Analysis of Ridge and Lasso Regression on Simulated Data

Alexandre H.

Data Generation

This section is focused on generating the dataset used for the analysis. It involves loading necessary libraries, initializing parameters, and creating the training and testing datasets.

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.3.2
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```
library("tidyverse")
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::expand() masks Matrix::expand()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x tidyr::pack()   masks Matrix::pack()
## x tidyr::unpack() masks Matrix::unpack()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
set.seed(0)
p <- 50
n <- 1100

X <- matrix(rnorm(n*p, 0, 1), nrow=n, ncol=p)
epsilon <- rnorm(n, 0, 1)
beta <- c(2, 2, 2, 2, 2, rep(0, p-5))
y <- X %*% beta + epsilon

train_id <- sample(seq_len(nrow(X)), 100)
```

```r
X_train <- X[train_id,]
y_train <- y[train_id]
X_test <- X[-train_id,]
y_test <- y[-train_id]

grid <- 10^seq(10,-2,length = 100)
```

Part 1: Cross-Validation for Ridge and Lasso Regression

This part involves performing cross-validation on both ridge and lasso regression models, calculating and comparing their test mean squared errors (MSEs).

```r
cvr <- cv.glmnet(X_train,y_train,alpha=0,lambda=grid)
bestlam_r <- cvr$lambda.min
best_fit_r <- glmnet(X_train,y_train,alpha=0,lambda=bestlam_r)
test_pred_r <- predict(best_fit_r, newx=X_test, s=bestlam_r)

mean((y_test - test_pred_r)^2)
```

```
## [1] 1.87885
```

```r
cvl <- cv.glmnet(X_train,y_train,alpha=1,lambda=grid)
bestlam_l <- cvl$lambda.min
best_fit_l <- glmnet(X_train,y_train,alpha=1,lambda=bestlam_l)
test_pred_l <- predict(best_fit_l, newx=X_test, s=bestlam_l)
mean((y_test - test_pred_l)^2)
```

```
## [1] 1.150411
```

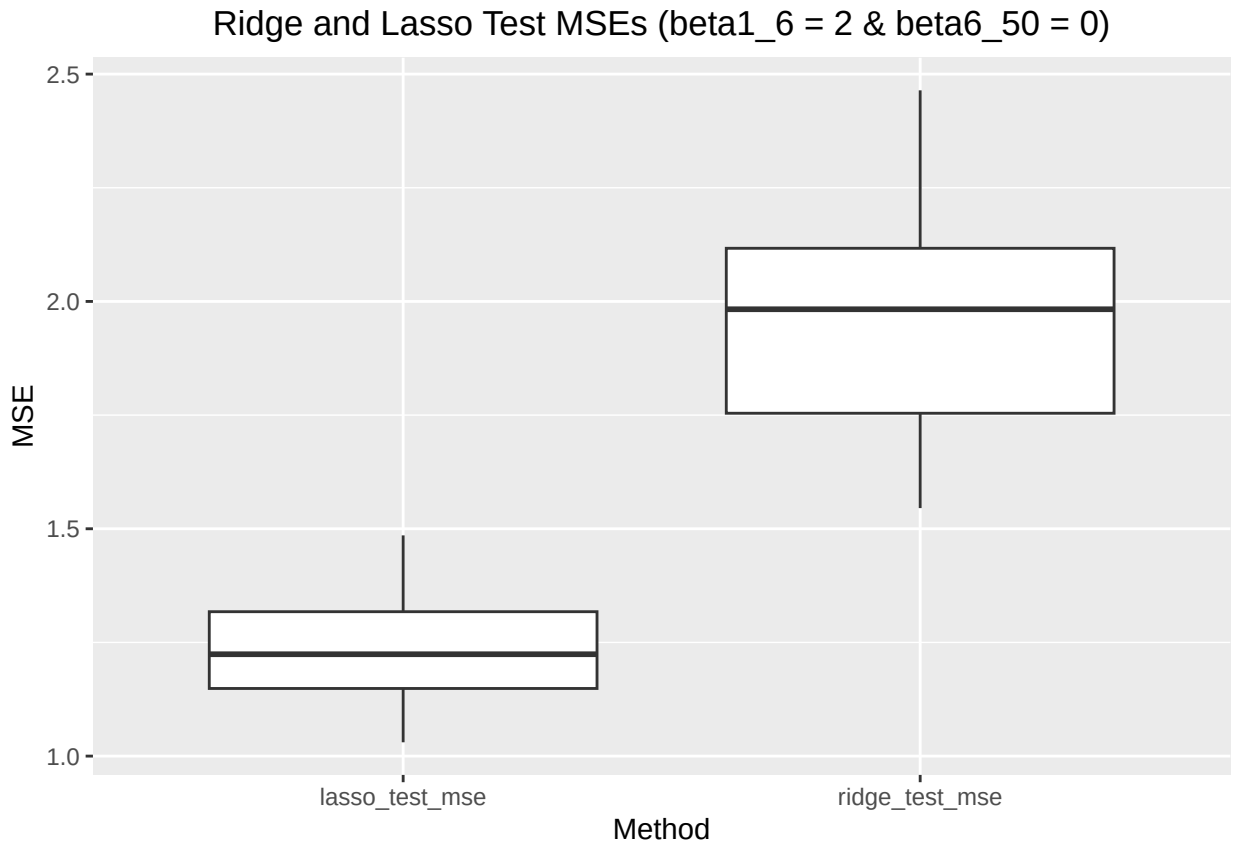
The cross validation lasso regression has a smaller test MSE than the cross validation ridge regression.

Part 2: MSE Comparison of Ridge and Lasso Across Multiple Simulations

In this section, multiple simulations are run to compare the test MSEs of ridge and lasso regression methods, followed by visual representation through a boxplot.

```r
ridge_test_mse <- rep(NA, 50)
lasso_test_mse <- rep(NA, 50)


for (i in 2:50){

  set.seed(i)
  p <- 50
  n <- 1100

  X <- matrix(rnorm(n*p, 0, 1), nrow=n, ncol=p)
```

```r
  epsilon <- rnorm(n, 0, 1)
  beta <- c(2, 2, 2, 2, 2, rep(0, p-5))
  y <- X %*% beta + epsilon

  train_id <- sample(seq_len(nrow(X)), 100)
  X_train <- X[train_id,]
  y_train <- y[train_id]
  X_test <- X[-train_id,]
  y_test <- y[-train_id]

  grid <- 10^seq(10,-2,length = 100)

  cvr <- cv.glmnet(X_train,y_train,alpha=0,lambda=grid)
  bestlam_r <- cvr$lambda.min
  best_fit_r <- glmnet(X_train,y_train,alpha=0,lambda=bestlam_r)
  test_pred_r <- predict(best_fit_r, newx=X_test, s=bestlam_r)
  ridge_test_mse[i] <- mean((y_test - test_pred_r)^2)

  cvl <- cv.glmnet(X_train,y_train,alpha=1,lambda=grid)
  bestlam_l <- cvl$lambda.min
  best_fit_l <- glmnet(X_train,y_train,alpha=1,lambda=bestlam_l)
  test_pred_l <- predict(best_fit_l, newx=X_test, s=bestlam_l)
  lasso_test_mse[i] <- mean((y_test - test_pred_l)^2)
}

ridge_test_mse <- ridge_test_mse[2:50]
lasso_test_mse <- lasso_test_mse[2:50]

df <- data.frame(ridge_test_mse, lasso_test_mse)
df <- df %>% pivot_longer(cols = c(ridge_test_mse, lasso_test_mse), names_to = "method", values_to = "t
ggplot(df, aes(x=method, y=test_mse)) +
  geom_boxplot() +
  ggtitle("Ridge and Lasso Test MSEs (beta1_6 = 2 & beta6_50 = 0)") +
  xlab("Method") +
  ylab("MSE") +
  theme(plot.title = element_text(hjust = 0.5))
```



The boxplot tells us that the lasso regression generally has a lower test MSE compared to ridge regression. This is consistent with the expectation because lasso can make exactly zero irrelevant coefficients. This provides a more parsimonious model that better captures our generated data.

Ridge regression tends to shrink all coefficients, but it does not set any to exactly zero. In situations where many predictors are irrelevant, this can result in modeling noise that takes away predictive capability. The boxplot shows that the ridge test MSE values are generally higher, which is consistent with this explanation.

The interquartile range for lasso is tighter than that for ridge. This tells us that the lasso performance is more consistent across different data realizations compared to ridge.

To conclude, given the way we have generated our data, the boxplot is in line with expectations. The lasso regression performs better in terms of test MSE, and more consistently than ridge regression in this simulation.

Part 3: Comparative Evaluation of Ridge and Lasso Regression with Uniform Beta Coefficients

Part 3.1: Ridge vs. Lasso with Uniform Beta Coefficients

This subsection revisits the cross-validation process for ridge and lasso regression with a modified data generation where all beta coefficients are uniform, assessing the performance of each model.

```
set.seed(0)
p <- 50
```

```r
n <- 1100

X <- matrix(rnorm(n*p, 0, 1), nrow=n, ncol=p)
epsilon <- rnorm(n, 0, 1)
beta <- rep(0.5, p)
y <- X %*% beta + epsilon

train_id <- sample(seq_len(nrow(X)), 100)
X_train <- X[train_id,]
y_train <- y[train_id]
X_test <- X[-train_id,]
y_test <- y[-train_id]

grid <- 10^seq(10,-2,length = 100)

cvr <- cv.glmnet(X_train,y_train,alpha=0,lambda=grid)
bestlam_r <- cvr$lambda.min
best_fit_r <- glmnet(X_train,y_train,alpha=0,lambda=bestlam_r)
test_pred_r <- predict(best_fit_r, newx=X_test, s=bestlam_r)

mean((y_test - test_pred_r)^2)
```

```
## [1] 1.640216
```

```r
cvl <- cv.glmnet(X_train,y_train,alpha=1,lambda=grid)
bestlam_l <- cvl$lambda.min
best_fit_l <- glmnet(X_train,y_train,alpha=1,lambda=bestlam_l)
test_pred_l <- predict(best_fit_l, newx=X_test, s=bestlam_l)
mean((y_test - test_pred_l)^2)
```

```
## [1] 1.834476
```

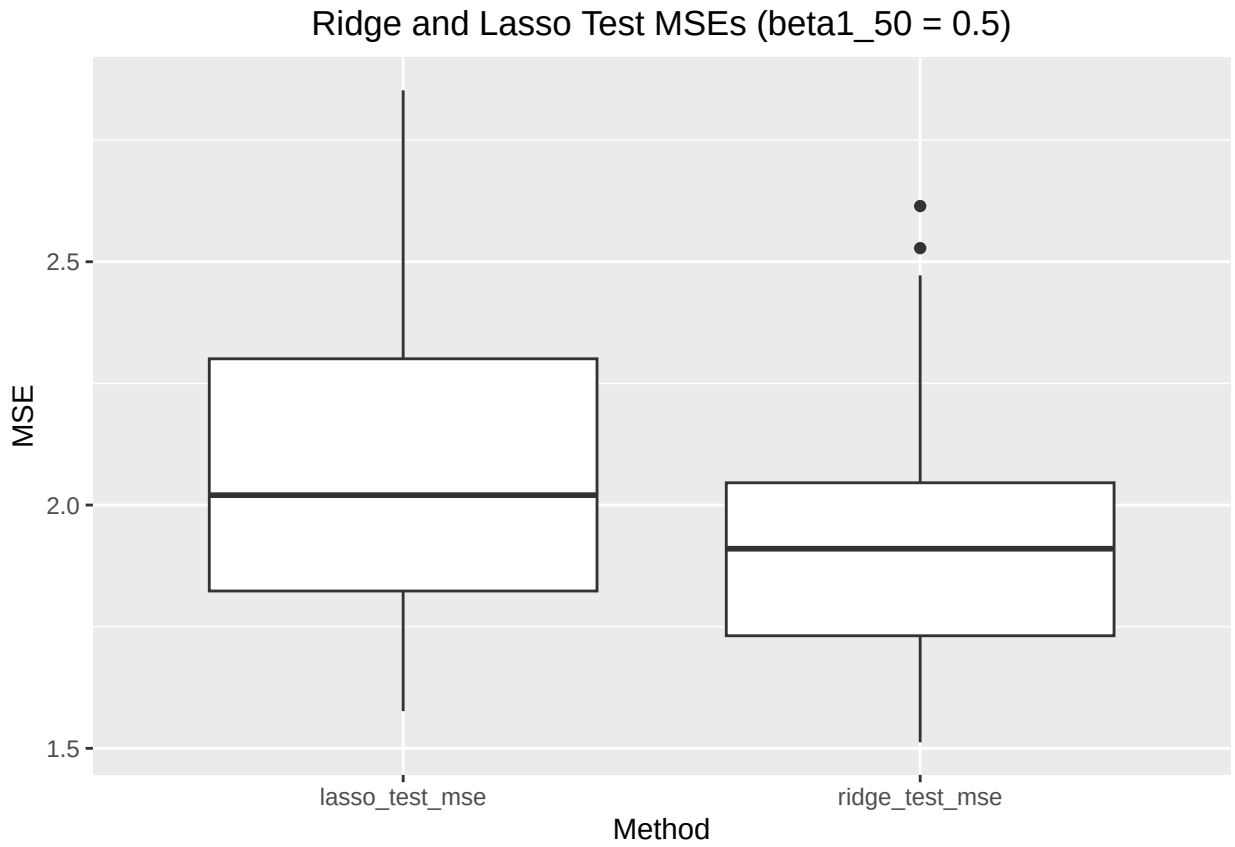
The cross validation ridge regression has a smaller test MSE than cross validation lasso regression.

Part 3.2: Comparative Analysis of Test MSEs for Ridge and Lasso with Uniform Betas

The final part conducts a similar analysis to Part 3.1 but over multiple simulations, providing a detailed comparison of the test MSEs for ridge and lasso regression in the context of uniformly distributed beta coefficients.

```r
ridge_test_mse <- rep(NA, 50)
lasso_test_mse <- rep(NA, 50)


for (i in 2:50){

  set.seed(i)
  p <- 50
  n <- 1100
```

```
  X <- matrix(rnorm(n*p, 0, 1), nrow=n, ncol=p)
  epsilon <- rnorm(n, 0, 1)
  beta <- rep(0.5, p)
  y <- X %*% beta + epsilon

  train_id <- sample(seq_len(nrow(X)), 100)
  X_train <- X[train_id,]
  y_train <- y[train_id]
  X_test <- X[-train_id,]
  y_test <- y[-train_id]

  grid <- 10^seq(10,-2,length = 100)

  cvr <- cv.glmnet(X_train,y_train,alpha=0,lambda=grid)
  bestlam_r <- cvr$lambda.min
  best_fit_r <- glmnet(X_train,y_train,alpha=0,lambda=bestlam_r)
  test_pred_r <- predict(best_fit_r, newx=X_test, s=bestlam_r)
  ridge_test_mse[i] <- mean((y_test - test_pred_r)^2)

  cvl <- cv.glmnet(X_train,y_train,alpha=1,lambda=grid)
  bestlam_l <- cvl$lambda.min
  best_fit_l <- glmnet(X_train,y_train,alpha=1,lambda=bestlam_l)
  test_pred_l <- predict(best_fit_l, newx=X_test, s=bestlam_l)
  lasso_test_mse[i] <- mean((y_test - test_pred_l)^2)
}

ridge_test_mse <- ridge_test_mse[2:50]
lasso_test_mse <- lasso_test_mse[2:50]

df <- data.frame(ridge_test_mse, lasso_test_mse)
df <- df %>% pivot_longer(cols = c(ridge_test_mse, lasso_test_mse), names_to = "method", values_to = "t
ggplot(df, aes(x=method, y=test_mse)) +
  geom_boxplot() +
  ggtitle("Ridge and Lasso Test MSEs (beta1_50 = 0.5)") +
  xlab("Method") +
  ylab("MSE") +
  theme(plot.title = element_text(hjust = 0.5))
```



Both lasso and ridge test MSEs seem to be close in their median values, which makes sense. Since all predictors have a non-zero effect on y, the lasso regression doesn't necessarily need to set coefficients to zero. The strength of lasso is more apparent when some predictors have no effect, as it makes coefficients exactly zero. Although in this case this strength is not as apparent as none of the coefficient were defined as zero in the data generation.

Both lasso and ridge seem to have a similar spread in terms of their interquartile range. However, there is an outlier for the ridge regression, this would suggest that one of the seed values generated a particularly bad test MSE for ridge regression.

We should note that in this case, ridge does not completely outperform lasso as in the case where lasso outperformed ridge. This is because certain betas were generated as zero in the previous case. Here we do see overlap between the boxes, and although the median test mse is lower for ridge regression, the difference is not as clear as in the previous case.