Lesson 15 Boardsheet

Kevin Cummiskey

2/19/2020

Review

Last lesson, we discussed models for the probability of a called strike based on the location of the pitch when it crosses the plate.

1. Discuss two limitations of using a linear regression model in this situation.

2. Instead of using a linear regression model, we could use the following model:

$$\text{Strike}_i \sim \text{Bernoulli}(p_i)$$
$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 px_i + \beta_2 pz_i$$

a. By what name do we typically refer to this model?

b. Does this model address any limitations of linear regression?

3. Linear regression is fundamentally flawed as a model for strike probability because pitches near the middle of the strike zone are nearly always called strikes and pitches far away from the strike zone are nearly always called balls. A one inch difference in the middle of the plate is not as important as a one inch difference on the corner of the plate. Therefore, we want to consider more general models of the form:

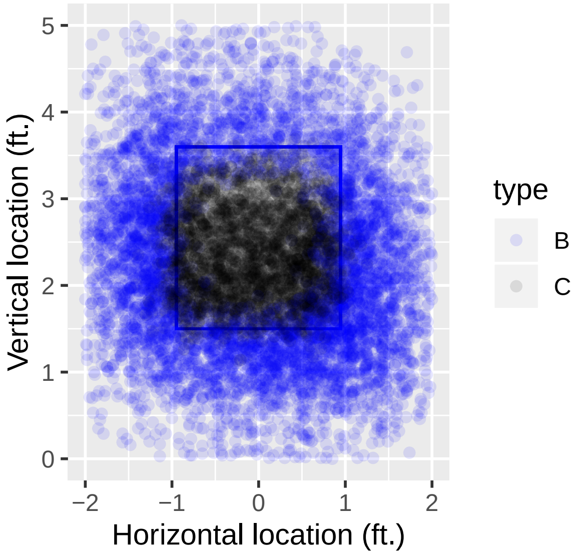
$$\text{Strike}_i \sim \text{Bernoulli}(p_i)$$
$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + f(px_i, pz_i)$$

In R, the `gam` function of the `mgcv` package will fit thin plate regression splines for $f(px, py)$. Deriving how to fit these splines by hand is outside the scope of this course. However, you should understand they represent a flexible way of modeling nonlinear relationships. In future analyses, we will use these models to adjust for pitch location when assessing the relationship between other variables (count, catcher, etc) and called strikes. For more information on thin plate regression splines, see https://www.mailman.columbia.edu/research/population-health-methods/thin-plate-spline-regression.

Modeling Called Strike Percentage

Let's look at some data. First, read in pitchf/x data using the `mlbgameday` package. I recommend you read in the data overnight and then save it locally on your computer in a file not being backed up to OneDrive.

```r
library(tidyverse)

#Run this code once
#library(mlbgameday)
#gamedat <- get_payload(start = "2016-05-01", end = "2016-05-31")
#pitches <- inner_join(gamedat$pitch, gamedat$atbat,
#                       by = c("num","url"))
#recommend writing to someplace on your harddrive
#pitches %>% write_csv("C:/Users/kevin.cummiskey/Data/pitches.csv")

#Read in the pitches you saved above
pitches <- read_csv(file = "C:/Users/kevin.cummiskey/Data/pitches.csv")

#we only want balls and called strikes
#note the types are different that in our text!!
taken <- pitches %>%
  filter(type %in% c("C","B"))

#Let's visualize the pitches
#if there is code you use a lot, you can put it in a separate file and
#source it.  The file kzoneplot contains the code on page 165 for creating
# the strike zone
source("kzoneplot.R")

k_zone_plot %+% sample_n(taken, 10000) +
  aes(color = type) +
  geom_point(alpha = 0.1) +
  scale_color_manual(values = c("blue", "black"))
```



Fit a model

Let’s fit the model in 3 to this data.

```r
library(mgcv)
strike_mod <- gam(type == "C" ~ s(px, pz),
                  family = binomial,
                  data = taken)
```

How can we use this model?

First, let's look at predicted probabilities of some pitches.

```
# a pitch right down the middle
predict(strike_mod,
        newdata = data.frame(px = 0, pz = 2.5),
        type = "response")
```

```
##         1
## 0.9992354
```

```
# a pitch on the inside corner for a right-handed batter
predict(strike_mod,
        newdata = data.frame(px = -1, pz = 2.5),
        type = "response")
```

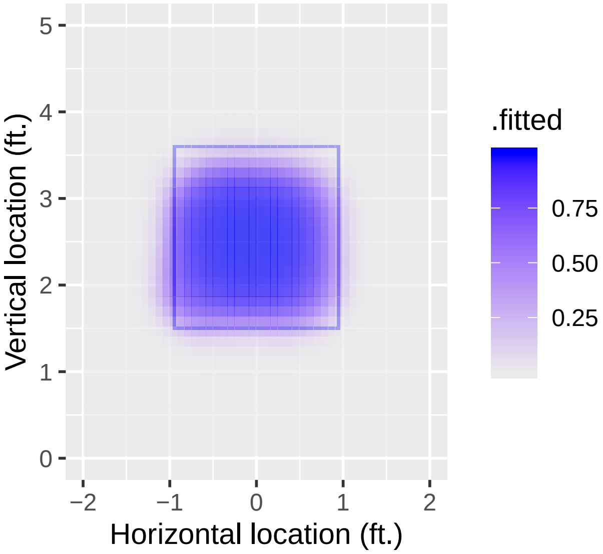
```
##         1
## 0.6193452
```

```
# a pitch on the outside corner for a right-handed batter
predict(strike_mod,
        newdata = data.frame(px = 1, pz = 2.5),
        type = "response")
```

```
##         1
## 0.2661631
```

Visualizing the estimated surface

```
library(modelr)
library(broom)
#create a grid of points in the strike zone
grid <- taken %>%
  data_grid(px = seq_range(px, n = 100),
            pz = seq_range(pz, n = 100))

#get predicted values from the model on the grid
grid_hats <- strike_mod %>%
  augment(type.predict = "response", newdata = grid)

#plot the results
tile_plot <- k_zone_plot %+% grid_hats +
  geom_tile(aes(fill = .fitted),alpha = 0.7) +
  scale_fill_gradient(low = "gray92", high = "blue")
tile_plot
```



Adjusting for batter handedness

We may suspect whether the batter is left-handed or right-handed (variable: `stand`) is important in determing whether a pitch is called a strike.

Let's fit a model that includes `stand`.

$$\text{Strike}_i \sim \text{Bernoulli}(p_i)$$

$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + f(px_i, pz_i) + \beta_1 stand_i$$

How do we intepret β_1 in this model?

```
hand_mod <- gam(type == "C" ~ s(px,pz) + stand,
                family = binomial,
                data = taken)
summary(hand_mod)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## type == "C" ~ s(px, pz) + stand
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.25567    0.27849 -15.281  < 2e-16 ***
## standR       0.17822    0.03456   5.158  2.5e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##            edf Ref.df Chi.sq p-value
## s(px,pz) 28.02  28.82  10738  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.734   Deviance explained = 70.1%
## UBRE = -0.61654  Scale est. = 1         n = 63697
```

How do we interpret these results?

Let's look at some pitches.

```
#LEFT CORNER STRIKE
#right-handed batter
predict(hand_mod,
        newdata = data.frame(px = -1, pz = 2.5, stand = "R"),
        type = "response")
```

```
##         1
## 0.6446487
```

```
#left-handed batter
predict(hand_mod,
        newdata = data.frame(px = -1, pz = 2.5, stand = "L"),
        type = "response")
```

```
##         1
## 0.6028551
```

```
#RIGHT CORNER STRIKE
#right-handed batter
predict(hand_mod,
        newdata = data.frame(px = 1, pz = 2.5, stand = "R"),
        type = "response")
```

```
##         1
## 0.2742944
```

```
#left-handed batter
predict(hand_mod,
        newdata = data.frame(px = 1, pz = 2.5, stand = "L"),
        type = "response")
```

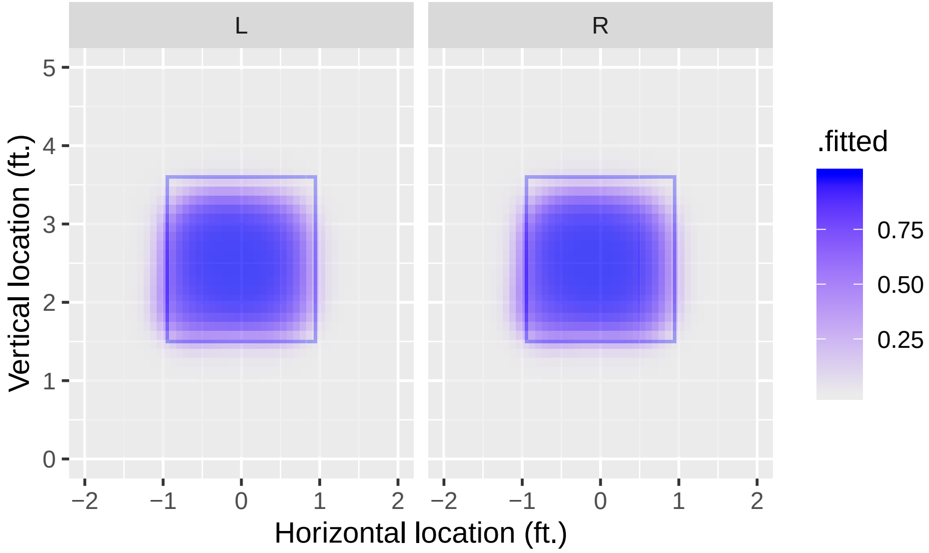
```
##         1
## 0.2402763
```

Let's visualize the results.

```
#create a grid of points
hand_grid <- taken %>%
  data_grid(px = seq_range(px, n = 100),
            pz = seq_range(pz, n = 100),
            stand)
#get model predictions on the grid
hand_grid_hats <- hand_mod %>%
  augment(type.predict = "response",
          newdata = hand_grid)
#plot predictions by handedness
```

```
tile_plot %+% hand_grid_hats +
  facet_grid(. ~ stand)
```

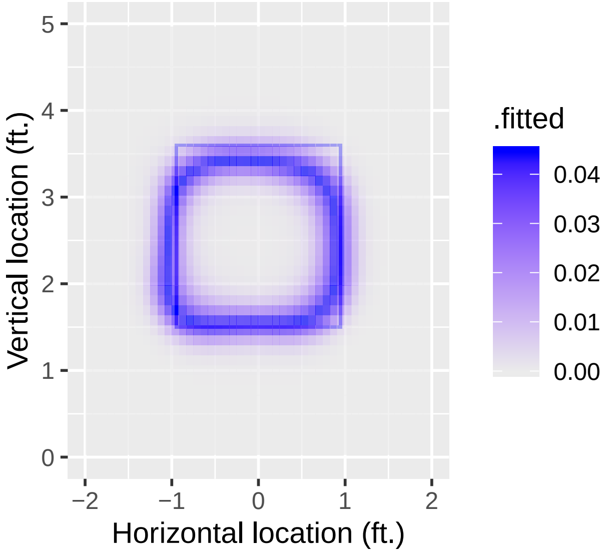
```
## Warning: Removed 15872 rows containing missing values (geom_tile).
```



Let’s see where the predictions differ the most.

```
#this code calculates the difference in predictions for
#left and right handed hitters at the same pitch location
diffs <- hand_grid_hats %>%
  group_by(px,pz) %>%
  summarize(N = n(), .fitted = diff(.fitted))
tile_plot %+% diffs
```

```
## Warning: Removed 7936 rows containing missing values (geom_tile).
```


Vertical location (ft.)
Horizontal location (ft.)
.fitted