Stats 50: Linear Regression Analysis of NCAA Basketball Data

April 8, 2016

Today we will analyze a data set containing the outcomes of every game in the 2012-2013 regular season, and the postseason NCAA tournament. Our goals will be to:

- Estimate the quality of each team via linear regression to obtain objective rankings
- Predict the winner and margin of victory in future games
- Get better at using `R` to manipulate and analyze data!

1. Loading in the data

First, read in the files `games.csv` and `teams.csv` into the data frames `games` and `teams`. You don't have to download these data files as they are hosted on the Internet; just read them in using the URLs below. We will load them "as is" (i.e. strings not converted to factors).

```
games <- read.csv("http://statweb.stanford.edu/~jgorham/games.csv",as.is=TRUE)
teams <- read.csv("http://statweb.stanford.edu/~jgorham/teams.csv",as.is=TRUE)
```

The `games` data has an entry for each game played, and the `teams` data has an entry for each Division 1 team (there are a few non-D1 teams represented in the `games` data). First, let's make one vector containing all of the team names, because the three columns do not perfectly agree. This will be useful later.

```
all.teams <- sort(unique(c(teams$team,games$home,games$away)))
```

Take some time and explore these two data files; for example, glance at the first few rows using `head`. Then try and answer the following questions:

Problems

1. How many games were played? How many teams are there?
2. Did Stanford make the NCAA tournament? What was its final AP and USA Today ranking?
3. How many games did Stanford play?
4. What was Stanford's win-loss record?

Solutions

1. We can answer these questions by calling `nrow` on both datasets:

```
nrow(teams); nrow(games)
```

```
## [1] 347
```

```
## [1] 5541
```

There are 347 teams and 5541 games played.

2. Let's take a look at the Stanford row in the `teams` dataset:

```
teams[teams$team == "stanford-cardinal", ]
```

```
##                   team conference inTourney apRank usaTodayRank
## 249 stanford-cardinal     pac-12         0     NA           NA
```

It appears Stanford failed to make the tournament in 2012-2013, and was unranked in both polls at the end of the season. :(

3. Let's pull out all the games in which Stanford played.and store them in a new data frame. Then we can check the length of that data frame.

```
games.stanford <- games[(games$home == "stanford-cardinal") |
                        (games$away == "stanford-cardinal"), ]
nrow(games.stanford)
```

```
## [1] 34
```

4. We need to identify the games that Stanford won. The `with` function is a useful trick so that I don't have to keep typing `games.stanford$` in front of every variable each time.

```
won <- with(games.stanford, ((home == "stanford-cardinal") & (homeScore > awayScore)) |
                            ((away == "stanford-cardinal") & (homeScore < awayScore)))
nrow(games.stanford[won, ])
```

```
## [1] 19
```

Stanford won 19 games, and since they played 34 games, their win-loss record was 19-15.

2. A Linear Regression Model for Ranking Teams

Now, let's try and use a linear regression model to determine which teams are better than others. The general strategy is to define a statistical model such that the parameters correspond to whatever quantities we want to estimate; in this case, we care about estimating the "quality" of each basketball team.

Our response variable for today is the margin of victory (or defeat) for the home team in a particular game. That is, define

$$y_i = (\text{home score} - \text{away score}) \text{ in game } i \tag{1}$$

Now, we want to define a linear regression model that *explains* the response, y_i, in terms of both teams' merits. The simplest such model will look something like

$$y_i = \text{quality of home}(i) - \text{quality of away}(i) + \text{noise} \tag{2}$$

where home(i) and away(i) are the home and away teams for game i. Keeping in mind the general strategy, this means that we want to define the coefficients β such that β_j represents the "quality" of team j. Now it

just remains the define the predictors X. To formulate this model as a linear regression in standard form, we need a definition for x_{ij} such that

$$y_i = \sum_j x_{ij}\beta_j + \varepsilon_i \tag{3}$$

How can we do this? A little clever thinking shows that we can define one predictor variable for each team, which is a sort of "signed dummy variable." In particular, for game i and team j, let

$$x_{ij} = \begin{cases} +1 & j \text{ is home}(i) \\ -1 & j \text{ is away}(i) \\ 0 & j \text{ didn't play.} \end{cases} \tag{4}$$

For example, if game i consists of team 1 visiting team 2, then $x_i = (-1, 1, 0, 0, \ldots, 0)$.

Now we can check that

$$\sum_j x_{ij}\beta_j = \beta_{\text{home}(i)} - \beta_{\text{away}(i)} \tag{5}$$

as desired, so the coefficient β_j corresponds exactly to the quality of team j in our model. Great! Now let's try and code this up.

Let's initialize a data frame `X0` with `nrow(games)` rows and `length(all.teams)` columns, filled with zeros. We also label the columns of `X0` accordingly. We call it `X0` for now, because we're going to need to make a slight modification in the next section before we can run the regression.

```
X0 <- as.data.frame(matrix(0,nrow(games),length(all.teams)))
names(X0) <- all.teams
```

Right now the matrix is just filled with zeros. You'll fill in the actual entries yourself below.

Problems

1. Create a vector `y` corresponding to the response variable as described in Equation (1) above.
2. Fill in `X0` column by column, according to Equation (4).

Solutions

1. We define the response vector `y`:

```
y <- games$homeScore - games$awayScore
```

2. Next we fill in `X0`:

```
## Fill in the columns, one by one
for(team in all.teams) {
  X0[,team] <- 1*(games$home==team) - 1*(games$away==team)
}
```

Now, we are in good shape, because our problem is formulated as a standard linear regression! Now we can use all of the usual tools to estimate β, make predictions, etc.

3. An Identifiability Problem

When we fit our model, we will ask `R` to find the best-fitting β vector. There is a small problem, however: for any candidate value of β, there are infinitely many other values $\tilde{\beta}$ that make **exactly** the same predictions. So the "best β" is not uniquely defined.

For any constant c, suppose that I redefine $\tilde{\beta}_j = \beta_j + c$. Then for every game i,

$$\tilde{\beta}_{\text{home}(i)} - \tilde{\beta}_{\text{away}(i)} = \beta_{\text{home}(i)} - \beta_{\text{away}(i)} \tag{6}$$

so the distribution of y is identical for parameters $\tilde{\beta}$ and β, no matter what c is. We can never distinguish these two models from each other, because the models make identical predictions no matter what. In statistical lingo, this is called an *identifiability* problem. It very often arises with dummy variables.

Intuitively, this problem occurs because our response is a "difference" of team performances, i.e. margin of scores in each game, and so the outcome only depends on the relative quality of two teams rather than the absolute qualities. If two very bad teams play against each other, we might expect the score differential to be 2 points, but if two very good teams play against each other, we might *also* expect the score differential to be 2 points.

To fix this problem, we can pick a "special" baseline team j and require that $\beta_j = 0$. We will take Stanford's team as the baseline; this has the additional benefit of allowing us to compare Stanford to other teams directly using the estimated coefficients.

Problem

Modify the `X0` matrix from the previous section to implement the above restriction. Name the modified matrix `X`.

(Actually, `lm` in `R` is smart enough to fix this automatically for you by arbitrarily picking one team to be the baseline. But let's not blindly rely on that, and instead do it ourselves, so we can better understand what `R` is actually doing.)

Solution

We can effectively force $\beta_j = 0$ for the j corresponding to Stanford by eliminating that column from the predictor matrix.

```
X <- X0[,names(X0) != "stanford-cardinal"]
```

[*Bonus Questions*] If you have time, consider the following questions:

- Suppose that we had chosen a different team as our baseline.
  - How would the estimates be different?
  - Would we obtain identical rankings?
  - Would we obtain identical standard errors?
- Under what circumstances would we still have an identifiability problem even after constraining $\beta_{\text{Stanford}} = 0$?

4. Fitting the model

Now, let's fit our model.

Problem

Fit the model using the `lm` function, regressing y on X with the below conditions. Recall that if you don't know exactly how to use the `lm` function, you should look at the documentation by calling `?lm`.

1. Use all the columns in the X matrix, and make sure to fit the model without an intercept. (*Hint:* The `formula` should look like "`y ~ 0 + .`". What do each of these parts mean?)
2. Only include regular season games in the model.

Explore the estimated coefficients using `summary`. What is the R^2 value?

Solution

```
reg.season.games <- which(games$gameType=="REG")
fit <- lm(y ~ 0 + ., data=X, subset=reg.season.games)
head(coef(summary(fit)))
```

```
##                            Estimate Std. Error     t value     Pr(>|t|)
## `air-force-falcons`      -6.687934   2.897768  -2.3079603 2.104205e-02
## `akron-zips`             -2.557678   2.841221  -0.9002039 3.680552e-01
## `alabama-a&m-bulldogs`  -30.590013   2.908702 -10.5167213 1.338001e-25
## `alabama-crimson-tide`   -2.851010   2.802443  -1.0173304 3.090456e-01
## `alabama-state-hornets` -29.958427   2.849296 -10.5143267 1.371671e-25
## `albany-great-danes`    -13.334555   2.818343  -4.7313458 2.291935e-06
```

```
summary(fit)$r.squared
```

```
## [1] 0.551235
```

It looks like our model explains about half of the variability in basketball scores.

5. Interpreting the Model

Now, let's try to interpret the model that we just fit.

Problems

1. Based on this model, what would be a reasonable point spread if Stanford played Berkeley (`california-golden-bears`)? What if Stanford played Louisville (`louisville-cardinals`) (that year's national champions)?
2. What would be a reasonable point spread if Duke (`duke-blue-devils`) played North Carolina (`north-carolina-tar-heels`)? If North Carolina played NC State (`north-carolina-state-wolfpack`)?
3. Does the dataset and model support the notion of home field advantage? How many points per game is it?

Solutions

1. Since $\beta_{\text{Stanford}} = 0$, the estimated coefficient $\hat{\beta}_{\text{Berkeley}}$ is exactly the estimated score difference for Berkeley against Stanford (and similarly for Louisville).

```
coef(fit)["`california-golden-bears`"]; coef(fit)["`louisville-cardinals`"]
```

```
## `california-golden-bears`
##                  -1.575168
```

```
## `louisville-cardinals`
##                11.94171
```

So we might predict Stanford to beat Berkeley by 2 points but lose to Louisville by 12 points. (Ignore the column name here; `R` just keeps the name of the first value input.)

2. The expected score difference is

$$\hat{\beta}_{\text{Duke}} - \hat{\beta}_{\text{UNC}} \tag{7}$$

So we can answer the question by comparing their coefficients:

```
coef(fit)["`duke-blue-devils`"] - coef(fit)["`north-carolina-tar-heels`"]
```

```
## `duke-blue-devils`
##            7.00085
```

Similarly,

```
coef(fit)["`north-carolina-tar-heels`"] - coef(fit)["`north-carolina-state-wolfpack`"]
```

```
## `north-carolina-tar-heels`
##                  0.05843945
```

So we might expect Duke to beat UNC by 7 points, and UNC to have a very slight edge over NC State.

3. We can assess home field advantage by including another column, which is 1 if and only if that game was played in a non-neutral location:

```
homeAdv <- 1 - games$neutralLocation
Xh <- cbind(homeAdv=homeAdv, X)
homeAdv.mod <- lm(y ~ 0 + ., data=Xh, subset=reg.season.games)
head(coef(summary(homeAdv.mod)), 1)
```

```
##         Estimate Std. Error  t value      Pr(>|t|)
## homeAdv 3.528499  0.1568782 22.49197 8.654579e-107
```

It looks like home field advantage is worth about 3.5 points per match and is clearly significant.