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?

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).

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.)

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?

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?