(0)
```

```
[1] 0.25
```

Part 3

They do not in general coincide because $E[f(X)]$ does not in general equal $f(E[X])$. (Linear/affine functions are the special case in which you *can* exchange E and f.) The average partial effect answers this question: "suppose I slightly increased `x` for everyone in my population of interest; how would my predicted probabilities that $Y = 1$ change on average?" The partial effect at the average answers a different question: "suppose I found someone whose `x` value equals the population mean; if I increased her `x` by a small amount, how would my predicted probability that $Y = 1$ change?"

Exercise C - (5 min)

My simulation code from above generated data from a logistic regression model using the "latent variable" approach. Write code to accomplish the same result using the *direct* approach.

Solution

```r
# These lines are identical to those from above so we get the same x-values:
set.seed(1234)
```

```r
n <- 500
alpha <- 0.5
beta <- 1
x <- rnorm(n, mean = 1.5, sd = 2)
# Here's the only thing that changes:
y <- rbinom(n, size = 1, prob = plogis(alpha + beta * x))
mydat2 <- tibble(x, y)
```

Exercise D - (10 min)

1. Construct approximate 95% confidence intervals for the parameters β_0 and β_1 based on the logistic regression output from above. Do your confidence intervals include the true parameter values that we used to simulate the data?
2. Interpret the estimated slope coefficient from `lreg`.
3. Try using `coef()` with `lreg`. What do you get? Does it work as expected? Now try the `broom` functions `tidy()` and `glance()` along with the function `modelsummary()` from the `modelsummary` package. What do you get?
4. As discussed above, β is *not* the derivative of $\texttt{plogis}(\alpha + \beta x)$ with respect to x.
   a. Use the "divide by 4" rule to calculate the *maximum* possible partial effect of x on the predicted probability that $Y = 1$ using the results of `lreg`.
   b. Calculate the partial effect of x on the predicted probability that $Y = 1$ at evaluated at the sample mean value of X (the partial effect at the average).
   c. Calculate the average partial effect of x over the observed sample.
   d. Compare your answers to (a), (b), and (c).

Solutions

Part 1

```r
# Setup code pasted from lecture slides
set.seed(1234)
n <- 500
x <- rnorm(n, mean = 1.5, sd = 2) # Generate X
ystar <- 0.5 + 1 * x + rlogis(n) # Generate "latent variable"
y <- 1 * (ystar > 0) # Transform to 0/1
mydat <- tibble(x, y)
lreg <- glm(y ~ x, family = binomial(link = 'logit'), mydat)
summary(lreg)
```

```

Call:
glm(formula = y ~ x, family = binomial(link = "logit"), data = mydat)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.3630     0.1344   2.700  0.00693 **
x             0.9638     0.1004   9.596  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 555.65  on 499  degrees of freedom
Residual deviance: 381.74  on 498  degrees of freedom
AIC: 385.74

Number of Fisher Scoring iterations: 6
```

The confidence interval for the regression intercept is approximately 0.36 $\pm$ 0.27 which includes the true value: $\alpha = 0.5$. Similarly, the confidence interval for the regression slope is 0.96 $\pm$ 0.2 which includes the true value: $\beta = 1$.

Part 2

Consider two people: Alice has a value of x that is one unit higher than Bob's value of x. Our model predicts that the log odds of $Y = 1$ are 0.96 higher for Alice than for Bob. Equivalently, we predict that Alice's odds of $Y = 1$ are a multiplicative factor of $\exp(0.96) \approx 2.6$ larger than Bob's odds. Whether this is a large or a small difference measured on the *probability scale* depends on Alice's specific value of x.

Part 3

```r
# They work as expected!
library(broom)
library(modelsummary)

tidy(lreg)
```

```
# A tibble: 2 × 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)    0.363     0.134      2.70 6.93e- 3
2 x              0.964     0.100      9.60 8.34e-22
```

```r
glance(lreg)
```

```
# A tibble: 1 × 8
  null.deviance df.null logLik   AIC   BIC deviance df.residual  nobs
          <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>       <int> <int>
1          556.     499  -191.  386.  394.     382.         498   500
```

```r
modelsummary(lreg)
```

	(1)
(Intercept)	0.363
	(0.134)
x	0.964
	(0.100)
Num.Obs.	500
AIC	385.7
BIC	394.2
Log.Lik.	−190.872
F	92.077
RMSE	0.35

Part 4

```r
# Divide by 4 rule
alpha_hat <- coef(lreg)[1]
beta_hat <- coef(lreg)[2]
beta_hat / 4
```

```
        x
0.2409399
```

```r
# Partial effect at average x
beta_hat * dlogis(alpha_hat + beta_hat * mean(x))
```

```
        x
0.1162997
```

```r
# Average partial effect
beta_hat * mean(dlogis(alpha_hat + beta_hat * x))
```

```
        x
0.1177498
```

Exercise E - (∞ min)

Data from the "two truths and a lie" experiment described at the beginning of these slides are available from `two-truths-and-a-lie-2022-cleaned.csv` in the `data` directory of my website: `https://ditraglia.com/data/`.

1. Read the dataset directly into R from the web, storing it in a tibble called `two_truths`.
2. Run a logistic regression that predicts `guessed_right` based on `certainty`. Make a nicely-formatted table of regression results using `modelsummary()` and comment briefly on your findings.
3. Use `ggplot()` to depict the regression from part 2, adding jittering to make the raw data clearly visible. You may need to adjust the `width` and `height` parameters of `geom_jitter()`.

Solutions

Part 1

```r
data_url <- 'https://ditraglia.com/data/two-truths-and-a-lie-2022-cleaned.csv'
two_truths <- read_csv(data_url)
```

Part 2

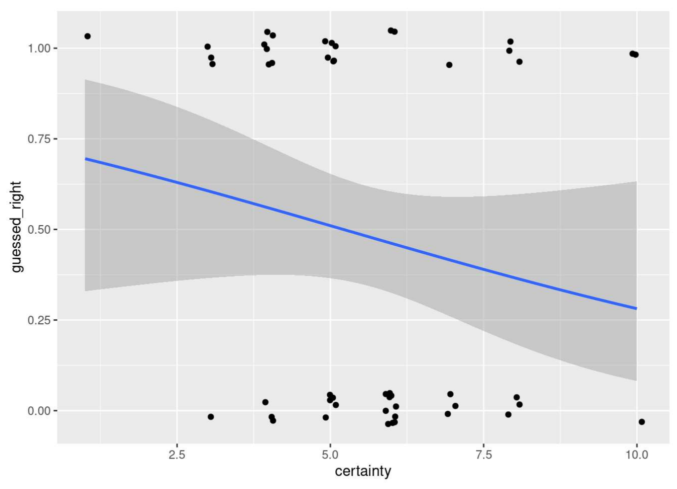
The estimated coefficient for `certainty` is *negative*! This means that on average we predict more wrong guesses as students' subjective certainty in their guesses increases. But notice that our estimate is fairly noisy: the coefficient estimate is around -0.2 and the standard error is around 0.16. We haven't found any compelling evidence that the relationship runs in either direction. Maybe the 2023 Core ERM students will be more obliging!

```r
two_truths_reg <- glm(guessed_right ~ certainty,
                      family = binomial(link = 'logit'),
                      data = two_truths)
library(modelsummary)
modelsummary(two_truths_reg)
```

	(1)
(Intercept)	1.020
	(0.932)
certainty	−0.196
	(0.159)
Num.Obs.	50
AIC	71.6
BIC	75.5
Log.Lik.	−33.814
F	1.524
RMSE	0.49

Solution - Part 3

```r
two_truths |>
  ggplot(aes(x = certainty, y = guessed_right)) +
  stat_smooth(method = 'glm', method.args = list(family = 'binomial'),
              formula = y ~ x) +
  geom_jitter(width = 0.1, height = 0.05)
```


guessed_right
1.00
0.75
0.50
0.25
0.00
2.5
5.0
7.5
10.0
certainty