An introduction to statistical learning - Ch6 - Ex9

Thalles Quinaglia Liduares

24/03/2023

```
library(ISLR)
library(caret)
```

```
## Carregando pacotes exigidos: ggplot2
```

```
## Carregando pacotes exigidos: lattice
```

Upload data

```
data<-ISLR::College
```

Split data

```
set.seed(1234)
trainIndex <- createDataPartition(College$Apps, p = 0.7, list = FALSE)
trainData <- College[trainIndex, ]
testData <- College[-trainIndex, ]
```

b. Fit a linear model using least squares on the training set, and report the test error obtained.

```
# Fit linear model on training set
lm_fit <- lm(Apps ~ ., data = trainData)


# Predict on test set
lm_pred <- predict(lm_fit, newdata = testData)

# Calculate RMSE
lm_rmse <- sqrt(mean((lm_pred - testData$Apps)^2))

# Print test error
cat("Test RMSE for linear model:", lm_rmse, "\n")
```

```
## Test RMSE for linear model: 985.3601
```

c. Fit a ridge regression model on the training set, with λ chosen by cross-validation. Report the test error obtained.

```
library(glmnet)
```

```
## Carregando pacotes exigidos: Matrix
```

```
## Loaded glmnet 4.1-6
```

Convert data to matrix format

```
x_train <- model.matrix(Apps ~ ., data = trainData)[,-1]
y_train <- trainData$Apps
```

Fit ridge regression model with cross-validation

```
set.seed(123)

cv_fit <- cv.glmnet(x_train, y_train, alpha = 0, nfolds = 5)
```

Choose best lambda

```
best_lambda <- cv_fit$lambda.min
```

Fit ridge regression model with chosen lambda

```
ridge_fit <- glmnet(x_train, y_train, alpha = 0, lambda = best_lambda)
```

Predict on test set

```
x_test <- model.matrix(Apps ~ ., data = testData)[,-1]
ridge_pred <- predict(ridge_fit, newx = x_test)
```

Calculate RMSE

```
ridge_rmse <- sqrt(mean((ridge_pred - testData$Apps)^2))
```

Print test error

```
cat("Test RMSE for ridge regression model:", ridge_rmse, "\n")
```

```
## Test RMSE for ridge regression model: 900.8327
```

d. Fit a lasso model on the training set, with λ chosen by crossvalidation. Report the test error obtained, along with the number of non-zero coefficient estimate.

Fit lasso model with cross-validation

```
set.seed(1234)
cv_fit <- cv.glmnet(x_train, y_train, alpha = 1, nfolds = 5)
```

Choose best lambda

```
best_lambda <- cv_fit$lambda.min
```

Fit lasso model with chosen lambda

```
lasso_fit <- glmnet(x_train, y_train, alpha = 1, lambda = best_lambda)
```

Predict on test set

```
x_test <- model.matrix(Apps ~ ., data = testData)[,-1]
lasso_pred <- predict(lasso_fit, newx = x_test)
```

Calculate RMSE

```
lasso_rmse <- sqrt(mean((lasso_pred - testData$Apps)^2))
```

Print test error

```
cat("Test RMSE for lasso model:", lasso_rmse, "\n")
```

```
## Test RMSE for lasso model: 949.881
```

Print number of non-zero coefficient estimates

```
cat("Number of non-zero coefficient estimates:", sum(coef(lasso_fit) != 0), "\n")
```

```
## Number of non-zero coefficient estimates: 16
```