

# Lab: Regularization & Ridge Regression

1. APM & RAPM to measure each NBA player's offensive and defensive "Impact"

We can measure player offensive skill fairly well using box score data, but the box score doesn't tell the full story (see: Westbrook) and won't tell us which role players are impactful or even whose good defensively for non-bigs.

A statistician wants to turn to statistical modeling to measure each NBA player's impact.

This motivates the Adjusted Plus Minus (APM) Model:

Variables  $i$  = index of the  $i^{\text{th}}$  possession in the dataset

$OPI(i), \dots, OPS(i)$  are the 5 offensive players of possession  $i$

$DPI(i), \dots, DPS(i)$  are the 5 defensive players of possession  $i$

$Y_i$  = outcome (points) of the  $i^{\text{th}}$  possession

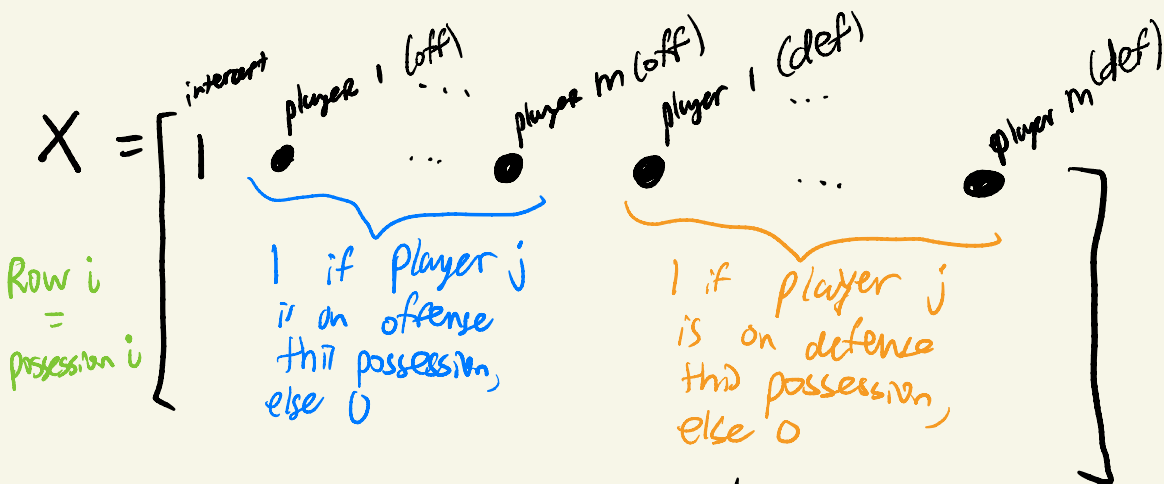
parameters Each player  $j=1, \dots, m$  gets an offensive strength parameter  $\beta_j$  and a defensive strength parameter  $\gamma_j$ . Intercept is  $\alpha_0$ .

Model

$$y_i = \alpha_0 + \beta_{OP1(i)} + \beta_{OP2(i)} + \dots + \beta_{OP5(i)} - \gamma_{DP1(i)} - \gamma_{DP2(i)} - \dots - \gamma_{DP5(i)} + \epsilon_i$$

$\epsilon_i$  is mean 0 noise.

Matrix vector notation



$$y = X^T \beta + \epsilon$$

$$y_i = x_i^T \beta + \epsilon_i$$

$$x_i = i^{th} \text{ row of } X$$

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

$$\beta = \begin{pmatrix} \alpha_0 \\ \beta_1 \\ \vdots \\ \beta_m \\ \gamma_1 \\ \vdots \\ \gamma_m \end{pmatrix}$$

$$\epsilon = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

If you use OLS to fit the coefficients  $\beta, \delta$  you get Adjusted Plus Minus (APM).

This will overfit, particularly for players with few possessions, and due to extreme **Multicollinearity** (teammates who almost always play together).

**Regularized** Adjusted Plus Minus (RAPM)

fits this model using **Ridge Regression** (glmnet in R) to shrink the coefficients towards zero.

\* **Your task:** fit APM and RAPM for one season's worth of possessions, visualize and compare.

You'll need to manually make the  $X$  matrix!

# Ridge Regression Code in R:

```
m1 = -3; m2 = 3; lambdas = 10^(seq(m1,m2,by=0.2));  
lambdas  
ridge_model = cv.glmnet(  
  x = X_rapm_j, y = y_j, XXXXXXXXXX nfold = 5, alpha = 0, family="gaussian", lambda = lambdas, standardize=FALSE  
)  
lambda = ridge_model$lambda.min  
plot(ridge_model)  
print(paste0("lambda = ", lambda))
```

automatically tunes  
lambda over a grid  
by cross validation

make sure  
to include  
this so as  
to not  
standardize  
the X  
matrix!!  
glmnet  
default is to  
standardize