Name: CHAN King Yeung

SID: 1155119394

STAT4001 Homework 2

Question 1

given

both and are unbiased estimators

both and are biased estimators

Question 2







1. Comparing with and , the coefficient estimates change substantially given there is a scaling constant on and penalty terms is independent of the column space of .

Comparing with and , the coefficient estimates are scale invariant which is expected since linear transformation does not alter the column space of . In other words, remains unchanged.

Comparing and , the predicated values also changed due to the variant property of ridge regression. Standardising is a recommended way to reduce the effect while scaling .

Comparing and , the predicated values remain the same given the fact that the column space of does not change over the linear transformation on it.

Question 3

Given , we have , and we can prove that minimises at .

1. When , we have . To minimise , we formulate

minimises .

1. When , we plug-in and into

Since which contradicts the minimisation, cannot minimise if .

1. When , we plug-in into

In addition, . We plug-in into

minimises .

Question 4

given , (see Question 1a) and

given

given and

given and (see Question 1b)

given , ,

given

Question 5

1. Please refer to the following console of output (see Appendix for your reference)

|  |
| --- |
| > set.seed(4001)  > x = rnorm(100, 0, 1)  > e = rnorm(100, 0, 0.1) |

1. Please refer to the following console of output (see Appendix for your reference)

|  |
| --- |
| > y = 1 + x + x ^ 2 + x ^ 3 + e |

1. Please refer to the following console of output (see Appendix for your reference)

|  |
| --- |
| > library(glmnet)  >  > X = data.frame(x)  > names(X) = "X"  > for(i in 2:10) {  + X = cbind(X, x ^ i)  + names(X)[i] = paste0("X ^ ", i)  + }  > X = as.matrix(X)  >  > cv = cv.glmnet(X, y, alpha = 1)  > plot(cv)    > cv$lambda.min  [1] 0.1143797  >  > lasso = glmnet(X, y, alpha = 1)  > predict(lasso, type = "coefficient", s = cv$lambda.min)  11 x 1 sparse Matrix of class "dgCMatrix"  1  (Intercept) 1.0570168  X 0.9242384  X ^ 2 0.9292638  X ^ 3 1.0034351  X ^ 4 .  X ^ 5 .  X ^ 6 .  X ^ 7 .  X ^ 8 .  X ^ 9 .  X ^ 10 . |

The optimal value of is .

The resulting coefficient estimates are

1. Please refer to the following console of output (see Appendix for your reference)

|  |
| --- |
| > y = 1 + x ^ 7 + e  >  > cv = cv.glmnet(X, y, alpha = 1)  > plot(cv)    > cv$lambda.min  [1] 4.597186  >  > lasso = glmnet(X, y, alpha = 1)  > predict(lasso, type = "coefficient", s = cv$lambda.min)  11 x 1 sparse Matrix of class "dgCMatrix"  1  (Intercept) 1.620063  X .  X ^ 2 .  X ^ 3 .  X ^ 4 .  X ^ 5 .  X ^ 6 .  X ^ 7 0.970864  X ^ 8 .  X ^ 9 .  X ^ 10 . |

The optimal value of is .

The resulting coefficient estimates are

Appendix

|  |
| --- |
| ## Question 5  # Part a  set.seed(4001)  x = rnorm(100, 0, 1)  e = rnorm(100, 0, 0.1)  # Part b  y = 1 + x + x ^ 2 + x ^ 3 + e  # Part c  library(glmnet)  X = data.frame(x)  names(X) = "X"  for(i in 2:10) {  X = cbind(X, x ^ i)  names(X)[i] = paste0("X ^ ", i)  }  X = as.matrix(X)  cv = cv.glmnet(X, y, alpha = 1)  plot(cv)  cv$lambda.min  lasso = glmnet(X, y, alpha = 1)  predict(lasso, type = "coefficient", s = cv$lambda.min)  # Part d  y = 1 + x ^ 7 + e  cv = cv.glmnet(X, y, alpha = 1)  plot(cv)  cv$lambda.min  lasso = glmnet(X, y, alpha = 1)  predict(lasso, type = "coefficient", s = cv$lambda.min) |