Lab 5: Comparing Ridge, Lasso, and OLS

Kaori Hirano

2023-06-07

Packages

```
# load packages here
library(ISLR2)
library(leaps) # for best subset selection
suppressPackageStartupMessages(library(glmnet)) # for ridge, LASSO, and elastic net
suppressPackageStartupMessages(library(tidyverse))
suppressPackageStartupMessages(library(readr))
library(patchwork) # for plot arrangement
```

Data

```
mls22 <- read_csv("data/mls22.csv", show_col_types = FALSE)
d <- mls22 |> drop_na()
```

Data Wrangling

Q1

```
d <- d |>mutate(Nation, usa = if_else(Nation == "USA", 1, 0)) # 0 for non us, 1 for us

# 1 for gk, else 0
gkval <- grepl("GK", d$Pos, fixed = TRUE)
d <- d |> mutate(Pos, gk = if_else(gkval == TRUE, 1, 0))
```

```
# 1 for df, else 0
dfval <- grepl("DF", d$Pos, fixed = TRUE)
d <- d |> mutate(Pos, df = if_else(dfval == TRUE, 1, 0))
# head(d) works!

# 1 if mf, 0 if not
mfval <- grepl("MF", d$Pos, fixed = TRUE)
d <- d |> mutate(Pos, mf = if_else(mfval == TRUE, 1, 0))

# 1 for FW and 0 for no
fwval <- grepl("FW", d$Pos, fixed = TRUE)
d <- d |> mutate(Pos, fw = if_else(fwval == TRUE, 1, 0))

# makes log of salary for model
d$logs <- log(d$base_salary)
```

Comparing Predictive Performance

Q2

```
set.seed(16)
# splits data into training and val
dim(d)
```

```
[1] 727  34
```

```
mls_cv <- data.frame(sample(727, (727*.7))) # creates cv set with 70 of data
mls_test <- data.frame(sample(727, (727*.3))) # creates test set with other 30
```

Q3

```
predict.regsubsets <- function(object, newdata, id, ...) {
if(is.symbol(object$call[[2]])){
i <- 2
evals_form <- function(x){
  # sets up predict method
```

```
!rlang::is_formula(eval(x), scoped = TRUE)
}
pos_evals_form <- possibly(evals_form, otherwise = FALSE)
while(pos_evals_form(object$call[[i]])){
i <- i + 1
}
tt <- eval(object$call[[i]])
} else {
tt <- as.formula(object$call[[2]])
}
mat <- model.matrix(tt, newdata)
coefj <- coef(object, id = id)
xvars <- names(coefj)
mat[, xvars] %*% coefj
}
```

```
# sets up model
set.seed(16)
train <- sample(c(TRUE, FALSE), nrow(d),
     replace = TRUE, prob=c(.7,.3))
test <- (!train)
val <- test
model <- lm(logs ~ usa + gk + df+ mf+fw+Age+MP+Starts+Min+Gls+
              Ast+PK+PKatt+CrdY+CrdR+xG+npxG+xAG+PrgC+PrgP+PrgR,
            data = d[train, ])

# gets prediction
regfit_best <- regsubsets(logs ~ usa + gk + df+ mf+fw+Age+MP+Starts+Min+Gls+
                            Ast+PK+PKatt+CrdY+CrdR+xG+npxG+xAG+PrgC+PrgP+PrgR,
                          d[train,], nvmax = 21)
reg_summary <- summary(regfit_best)
coef(regfit_best,9)
```

```
 (Intercept)          usa           gk           df          Age           MP
-1.064236816 -0.462333032 -0.601213201 -0.399351589  0.089420005 -0.048389533
      Starts          Min           xG          xAG
-0.066850289  0.001528255  0.060377046  0.070707811
```

```
#set.seed(17)
mls_cv <- data.frame(sample(727, (727*.7))) # creates cv set with 70 of data
```

```r
mls_test <- data.frame(sample(727, (727*.3))) # creates test set with other 30
#model <- lm(log(base_salary) ~ usa + gk + df+ mf+fw+Age+MP+Starts+Min+Gls+
#Ast+PK+PKatt+CrdY+CrdR+xG+npxG+xAG+PrgC+PrgP+PrgR,
#data = mls_cv)
#regfit_best <- predict.regsubsets(model, mls_cv, nvmax = 21)

#reg_summary <- summary(regfit_best)
#coef(regfit_best, 7)
```

```r
# sets up cross validation
# number of folds
k <- 10
n <- nrow(d)
# for replicability
set.seed(17)
# assign observations to folds
folds <- sample(rep(1:k, length = n))
# create container for cross-validation error
cv_errors <- matrix(NA, k, 21,
    dimnames = list(NULL, paste(1:21)))
```

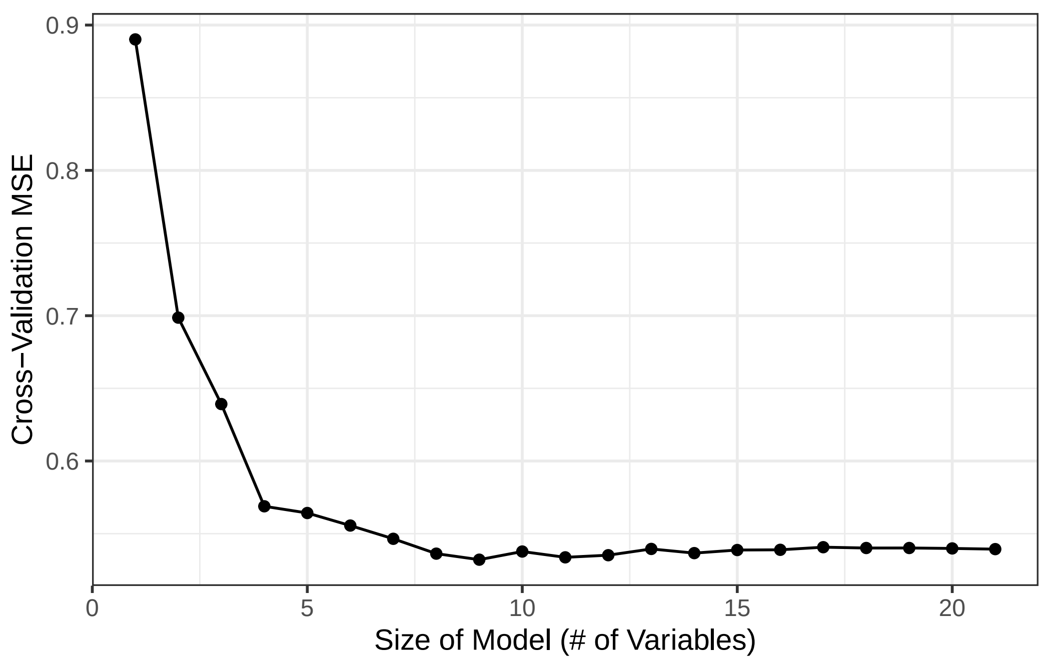
```r
# for loop for cross validation
for (j in 1:k) {
  best_fit <- regsubsets(logs ~ usa + gk + df+ mf+fw+Age+MP+Starts+Min+Gls+
                  Ast+PK+PKatt+CrdY+CrdR+xG+npxG+xAG+PrgC+PrgP+PrgR,
       data = d[folds != j, ],
       nvmax = 21)
  for (i in 1:21) {
    pred <- predict(best_fit, d[folds == j, ], id = i)
    cv_errors[j, i] <-
         mean((d$logs[folds == j] - pred)^2)
   }
 }
```

```r
# plots the cv mse
mean_cv_errors <- apply(cv_errors, 2, mean)
mean_cv_errors
```

```
        1         2         3         4         5         6         7         8
0.8901366 0.6986529 0.6391922 0.5688359 0.5642155 0.5555793 0.5464696 0.5362504
```

```
        9        10        11        12        13        14        15        16 
0.5320949 0.5377050 0.5337111 0.5351786 0.5394850 0.5366099 0.5387453 0.5388855 
       17        18        19        20        21 
0.5406909 0.5401248 0.5401391 0.5398296 0.5393567 
```

```
# plot translated to ggplot
data.frame(vars = 1:length(mean_cv_errors),
           mean_cv_errors = mean_cv_errors) %>%
  ggplot(aes(x = vars, y = mean_cv_errors)) +
  geom_point() +
  geom_line() +
  labs(x = "Size of Model (# of Variables)",
       y = "Cross-Validation MSE") +
  theme_bw()
```




```
# sets up best subsets on all
reg_best <- regsubsets(logs ~ usa + gk + df+ mf+fw+Age+MP+Starts+Min+Gls+
                Ast+PK+PKatt+CrdY+CrdR+xG+npxG+xAG+PrgC+PrgP+PrgR, data = d,
    nvmax = 21)
# prints the best, num determined by graph above
coef(reg_best, 9)
```

```
 (Intercept)          usa           gk           df          Age           MP 
-1.046936407 -0.489476449 -0.696560659 -0.357785130  0.089904587 -0.049572608 
      Starts          Min           xG          xAG 
-0.058431209  0.001413548  0.064132466  0.075624332 
```

```
# I was confused by the slack question/answer chain, so I'm
# also including only the best predictors as well just in case
# I misinterpreted the question
reg_best_only <- regsubsets(logs ~ usa + gk + df+ Age+MP+Starts+Min+xG+xAG,
                            data = d[val,],
    nvmax = 9)
coef(reg_best_only, 9)
```

```
 (Intercept)          usa           gk           df          Age           MP 
-1.027767019 -0.545131868 -0.839442309 -0.255948792  0.090867931 -0.048054298 
      Starts          Min           xG          xAG 
-0.030600963  0.001002134  0.072466022  0.090336974 
```

Based on this plot, the optimum number of variables seems to be 9 because this is where the mse is the lowest and seems to go up then taper off from there. These are being from the us, being defender, age, expected goals, being goalkeeper, games played, starts, xAG, and minutes played.

```
# calculates mse for q3
val_mat <- model.matrix(model, data = d[val, ])

val_errors <- rep(NA, 21)
for (i in 1:21) {
 coefi <- coef(regfit_best, id = i)
 pred <- val_mat[, names(coefi)] %*% coefi
 val_errors[i] <- mean((d$logs[val] - pred)^2)
}

mean(val_errors)
```

```
[1] 0.5287422
```

```
which.min(val_errors)
```

```
[1] 14
```

```r
coef(regfit_best, 14)
```

```
 (Intercept)          usa           gk           df          Age           MP 
-1.017479503 -0.471498799 -0.574458305 -0.378418936  0.087299421 -0.046746968 
      Starts          Min          Ast           PK         CrdY           xG 
-0.066320249  0.001515785 -0.036784071 -0.177272258 -0.034982947  0.252864237 
        npxG          xAG         PrgP 
-0.182133921  0.082652627  0.001722177 
```

```r
# gets mse using function
get_mse <- function(i){
 coefi <- coef(regfit_best, id = i)
 pred <- val_mat[, names(coefi)] %*% coefi
 val_errors[i] <- mean((d$logs[val] - pred)^2)
}

val_errors_purrr <- map_dbl(1:21,
                            get_mse)

#cbind(val_errors_purrr, val_errors)
which.min(val_errors_purrr)
```

```
[1] 14
```

```r
coef(regfit_best, 14) # here it says the best is 14, which is
```

```
 (Intercept)          usa           gk           df          Age           MP 
-1.017479503 -0.471498799 -0.574458305 -0.378418936  0.087299421 -0.046746968 
      Starts          Min          Ast           PK         CrdY           xG 
-0.066320249  0.001515785 -0.036784071 -0.177272258 -0.034982947  0.252864237 
        npxG          xAG         PrgP 
-0.182133921  0.082652627  0.001722177 
```

```
# about the same visually as 9 so that makes sense

# prints mse
mean(val_errors_purrr)
```

```
[1] 0.5287422
```

From the test set, the MSE is .5287.

Q4

```
set.seed(18)

# sets up matrix
x <- model.matrix(logs ~ usa + gk + df+ mf+fw+Age+MP+Starts+Min+Gls+
                    Ast+PK+PKatt+CrdY+CrdR+xG+npxG+xAG+PrgC+PrgP+PrgR,
                  d)[, -1]
y <- d$logs

# fits ridge
grid <- 10^seq(10, -2, length = 100)
ridge_mod <- cv.glmnet(x, y, alpha = 0, lambda = grid)

train <- sample(1:nrow(x), nrow(x) / 2)
val <- (-train)
y_val <- y[val]

# fit ridge on train
ridge_mod2 <- glmnet(x[train, ], y[train], alpha = 0,
    lambda = grid, thresh = 1e-12)
ridge_pred <- predict(ridge_mod2, s = 4, newx = x[val, ])

# get best l
cv_out <- cv.glmnet(x[train, ], y[train], alpha = 0)

bestlam <- cv_out$lambda.min
bestlam
```

```
[1] 0.05331591
```

```
# get mse
ridge_pred <- predict(ridge_mod, s = bestlam,
    newx = x[val, ])
mean((ridge_pred - y_val)^2)
```

```
[1] 0.5834029
```

The best lam is .0533. The test set MSE is .5834.

Q5

```
set.seed(18)
lasso_mod <- glmnet(x[train, ], y[train], alpha = 1,
    lambda = grid)

# get l
cv_out_lasso <- cv.glmnet(x[train, ], y[train], alpha = 1)
bestlaml <- cv_out_lasso$lambda.min
bestlaml
```

```
[1] 0.001046613
```

```
# gets MSe
lasso_pred <- predict(lasso_mod, s = bestlaml,
    newx = x[val, ])
mean((lasso_pred - y_val)^2)
```

```
[1] 0.6344207
```

The optimal l is .0010. The test set MSE is .6344. The approach with the lowest test MSE is from best subset

Comparing Variable Selection Approaches

Q6

```r
# refits to full data, uses best l to get coefs
out <- glmnet(x, y, alpha = 1, lambda = grid)
lasso_coef <- predict(out, type = 'coefficient', s = bestlaml)
lasso_coef
```

```
22 x 1 sparse Matrix of class "dgCMatrix"
                       s1
(Intercept) -1.1267563453
usa         -0.4762881026
gk          -0.3889820512
df          -0.2302142725
mf           .
fw           0.0466357573
Age          0.0858353320
MP          -0.0262327368
Starts       .
Min          0.0004454304
Gls          .
Ast          .
PK           .
PKatt        0.0013521990
CrdY        -0.0152958618
CrdR         0.0086974456
xG           0.0690201896
npxG         .
xAG          0.0441577259
PrgC         .
PrgP         0.0021735394
PrgR         .
```

The characteristics that seem to be important with the lasso model are being from the us, being a goalkeeper, defender, or forward, as well as age, num games played, minutes played (but barely), penalty kicks attempted (barely), yellow cards, red cards, expected goals, nonpenalty expected goals, and progressive passes by the player.

Q7

```r
# gets best subsets for 9, which was what the determined best number was
reg_best_full <- regsubsets(logs ~ usa + gk + df+ mf+fw+Age+MP+Starts+Min+Gls+
                  Ast+PK+PKatt+CrdY+CrdR+xG+npxG+xAG+PrgC+PrgP+PrgR, data = d,
    nvmax = 9)

# gets reduced best subsets using only the suggested number
reg_best_full1 <- regsubsets(logs ~ usa + gk + df+ Age+MP+Starts+Min+xG+xAG, data = d,
    nvmax = 9)

# checks that they are same and displays important characteristics
cbind(coef(reg_best_full1, 9), coef(reg_best_full, 9))
```

```
                   [,1]         [,2]
(Intercept) -1.046936407 -1.046936407
usa         -0.489476449 -0.489476449
gk          -0.696560659 -0.696560659
df          -0.357785130 -0.357785130
Age          0.089904587  0.089904587
MP          -0.049572608 -0.049572608
Starts      -0.058431209 -0.058431209
Min          0.001413548  0.001413548
xG           0.064132466  0.064132466
xAG          0.075624332  0.075624332
```

The characteristics that seem most important are being from the us, being goalkeeper, being defender, age, games played, minutes played, starts, expected non-penalty and overall goals.

Q8

They are mostly the same, with lasso having a few more important characteristics. I think the best subsets was mostly what I would think of in general with my limited soccer knowledge, like age and expected goals, while the lasso got into more details that might be something that someone who knows more about soccer may expect to influence pay, such as yellow cards and red cards. Overall, what I expected to be included was mostly included and the two models have overlap in what is selected.