Lab 10: Spline terms in regression models

Michael Lopez, Skidmore College

Overview

Spline terms are non-linear regression terms that are often able to more flexibly model the relationship between X and Y. Today, we'll implement a few spline models to better understand how they operate.

Baseball example

Let's return to our baseball example.

```
library(tidyverse)
library(Lahman)
head(Batting)

Batting_1 <- Batting %>%
  filter(yearID >= 1970, yearID <= 2000, AB >= 500) %>%
  mutate(X1B = H - X2B - X3B - HR,
         TB = X1B + 2*X2B + 3*X3B + 4*HR,
         RC = (H + BB)*TB/(AB + BB))

birth_years <- Master %>%
  select(playerID, birthYear)

Batting_2 <- Batting_1 %>% inner_join(birth_years)

Batting_2 <- Batting_2 %>%
  mutate(age = yearID - birthYear,
         age_sq = age^2)

ggplot(Batting_2, aes(x = age)) +
  geom_histogram(binwidth = 1)

ggplot(Batting_2, aes(x = age, y = RC)) +
  geom_line(aes(group = playerID)) +
  geom_point(aes(group = playerID)) +
  geom_smooth()
```

1. Describe the distribution of player age – it's center, shape, and spread
2. Roughly, describe the relationship between age and runs created. Additionally, identify one limitation of this data set with respect to answering this question.

Our interest lies in the assocation between age and runs created. Let's propose a few models. But instead of evaluating performance within a sample, let's split the data into training and testing data.

```
set.seed(0)
Batting_2$random_id <- rnorm(nrow(Batting_2))

training_data <- Batting_2 %>%
  filter(random_id < 0)

test_data <- Batting_2 %>%
```

```
  filter(random_id > 0)

dim(training_data)
dim(test_data)
dim(Batting_2)
```

3. Provide the primary benefits to evaluating models in data that the model is not fit on.

Let's try some models.

```
fit1 <- lm(RC ~ age, data = training_data)
fit2 <- lm(RC ~ age_sq + age, data = training_data)
fit3 <- lm(RC ~ ns(age, 4), data = training_data)
fit4 <- lm(RC ~ ns(age, 8), data = training_data)
```

4. As judged by AIC, which model above is best?

Let's see what predictions look like in a new data set. That is, how would our model do in the test data?

```
test_data <- test_data %>%
  mutate(RC_hat_fit1 = predict(fit1, test_data),
         RC_hat_fit2 = predict(fit2, test_data),
         RC_hat_fit3 = predict(fit3, test_data),
         RC_hat_fit4 = predict(fit4, test_data))

test_data %>% slice(1)
```

5. For the player listed above – Hank Aaron, in 1970 – which prediction was closest to his actual runs created?
6. Calculate the MSE and MAE for each of the four fits above, using the test data.
7. Use some baseball specific knowledge – what's definitely a variable needed to improve an age curve?

Splines with logistic regression

We return to our hockey data set to close out the lab. Recall, the shot data provided from the last two NHL seasons.

```
library(RCurl); library(tidyverse)
githubURL <- "https://raw.githubusercontent.com/statsbylopez/StatsSports/master/Data/pbp_data_hockey.rds
pbp_data <- readRDS(gzcon(url(githubURL)))
names(pbp_data)
```

Spline terms also operate well with logistic regression. In this case, we'll propose a few candidate models based on the distance of the shot.

```
library(splines)
pbp_data <- pbp_data %>%
  mutate(dist_sq = event_distance^2)
fit0 <- glm(event_type == "GOAL" ~ event_distance + dist_sq, family = "binomial", data = pbp_data)
fit1 <- glm(event_type == "GOAL" ~ ns(event_distance, 5), family = "binomial", data = pbp_data)
fit2 <- glm(event_type == "GOAL" ~ ns(event_distance, 10), family = "binomial", data = pbp_data)
```

8. Use AIC to pick your favorite model above.
9. Next, use Hosmer Lemeshow to identify if there is any lack of fit in `fit0`. Code for the HL test is provided below for `fit0`.

```r
pbp_data$shot_hat <- predict(fit0, pbp_data, type = "response")

tab_check <- pbp_data %>%
  mutate(shot_cat = cut(shot_hat, 10)) %>%
  group_by(shot_cat) %>%
  summarise(ave_exp_goals = sum(shot_hat),
            ave_act_goals = sum(event_type == "GOAL"),
            n_shots = n())

tab_check <- tab_check %>%
  mutate(diff_sq = (ave_exp_goals - ave_act_goals)^2/
           ((ave_exp_goals)))
tab_check

hm_test <- tab_check %>%
  summarise(test_stat = sum(diff_sq))
hm_test

1-pchisq(hm_test$test_stat, df = 8, lower.tail = TRUE)
```

10. Repeat question 9, using `fit1` and `fit2` (do separately). What does this suggest about the spline term?