Hu_Anqi_HW3

Anqi Hu

2/9/2020

Conceptual Exercises

1

```
set.seed(1234)

p = 20
n = 1000

x = matrix(rnorm(n*p, 0, 2),n,p)

b <- rnorm(p)
e <- rnorm(p)

b[2] <- 0
b[13] <- 0
b[16] <- 0
b[20] <- 0

y <- x %*% b + e
```

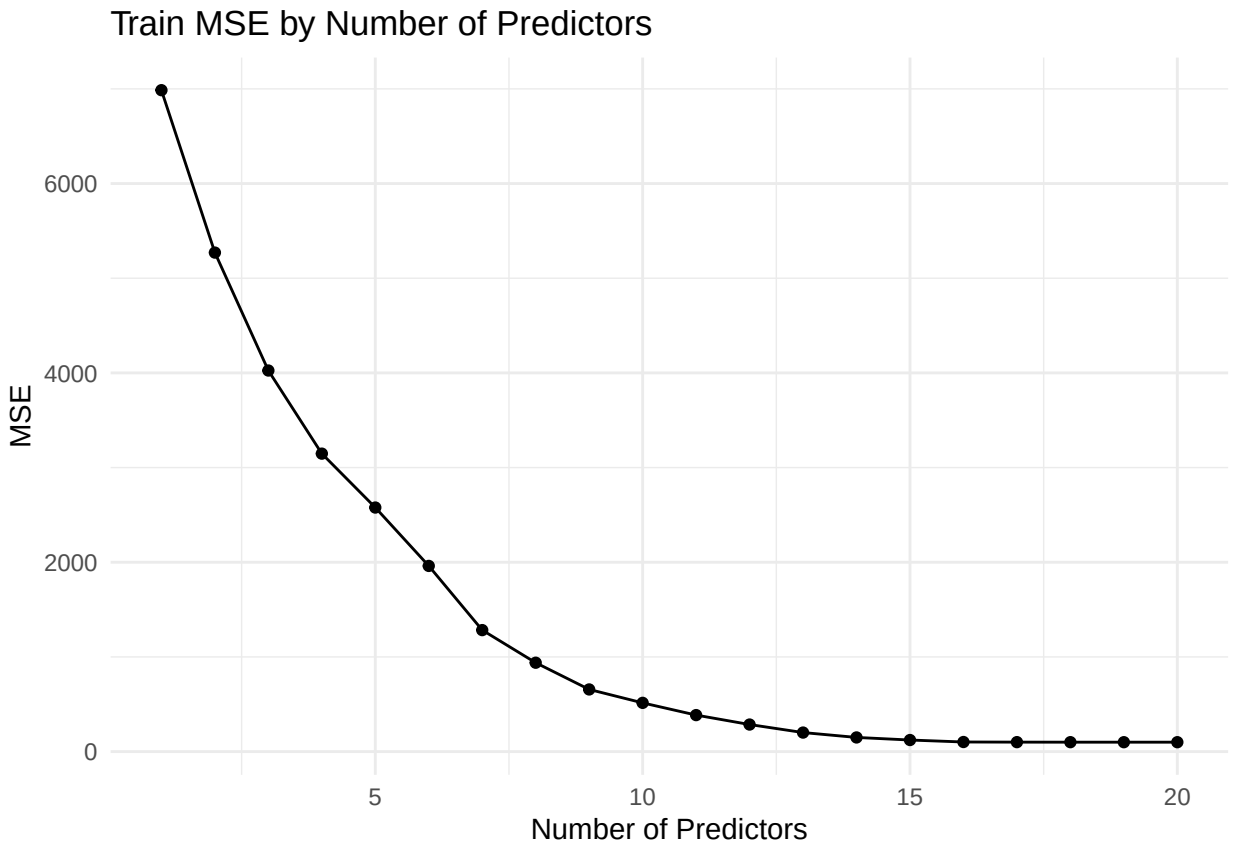
2

```
df <- data.frame(y, x)
split <- initial_split(df, prop = 0.1)
train <- training(split)
test <- testing(split)
```

3

```
best_subset <- regsubsets(y ~ ., data = train, nvmax = 20)

results <- summary(best_subset)
mse_list = data.frame(matrix(ncol = 2, nrow = 20))

for (i in 1:20) {
  best_coef = coef(best_subset, id = i)
  pred = as.matrix(train[, colnames(train) %in% names(best_coef)]) %*% best_coef[names(best_coef) %in%
```

```
  mse_list[i,] = c(i, mse(preds = pred[,1], actuals = train["y"]))
}

colnames(mse_list) <- c("size", "MSE")

ggplot(mse_list) +
  geom_line(aes(x = size, y = MSE)) +
  geom_point(aes(x = size, y = MSE)) +
  labs(title = "Train MSE by Number of Predictors",
       x = "Number of Predictors")
```

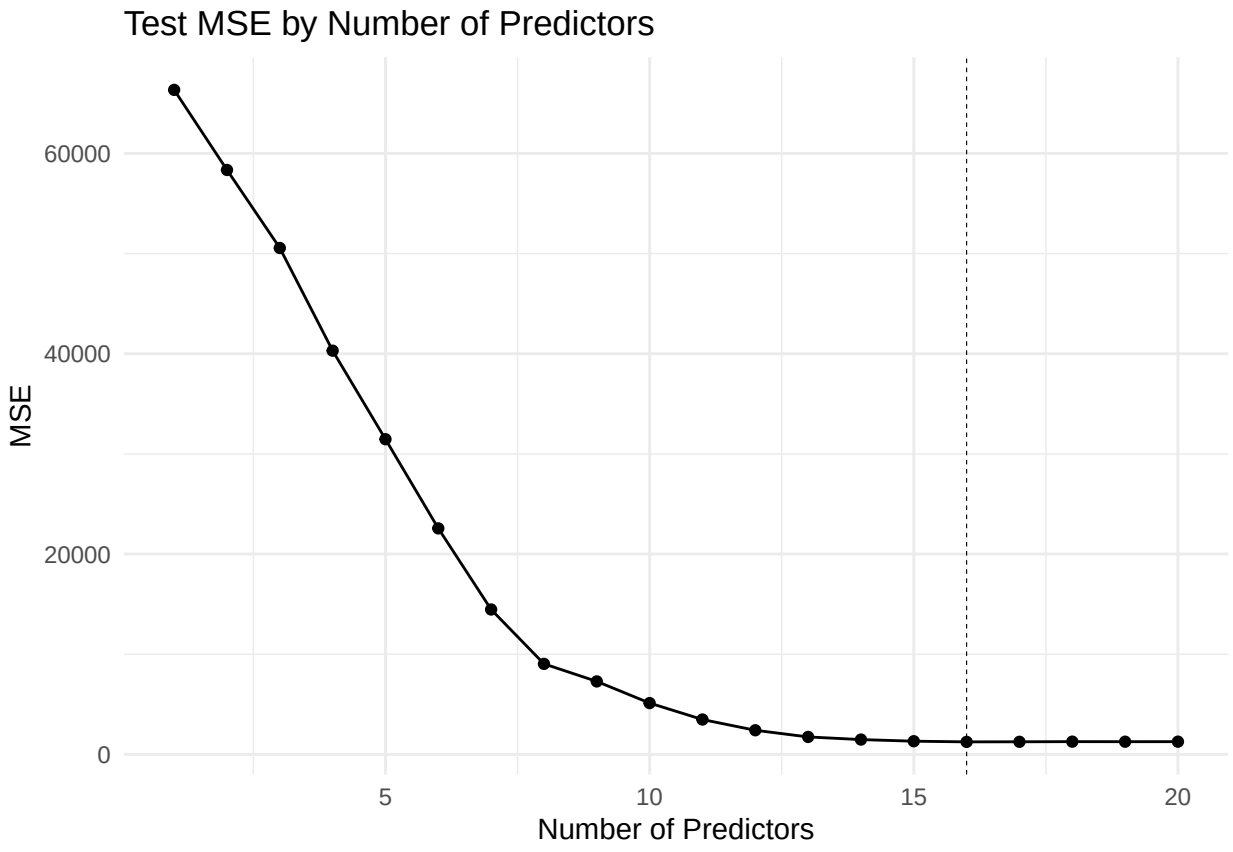


For the training set, the model that includes all 20 features has the smallest MSE.

4

```
mse_test = data.frame(matrix(ncol = 2, nrow = 20))

for (i in 1:20) {
  best_coef = coef(best_subset, id = i)
  pred = as.matrix(test[, colnames(test) %in% names(best_coef)]) %*% best_coef[names(best_coef) %in% co

  mse_test[i,2] = mse(preds = pred[,1], actuals = test["y"])
  mse_test[i,1] = i
}
```

```r
colnames(mse_test) <- c("size", "MSE")

ggplot(mse_test) +
  geom_line(aes(x = size, y = MSE)) +
  geom_point(aes(x = size, y = MSE)) +
  geom_vline(aes(xintercept = which.min(mse_test[,2])),
             color= "black", linetype="dashed", size=0.2) +
  labs(title = "Test MSE by Number of Predictors",
       x = "Number of Predictors")
```



5

For the model size of 16, the test set MSE reaches the minimum value.

```r
b
```

```
##  [1] -1.687863  0.000000 -0.648057  0.261034 -1.219694 -1.550189  0.775057
##  [8]  1.758114  1.417998 -1.269144 -1.267759  0.773924  0.000000  0.221154
## [15]  0.213797  0.000000 -0.444971  2.123822 -0.503633  0.000000
```

6

```r
coef(best_subset, id = 16)
```

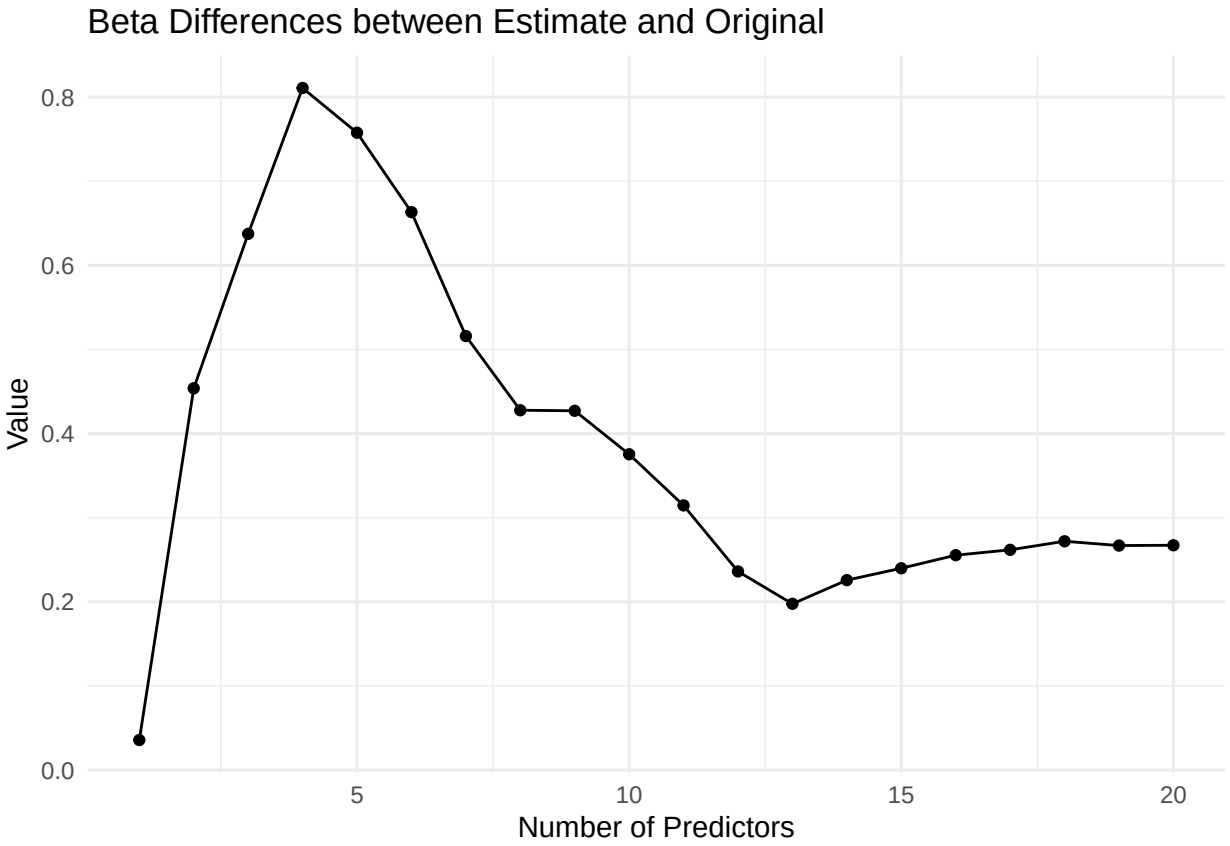
```
## (Intercept)          X1          X3          X4          X5          X6
##  -0.0419104  -1.7297910  -0.6513579   0.3242207  -1.1719746  -1.5503841
```

```
##          X7          X8          X9         X10         X11         X12
##   0.6454937   1.6871731   1.3766028  -1.2124854  -1.2880311   0.8467474
##         X14         X15         X17         X18         X19
##   0.2882920   0.2553055  -0.4802096   2.0159223  -0.5786489
```

Compared to the true model that was used to generate the dataset, the best model in the test set has a higher coefficient for 9 of the 20 predictors (x3, x5, x6, x9, x10, x11, x12, x14, x17), lower coefficient for the other ones.

7

```
mse_compare <- data.frame(matrix(ncol = 2, nrow = 20))

all_b <- data.frame(matrix(ncol = 20, nrow = 1))

all_b[1, ] <- b
colnames(all_b) <- c(paste("X", c(1:20), sep = ""))

for (i in (1:20)) {
  best_coef = coef(best_subset, id = i)

  sum_diff = 0
  coef = names(best_coef)[-1]

  for (c in coef) {
    diff = all_b[c] - best_coef[names(best_coef)][c]
    diff_sq = diff ^2
    sum_diff = sum_diff + diff_sq
  }

  val = sqrt(sum_diff)
  mse_compare[i, ] = c(i, val)
}

colnames(mse_compare) <- c("size", "value")

ggplot(mse_compare) +
  geom_line(aes(x = size, y = value)) +
  geom_point(aes(x = size, y = value)) +
  labs(title = "Beta Differences between Estimate and Original",
       x = "Number of Predictors",
       y = "Value")
```



Overall, value is declining as the number of predictors increases. This pattern differs from the one of the test MSE, yet the general trends are similar. In the test MSE plot, the peak is when the number of predictors is 1, whereas in the beta differences, the peak has shifted to when there are 4 predictors. In addition, this trend takes a dip when there are 14 predictors, whereas the test MSE has its lowest point at 16. These are similar in that regard.

Application Exercises

1

```
gss_train <- read.csv(file = "data/gss_train.csv")
gss_test <- read.csv(file = "data/gss_test.csv")

train_x <- model.matrix(egalit_scale ~ ., gss_train) [, -1]
train_y <- gss_train$egalit_scale

test_x <- model.matrix(egalit_scale ~ ., gss_test) [, -1]
test_y <- gss_test$egalit_scale
```

```
linear <- lm(egalit_scale ~ ., gss_train)

lin_x <- as.data.frame(test_x)

lin_pred <- predict(linear, lin_x)
```

```r
lin_mse = mse(preds = lin_pred, actuals = test_y)
lin_mse
```

```
## [1] 63.2136
```

The test MSE of linear regression is 63.2136.

2

```r
# 10-fold CV
ridge <- cv.glmnet(
  x = train_x,
  y = train_y,
  alpha = 0.0,
  nfolds = 10
)

# predict test set
lam = ridge$lambda.min
ridge_pred <- predict(ridge, newx = test_x, s = lam)
ridge_mse = mse(preds = ridge_pred, actuals = test_y)
ridge_mse
```

```
## [1] 61.0378
```

The test MSE of ridge regression is 61.0378.

3

```r
# 10-fold CV
lasso <- cv.glmnet(
  x = train_x,
  y = train_y,
  alpha = 1.0,
  nfolds = 10
)

# predict test set
lam = lasso$lambda.min
lasso_pred = predict(lasso, newx = test_x, s = lam)
lasso_mse = mse(preds = lasso_pred, actuals = test_y)
lasso_non_zero <- length(coef(lasso)[coef(lasso) != 0])

c(lasso_mse, lasso_non_zero)
```

```
## [1] 61.2234 15.0000
```

The test MSE of the Lasso regression is 61.2234 and it includes 15 non-zero coefficients.

4

```
a = seq(0, 1, by = 0.1)

gather <- data.frame(matrix(ncol = 3, nrow = 11))

en <- cv.glmnet(
    x = train_x,
    y = train_y,
    nfolds = 10)

lam = en$lambda.min

for (i in seq_along(a)) {
  en <- cv.glmnet(
    x = train_x,
    y = train_y,
    alpha = a[i])

  en_pred = predict(en, newx = test_x, s = lam)

  en_mse = mse(preds = en_pred, actuals = test_y)
  non_zero = length(coef(en)[coef(en) != 0])
  gather[i, ] = c(a[i], en_mse, non_zero)
}

colnames(gather) <- c("Alpha", "MSE", "Non-zero Coefficients")

gather[which.min(gather$MSE),]
```

```
##   Alpha     MSE Non-zero Coefficients
## 8   0.7 61.1596                    15
```

The test MSE of the Elastic Net model is 61.1596 and it includes 15 non-zero coefficients.

5

```
summary <- data.frame(matrix(ncol = 3, nrow = 4))

summary[1, ] = c("Linear Model" , lin_mse, R2(lin_pred, test_y))
summary[2, ] = c("Ridge Reg." , ridge_mse, R2(ridge_pred, test_y))
summary[3, ] = c("Lasso Reg." , lasso_mse, R2(lasso_pred, test_y))
summary[4, ] = c("Elastic Net" , gather[which.min(gather$MSE),]["MSE"], R2(en_pred, test_y))

colnames(summary) <- c("Model", "MSE", "R Squared")

summary
```

```
##          Model               MSE         R Squared
## 1 Linear Model 63.2136296230151 0.310401545747136
## 2   Ridge Reg. 61.0378065828335 0.327761282589625
## 3   Lasso Reg. 61.2234149089147 0.324700273119064
## 4  Elastic Net  61.159557125261 0.324091105876421
```

Comparing the four models, their MSEs do not seem to vary by much from one another. The Ridge Regression is the best-fitting so far with respect to MSE. The difference in value of the two most extreme MSEs is

less than 3. This would also mean that the change of different approaches did not help with improving the performance of prediction. In terms of R^2, the four values do not cover a wide range, either. None of the four seems to be performing extremely well.