Stats 50

Week 5: Bradley-Terry models

1 Introduction

The Bradley-Terry model is used to form rankings from pair-wise comparisons (like matchups pitting one team or athlete against another). It simultaneously estimates how good each team is while adjusting the teams' results for how good their opponents are.

Example: 1995–1996 Hogwarts Inter-House Quidditch Cup

Hogwarts quidditch data are woefully incomplete. The Inter-House Quidditch Cup is a round-robin (each team plays each other team once) tournament in which the winner is determined by total number of points scored. We know the results of four of the six matches from the 1995–96 cup:

Gryffindor def. Slytherin, 200–20*
Hufflepuff def. Gryffindor, 240–230
Hufflepuff def. Ravenclaw, 230–210*
Gryffindor def. Ravenclaw, 190–40*

*Result known but exact score unknown. Most likely score inferred from Rowling JK (2003) *Harry Potter and the Order of the Phoenix*. For details, see `http://harrypotter.wikia.com/wiki/Inter-House_Quidditch_Cup`.

2 The normal Bradley-Terry model

2.1 Data

For game $i = 1, ..., n$;

- S_i: home score minus away score in game i
- H_i: identity of home team in game i
- A_i: identity of away team in game i

Example: 1995–1996 Hogwarts Inter-House Quidditch Cup

It would seem that in Hogwarts Inter-House Quidditch Cup matches there are no home teams. So let's treat the first team alphabetically as the home team.

$$n = 4 \qquad S_1 = 180 \qquad H_2 = \text{Gryffindor} \qquad A_3 = \text{Ravenclaw}$$

2.2 Model

$$S_i = \alpha + \beta_{H_i} - \beta_{A_i} + \epsilon_i \qquad \epsilon_i \overset{\text{i.i.d.}}{\sim} \text{Normal}(0, \sigma^2)$$

In matrix notation:

$$\mathbf{X} = \begin{pmatrix} 1 & X_{12} & X_{13} & ... & X_{1p} \\ 1 & X_{22} & X_{23} & ... & X_{2p} \\ 1 & X_{32} & X_{33} & ... & X_{3p} \\ ... & ... & ... & ... & ... \\ 1 & X_{(n-1)2} & X_{(n-1)3} & ... & X_{(n-1)p} \\ 1 & X_{n2} & X_{n3} & ... & X_{np} \end{pmatrix}_{n\times p} \qquad \text{where } X_{ij} = \begin{cases} 1 & \text{if } i^{th} \text{ home team is Team } j-1 \\ -1 & \text{if } i^{th} \text{ away team is Team } j-1 \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{y} = \begin{pmatrix} S_1 \\ S_2 \\ S_3 \\ ... \\ S_{n-1} \\ S_n \end{pmatrix}_{n\times 1} \qquad \beta = \begin{pmatrix} \alpha \\ \beta_1 \\ \beta_2 \\ ... \\ \beta_{p-1} \end{pmatrix}_{p\times 1}$$

Example: 1995–1996 Hogwarts Inter-House Quidditch Cup

$$\mathbf{X} = \begin{pmatrix} 0 & 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & -1 & 0 \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} 180 \\ -10 \\ 20 \\ 150 \end{pmatrix} \qquad \beta = \begin{pmatrix} \alpha \\ \beta_{\text{Gryffindor}} \\ \beta_{\text{Hufflepuff}} \\ \beta_{\text{Ravenclaw}} \\ \beta_{\text{Slytherin}} \end{pmatrix}$$

Note that we have made the first column of **X** all zero instead of all one. This reflects the fact that neither team is home in each match, hence no home-field advantage. We may as well omit this column (and the corresponding α parameter). Furthermore, there is an identifiability issue here. Adding the same constant to the coefficient corresponding to each team would result in the exact same score differential prediction. To resolve this, we must define the coefficient for one team to be exactly zero. The choice is arbitrary, so let's choose Slytherin.

$$\mathbf{X} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} 180 \\ -10 \\ 20 \\ 150 \end{pmatrix} \qquad \beta = \begin{pmatrix} \beta_{\text{Gryffindor}} \\ \beta_{\text{Hufflepuff}} \\ \beta_{\text{Ravenclaw}} \end{pmatrix}$$

2.3 Solution

$$\hat{\beta} = \arg\min \left\{ ||\mathbf{y} - \mathbf{X}\beta||^2 \right\} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{1}$$

Example: 1995–1996 Hogwarts Inter-House Quidditch Cup

$$\hat{\beta} = \left(\begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{pmatrix} \right)^{-1} \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & -1 \end{pmatrix} \begin{pmatrix} 180 \\ -10 \\ 20 \\ 150 \end{pmatrix} = \begin{pmatrix} 180.0 \\ 143.3 \\ 76.7 \end{pmatrix}$$

$$\begin{pmatrix} \hat{\beta}_{\text{Gryffindor}} \\ \hat{\beta}_{\text{Hufflepuff}} \\ \hat{\beta}_{\text{Ravenclaw}} \\ \hat{\beta}_{\text{Slytherin}} \end{pmatrix} = \begin{pmatrix} 180.0 \\ 143.3 \\ 76.7 \\ 0 \end{pmatrix}$$

2.4 Prediction

$$\hat{y} = \mathbf{X}\hat{\beta}$$

Example: 1995–1996 Hogwarts Inter-House Quidditch Cup

$$\hat{y} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} 180.0 \\ 143.3 \\ 76.7 \end{pmatrix} = \begin{pmatrix} 180.0 \\ 36.7 \\ 66.7 \\ 103.3 \end{pmatrix}$$

And if Ravenclaw were to play against Slytherin, according to this model Ravenclaw would be favored by $\hat{\beta}_{\text{Ravenclaw}} - \hat{\beta}_{\text{Slytherin}} = 76.7 - 0 = 76.7$ points.

3 The regularized normal Bradley-Terry model

3.1 Solution

Instead of solving equation (1) to estimate β, the regularized version of the model solves:

$$\hat{\beta}_\lambda = \arg\min \left\{ ||\mathbf{y} - \mathbf{X}\beta||^2 + \lambda||\beta||^2 \right\} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \tag{2}$$

for some $\lambda > 0$. The idea is that the penalty encourages the estimated β's to be close to zero. Note that the identifiability issue of the Bradley-Terry model is resolved because equation (2) has a unique solution for any $\mathbf{X}$. Furthermore, it can be shown that the average $\hat{\beta}_j$ in the unique solution is zero. Hence encouraging the estimated β's to be close to zero can be interpreted as doing something similar to regression to the mean. To choose the appropriate value for λ, we use a procedure called cross validation (see Section 6).

3.2 R

To fit the regularized normal Bradley-Terry model in R, you will need to have installed (the first time) and loaded (every time) the `glmnet` package. To fit the model with cross validation to choose λ, use the `cv.glmnet()` function:

```
> model = cv.glmnet(X, y, alpha = 0, intercept = FALSE, standardize = FALSE)
```

To use the model to predict score differentials, use the `predict()` function:

```
> pred = predict(model, X, s = 'lambda.min')
```

To extract the estimated β's from the model, it is easiest to again use the `predict()` function (and the fact that if $\mathbf{X} = \mathbf{I}$, then the prediction is $\hat{y} = \mathbf{I}\hat{\beta} = \hat{\beta}$):

```
> beta = predict(model, diag(ncol(X)), s = ‘lambda.min’)
```

Note that the `diag()` function returns the identity matrix $\mathbf{I}$.

4 The binomial Bradley-Terry model

4.1 Data

$$W_i = \begin{cases} 1 & \text{if the home team won game } i \\ 0 & \text{if the away team won game } i \end{cases}$$

Example: 1995–1996 Hogwarts Inter-House Quidditch Cup

$$W_4 = 1$$

4.2 Model

$$P(W_i = 1) \equiv p_i = \frac{e^{\alpha + \beta_{H_i} - \beta_{A_i}}}{1 + e^{\alpha + \beta_{H_i} - \beta_{A_i}}}$$

4.3 Solution

$$\hat{\beta} = \arg\max \prod_{i=1}^{n} \left(p_i^{W_i} \cdot (1 - p_i)^{1 - W_i} \right) = \arg\min - \sum_{i=1}^{n} \left(W_i \log p_i + (1 - W_i) \log(1 - p_i) \right) \tag{3}$$

The optimization problem (3) is solved using gradient descent.

5 The regularized binomial Bradley-Terry model

5.1 Solution

Instead of solving (3), solve:

$$\hat{\beta}_\lambda = \arg\min \left\{ - \sum_{i=1}^{n} \left(W_i \log p_i + (1 - W_i) \log(1 - p_i) \right) + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}$$

5.2 R

In R, we use the same function (`cv.glmnet()`) to fit the model, but we specify `family = ‘binomial’`:

```
> model = cv.glmnet(X, y, alpha = 0, intercept = FALSE,
+ standardize = FALSE, family = ‘binomial’)
```

Predictions are obtained similarly, but we specify `type = ‘response’` to get probabilities:

```
> pred = predict(model, X, s = ‘lambda.min’, type = ‘response’)
```

To extract $\hat{\beta}$, **do not** specify `type = 'response'`:

```
> beta = predict(model, diag(ncol(X)), s = 'lambda.min')
```

6 Cross validation

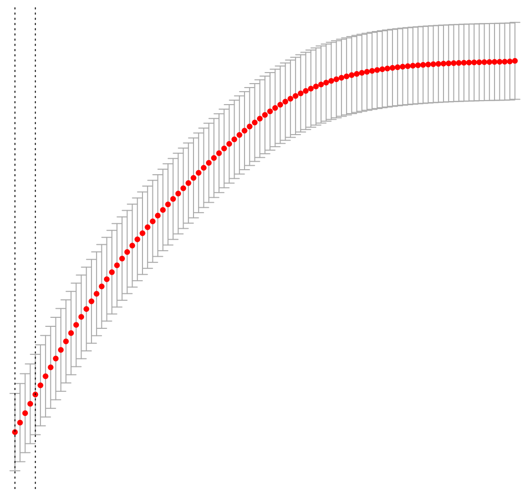
1. Randomly partition data into 10 pieces.
2. For piece $k = 1, ..., 10$:
   (a) Using all data except piece k, solve equation (2) for many different values of λ.
   (b) Use the result for each value of λ to predict the score differentials for piece k.
   (c) For each λ, assess the accuracy of the predictions using total squared error.
3. For each λ, add up the total error across the k pieces. Choose the λ with the smallest error.

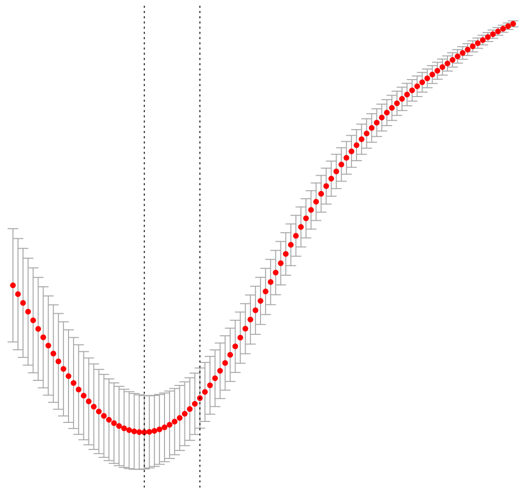
6.1 R

The `cv.glmnet()` function takes care of cross validation for you. The one thing you want to check is that the default range of λ values is sufficient. To view a diagnostic plot, use the `plot()` function:

```
> plot(model)
```

This plot shows cross validation error as a function of λ. What you want to see is a curve that decreases and then increases, achieving its minimum value some where in the interior of the range of λ values considered. If the curve is monotonically increasing or decreasing, that suggests that the optimal value of λ is outside of the range of values that you considered, and you need to specify a different sequence of values using the `lambda` argument to `cv.glmnet()`. Examples of a bad diagnostic plot and a good diagnostic plot are below.


Example of bad λ range


Example of good λ range