Stats 50

Week 6: Rasch models

1 Introduction

The Rasch model is just like the Bradley-Terry model, but the difference is that the adversaries do not appear on both sides of the matchup. In the Bradley-Terry model, each adversary is sometimes the home team and sometimes the away team. But in the Rasch model, each adversary is only ever on offense or only ever on defense. Whereas the Bradley-Terry model is more appropriate for making team rankings, the Rasch model is more appropriate for separately estimating the offensive and defensive skills of individual players.

2 The (regularized) normal Rasch model

Example: Offensive-Defensive Plus-Minus for NBA player evaluation

The following model is very close to the (non-SportVU) state of the art, ESPN's Real Plus-Minus, for evaluating the contributions of NBA players to their teams' chances of winning. For each possession over the course of the nba season, we observe the five offensive players on the floor, the five defensive players on the floor and whether the offensive team is the home team, as well as the number of points scored on the possession (typically 0, 1, 2 or 3).

Note that this is a special version of the Rasch model that includes five offensive adversaries and five defensive adversaries for each matchup, instead of just one each.

2.1 Data

For possession $i = 1, ..., n$:

- For $j = 1, ..., 5$:
  - · O_{ij} = identity of j^{th} offensive player on i^{th} possession
  - · D_{ij} = identity of j^{th} defensive player on i^{th} possession
- H_i = indicator of whether offense is home team on i^{th} possession
- P_i = number of points scored on i^{th} possession

Throughout Section 2 we will use p to denote the number of unique players in our dataset, and we will assume that each player appears on both offense and defense.

2.2 Model

$$\mathbb{E}[P_i] = \alpha + \sum_{j=1}^{5} \beta_{O_{ij}} + \sum_{j=1}^{5} \delta_{D_{ij}} + \theta H_i$$

Example: Offensive-Defensive Plus-Minus for NBA player evaluation

The Golden State Warriors have the ball on the road against the Portland Trail Blazers. The players on the floor and the relevant coefficients are:

$$\begin{aligned} \alpha &= 1.00 & \theta &= +.03 \\ \beta_{\text{Stephen Curry}} &= +.04 & \delta_{\text{Damian Lillard}} &= -.03 \\ \beta_{\text{Klay Thompson}} &= +.02 & \delta_{\text{C.J. McCollum}} &= -.02 \\ \beta_{\text{Harrison Barnes}} &= +.03 & \delta_{\text{Al-Farouq Aminu}} &= -.01 \\ \beta_{\text{Draymond Green}} &= +.02 & \delta_{\text{Maurice Harkless}} &= -.01 \\ \beta_{\text{Andrew Bogut}} &= -.01 & \delta_{\text{Mason Plumlee}} &= +.02 \end{aligned}$$

The expected number of points per possession in this situation is

$$1.00 + .04 + .02 + .03 + .02 - .01 - .03 - .02 - .01 - .01 + .02 = \boxed{1.05}.$$

The β's represent the offensive skills of the players, with larger being better. The δ's represent the defensive skills of the players, with smaller (i.e. more negative) being better.

In matrix notation:

$$\mathbf{X} = \begin{pmatrix} \overbrace{X_{11} \quad \ldots \quad X_{1p}}^{\text{Offense}} & \overbrace{X_{1(p+1)} \quad \ldots \quad X_{1(2p)}}^{\text{Defense}} & H_1 \\ X_{21} \quad \ldots \quad X_{2p} & X_{2(p+1)} \quad \ldots \quad X_{2(2p)} & H_2 \\ X_{31} \quad \ldots \quad X_{3p} & X_{3(p+1)} \quad \ldots \quad X_{3(2p)} & H_3 \\ \ldots \quad \ldots \quad \ldots & \ldots \quad \ldots \quad \ldots & \ldots \\ X_{(n-1)1} \quad \ldots \quad X_{(n-1)p} & X_{(n-1)(p+1)} \quad \ldots \quad X_{(n-1)(2p)} & H_{n-1} \\ X_{n1} \quad \ldots \quad X_{np} & X_{n(p+1)} \quad \ldots \quad X_{n(2p)} & H_n \end{pmatrix}_{n\times(2p+1)}$$

where for $j = 1, ..., p$:

$$X_{ij} = \begin{cases} 1 & \text{if player } j \text{ is on } \textbf{offense} \text{ on possession } i \\ 0 & \text{otherwise} \end{cases}$$

and for $j = p+1, ..., 2p$:

$$X_{ij} = \begin{cases} 1 & \text{if player } j-p \text{ is on } \textbf{defense} \text{ on possession } i \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{y} = \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ \ldots \\ P_{n-1} \\ P_n \end{pmatrix}_{n\times 1} \qquad \beta = \begin{pmatrix} \beta_1 \\ \ldots \\ \beta_p \\ \delta_1 \\ \ldots \\ \delta_p \\ \theta \end{pmatrix}_{p\times 1}$$

2.3 Solution

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

2.4 Prediction

$$\hat{y}_\lambda = \hat{\alpha}\mathbf{1}_n + \mathbf{X}\hat{\beta}_\lambda$$

2.5 R

As with the (regularized) normal Bradley-Terry model, we will fit the (regularized) normal Rash model in R by using the `cv.glmnet()` function from the R package `glmnet`. Unlike the B-T model, we will allow `glmnet` to fit its own intercept, which it does by default. This means we will NOT specify `intercept = FALSE`.

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

To extract the estimated regression coefficients from the model, use the `coef()` function:

```
> beta = coef(model, s = 'lambda.min')[, 1]
```

To use the model to predict points per possession, use the `predict()` function:

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

3 The (regularized) binomial Rasch model

Example: Batting average based on batter-pitcher matchup in baseball

3.1 Data

For at bat $i = 1, ..., n$:

- B_i = identity of batter in i^{th} at bat
- P_i = identity of pitcher in i^{th} at bat
- y_i = indicator of whether i^{th} at bat results in hit

3.2 Model

$$P(y_i = 1) = p_i = \frac{e^{\alpha+\beta_{B_i}+\delta_{P_i}}}{1+e^{\alpha+\beta_{B_i}+\delta_{P_i}}}$$

Example: Batting average based on batter-pitcher matchup in baseball

Mike Trout bats against Clayton Kershaw. The relevant regression coefficients are:

$$\alpha = -1.10 \qquad \beta_{\text{Mike Trout}} = +0.25 \qquad \delta_{\text{Clayton Kershaw}} = -0.30$$

Trout's expected batting average against Kershaw is

$$\frac{e^{-1.10+0.25-0.30}}{1+e^{-1.10+0.25-0.30}} = \frac{e^{-1.15}}{1+e^{-1.15}} = \boxed{.240}.$$

3.3 Solution

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

3.4 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, standardize = FALSE, family = ‘binomial’)
```

To extract $\hat{\beta}$, again use `coef()`:

```
> beta = coef(model, s = ‘lambda.min’)[, 1]
```

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

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

3.5 Extending the model

The model could be extended to account for such factors as ballpark, home-field advantage and whether the batter and pitcher have opposite handedness:

- S_i = identity of stadium in i^{th} at bat
- H_i = indicator of whether batter's team is home in i^{th} at bat
- O_i = indicator of whether batter, pitcher have opposite handedness in i^{th} at bat

Model:

$$p_i = \frac{e^{\eta_i}}{1+e^{\eta_i}} \qquad \text{where} \qquad \eta_i = \alpha + \beta_{B_i} + \delta_{P_i} + \tau_{S_i} + \theta H_i + \zeta O_i$$

4 Sparse matrix representation

As we move from modelling game results to modelling the outcomes of matchups between individual players, we can expect to be dealing with larger datasets. For both the examples presented in these notes, if we were to fit the model to a whole season's worth of data, we can expect hundreds of thousands of matchups featuring close to one thousand unique players. This presents a computational challenge.

Our matrix X will have over one hundred million entries, but 99% or more of those entries are zero, with the rest being one. We say that X is “sparse”. In R, we can leverage this fact to make the computations much faster, using the `sparseMatrix()` function in the package `Matrix`. This function takes two arguments, `i` and `j` which are vectors of the same length, and builds a sparse matrix filled with zero everywhere, except the entries specificied by the pairs (i, j). From here, `X` can be used in `cv.glmnet()` just like a matrix that had been constructed without sparse representation.

Example: Batting average based on batter-pitcher matchup in baseball

Suppose we observe the following matchups between batters and pitchers:

Batter	**Pitcher**
Andrew McCutchen	Dallas Keuchel
Nelson Cruz	Dallas Keuchel
Buser Posey	Dallas Keuchel
Jose Altuve	Gerritt Cole
Nelson Cruz	Gerritt Cole
Buster Posey	Gerritt Cole
Jose Altuve	Felix Hernandez
Andrew McCutchen	Felix Hernandez
Buster Posey	Felix Hernandez
Jose Altuve	Madison Bumgarner
Andrew McCutchen	Madison Bumgarner
Nelson Cruz	Madison Bumgarner

Then our matrix

$$X = \begin{array}{cccccccc} JA & AM & NC & BP & DK & GC & FH & MB \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}$$

can be constructed as a sparse matrix with

```
> i = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
> j = c(2, 3, 4, 1, 3, 4, 1, 2, 4, 1, 2, 3, 5, 5, 5, 6, 6, 6, 7, 7, 7, 8, 8, 8)
> X = sparseMatrix(i, j)
```

Because the pairs $(1,2),(2,3),(3,4),(4,1),...,(12,8)$ represent the nonzero entries of X.