Assignment22

John Bute

2024-11-08

We will now try to predict per capita crime rate in the Boston data set. 1. Try out some of the regression methods explored in this chapter, such as best subset selection, the LASSO, ridge regression, and PCR. Present and discuss results for the approaches that you consider.

Set up:

```
library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 4.1-8

library(leaps)
library(pls)

##
## Attaching package: 'pls'

## The following object is masked from 'package:stats':
##
##     loadings

set.seed(123)
Boston <- read.csv("C:/Users/johnb/Desktop/Machine Learning/data/Boston.csv")
y <- Boston$crim
x <- data.matrix(Boston[, c('zn', 'indus', 'chas', 'nox', 'rm', 'age', 'dis',
'rad', 'tax', 'ptratio', 'black', 'lstat', 'medv')])
train_indices <- sample(1:nrow(Boston), nrow(Boston) * 0.7)

x_train <- x[train_indices, ]
y_train <- y[train_indices]
x_test <- x[-train_indices, ]
y_test <- y[-train_indices]


train_data <- Boston[train_indices, ]
test_data <- Boston[-train_indices, ]
```

Best subset:

```
best_subset <- regsubsets(crim ~ ., data = train_data, nvmax = 13)
best_subset_summary <- summary(best_subset)
```

```r
best_model_size <- which.min(best_subset_summary$bic)
cat("Best Subset Model Size:", best_model_size, "\n")

## Best Subset Model Size: 2

best_model_coef <- coef(best_subset, best_model_size)
print(best_model_coef)

## (Intercept)         rad       lstat
##  -4.7907724   0.5677651   0.2570736

x_test_subset <- as.data.frame(x_test)
y_best_subset <- as.matrix(x_test_subset[, names(best_model_coef)[-1]]) %*%
best_model_coef[-1] + best_model_coef[1]
mse_best_subset <- mean((y_test - y_best_subset)^2)

print(mse_best_subset)

## [1] 19.31744
```

Lasso regression:

```r
x_train_matrix <- model.matrix(crim ~ ., train_data)[, -1]
x_test_matrix <- model.matrix(crim ~ ., test_data)[, -1]

lasso_model <- cv.glmnet(x_train_matrix, y_train, alpha = 1)

best_lambda_lasso <- lasso_model$lambda.min
cat("Best Lambda for LASSO:", best_lambda_lasso, "\n")

## Best Lambda for LASSO: 0.05596583

lasso_pred <- predict(lasso_model, s = best_lambda_lasso, newx =
x_test_matrix)
lasso_mse <- mean((y_test - lasso_pred)^2)
cat("LASSO MSE:", lasso_mse, "\n")

## LASSO MSE: 18.10354
```

Ridge regression:

```r
ridge_model <- cv.glmnet(x_train_matrix, y_train, alpha = 0)
best_lambda_ridge <- ridge_model$lambda.min
cat("Best Lambda for Ridge:", best_lambda_ridge, "\n")

## Best Lambda for Ridge: 0.5863068

ridge_pred <- predict(ridge_model, s = best_lambda_ridge, newx =
x_test_matrix)
ridge_mse <- mean((y_test - ridge_pred)^2)
cat("Ridge MSE:", ridge_mse, "\n")

## Ridge MSE: 17.57819
```

Ridge regression:

```
pcr_model <- pcr(crim ~ ., data = train_data, validation = "CV")
summary(pcr_model)
```

```
## Data:    X dimension: 354 14
##  Y dimension: 354 1
## Fit method: svdpc
## Number of components considered: 14
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           9.525    7.919    7.747    7.762    7.757    7.751    7.692
## adjCV        9.525    7.915    7.742    7.756    7.751    7.745    7.685
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       7.532    7.492    7.378     7.353     7.318     7.349     7.351
## adjCV    7.522    7.483    7.369     7.343     7.308     7.336     7.339
##        14 comps
## CV        7.355
## adjCV     7.341
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8
comps
## X       73.21    88.50    98.02    99.36    99.80    99.93    99.95
99.98
## crim    31.76    34.89    35.12    35.22    35.37    36.80    39.55
40.57
##       9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
## X       99.99    100.00    100.00    100.00    100.00    100.00
## crim    42.57     43.03     43.87     43.96     43.96     44.29
```

```
optimal_components <- which.min(pcr_model$validation$PRESS)
cat("Optimal number of components for PCR:", optimal_components, "\n")
```

```
## Optimal number of components for PCR: 11
```

```
pcr_pred <- predict(pcr_model, test_data, ncomp = optimal_components)
pcr_mse <- mean((y_test - pcr_pred)^2)
cat("PCR MSE:", pcr_mse, "\n")
```

```
## PCR MSE: 18.91475
```

```
model_performance <- data.frame(
  Model = c("Best Subset", "LASSO", "Ridge", "PCR"),
  MSE = c(mse_best_subset, lasso_mse, ridge_mse, pcr_mse)
)

print(model_performance)
```

```
##         Model      MSE
## 1 Best Subset 19.31744
## 2       LASSO 18.10354
## 3       Ridge 17.57819
## 4         PCR 18.91475
```

2. Propose a model (or set of models) that seem to perform well on this data set, and justify your answer. Make sure that you are evaluating model performance using validation set error, cross- validation, or some other reasonable alternative, as opposed to using training error.

We are using ridge regression, as it has the lowest MSE among all the models. However, if we want to not use all the predictors, then LASSO might be useful, as it selects only the most important features, and thus is more interpretable.

3. Does your chosen model involve all of the features in the data set? Why or why not?

```
coef(ridge_model)

## 15 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept)  1.595467704
## X            0.001586564
## zn          -0.003129776
## indus        0.029209222
## chas        -0.180245853
## nox          1.830983668
## rm          -0.147872577
## age          0.006070725
## dis         -0.087472414
## rad          0.042686496
## tax          0.001937224
## ptratio      0.068259453
## black       -0.002595542
## lstat        0.034533319
## medv        -0.023362747
```

As you see, it does use all the features in the dataset, as we are using ridge regression. Ridge regression shrinks coefficients towards zero (without setting them to zero) and thus we include all the features. However, some are way more impactful than others.