Centro de Investigación en Matemáticas (CIMAT). Unidad Monterrey

Cómputo Estadístico
Tarea 6

Marcelo Alberto Sanchez Zaragoza

24 de noviembre de 2021

1. Problema 1

Es bien sabido que la regresión ridge tiende a dar valores de coeficientes similares a las variables correlacionadas, mientras que lasso puede dar valores de coeficientes totalmente diferentes a las variables correlacionadas. Se explorará esta propiedad en un entorno sencillo.
Supongamos que $n = 20, p = 2, x_{11} = x_{12}, x_{21} = x_{22}$. Además, supongamos que $y_1 + y_2 = 0$ y $x_{11} + x_{21} = 0$ y $x_{12} + x_{22} = 0$, de modo que la estimación del intercepto en minutos cuadros, regresión de Ridge o en el modelo de lasso es cero: $\hat{\gamma} = 0$.

a) Plantea el problema de la optimización con la regresión ridge bajo estas suposiciones.

b) Argumenta que bajo estas suposiciones, las estimaciones de los coeficientes de ridge satisfacen $\hat{\beta}_1 = \hat{\beta}_2$.

c) Planten el problema de la optimización con la regresión lasso bajo estas suposiciones.

d) Argumenta que en este contexto, los coeficientes de lasso $\hat{\beta}_1$ y $\hat{\beta}_2$ no son únicos; es decir, hay muhcas soluciones posibles al problema de optimización en (c). Describe estas soluciones.

Solución

Inciso a)
Tenemos que una forma general de optimización de regresión ridge es igual a:

$$Min: \ \sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \sum_{j=1}^{p}\hat{\beta}x_j)^2 + \lambda\sum_{i=1}^{p}\hat{\beta}_i^2$$

dado que el problema menciona que $\beta_0 = 0$ y que $n = p = 2$ tenemos lo siguiente:

$$Min: \ (y_1 + \hat{\beta}_1 x_{11} - \hat{\beta}_2 x_{12})^2 + (y_2 - \hat{\beta}_1 x_{21} - \hat{\beta}_2 x_{22})^2 + \lambda(\hat{\beta}_1{}^2 + \hat{\beta}_2{}^2)$$

$$f(\hat{\beta}_1, \hat{\beta}_2) = (y_1 + \hat{\beta}_1 x_{11} - \hat{\beta}_2 x_{12})^2 + (y_2 - \hat{\beta}_1 x_{21} - \hat{\beta}_2 x_{22})^2 + \lambda(\hat{\beta}_1{}^2 + \hat{\beta}_2{}^2)$$

Inciso b)
Ahora vamos a derivar la expresión anterior respecto a $\hat{\beta}_1$ y $\hat{\beta}_2$, posteriormente igualar a cero y encontrar expresiones para los beta.
Derivando la expresión vamos tenemos:

$$\frac{\partial f}{\partial \hat{\beta}_1} = 2(y_1 - \hat{\beta}_1 x_{11} - \hat{\beta}_2 x_{12})(-x_{11}) + 2(y_2 - \hat{\beta}_1 x_{21} - \hat{\beta}_2 x_{22})(-x_{21}) + 2\lambda\hat{\beta}_1 = 0$$

dado que $x_{12} = x_{11}$ y $x_{22} = x_{21}$, tenemos:

$$\hat{\beta}_1(x_{11}^2 + x_{21}^2 + \lambda) - y_1x_{11} - y_2x_{21} + \hat{\beta}_2(x_{11}^2 + x_{21}^2) = 0$$
$$\hat{\beta}_1 = \frac{y_1x_{11} + y_2x_{21} - \hat{\beta}_2(x_{11}^2 + x_{21}^2)}{(x_{11}^2 + x_{21}^2 + \lambda)}$$

Analogamente realizamos el mismo procedimiento para $\hat{\beta}_2$ y obtuvimos:

$$\hat{\beta}_2 = \frac{y_1x_{11} + y_2x_{21} - \hat{\beta}_1(x_{11}^2 + x_{21}^2)}{(x_{11}^2 + x_{21}^2 + \lambda)}$$

Observando estos resultados podemos decir que $\hat{\beta}_1 = \hat{\beta}_2$.

Inciso c)
Ahora proponemos un problema de optimización para la regresión lasso, tenemos:

$$Min: \ (y_1 + \hat{\beta}_1x_{11} - \hat{\beta}_2x_{12})^2 + (y_2 - \hat{\beta}_1x_{21} - \hat{\beta}_2x_{22})^2 + \lambda(|\hat{\beta}_1| + |\hat{\beta}_2|)$$

$$g(\hat{\beta}_1, \hat{\beta}_2) = (y_1 + \hat{\beta}_1x_{11} - \hat{\beta}_2x_{12})^2 + (y_2 - \hat{\beta}_1x_{21} - \hat{\beta}_2x_{22})^2 + \lambda(|\hat{\beta}_1| + |\hat{\beta}_2|)$$

Inciso d)

2. Problema 2

Considerando el conjunto de datos *College* de la libreria ISLR, vamos a predecir el número de solicitudes recibidas(APPs) usando las otras variables del conjunto de datos.

a) Divide el conjunto de datos en un conjunto de entrenamiento y un conjunto de prueba y ajusta un modelo lineal usando mínimos cuadrados en el conjunto de entrenamiento. Reporta el error de prueba obtenido.

b) Ajusta un modelo de regresión Ridge sobre el conjunto de entrenamiento, con α elegido por validación cruzada. Reporta el error de prueba obtenido.

c) Ajusta un mmodelo de lasso en el conjunto de entrenamiento, con α elegido por validación cruzada. Reporta el error de prueba obtenido, junto con el número de estimaciones de coeficientes no nulos.

d) Ajusta un modelo PCR en el conjunto de entrenamiento, con M elegido por la validación cruzada. Reporta el error de prueba obtenido, junto con el valor de M seleccionado mediante validación cruzada.

e) Ajusta un modelo PLS en el conjunto de entrenamiento, con M elegido por la validación cruzada. Informe el error de prueba obtenido, junto con el valor de M seleccionado mediante validación cruzada.

f) Comenta los resultados obtenidos. ¿ Con qué precisión podemos predecir el número de solicitudes de estudios universitarios recibidas? ¿ Hay mucha diferencia entre los errores de prueba resultantes de estos cinco enfoques?

Solución

Inciso a)

Realizamos la división de los datos en la siguientes lineas de código.

```
library(ISLR)
set.seed(1)
sum(is.na(College))

[1] 0

length(College[,1])

[1] 777
```

```
train.size = dim(College)[1] / 2 #621
train = sample(1:dim(College)[1], train.size)
length(train)

[1] 388

test = -train
College.train = College[train, ]
College.test = College[test, ]
length(College.test[,1])

[1] 389

length(College.train[,1])

[1] 388
```

Inciso b)

En este caso ajustamos un modelo de regresión Ridge sobre el conjunto de entrenamiento, donde primero encontramos el valor de α.

```
library(glmnet)
train.mat = model.matrix(Apps~., data=College.train)
test.mat = model.matrix(Apps~., data=College.test)
grid = 10^seq(10, -2, length=100)
mod.ridge = cv.glmnet(train.mat,
                      College.train[, "Apps"],
                      alpha=0, lambda=grid,
                      thresh=1e-12)
lambda.best = mod.ridge$lambda.min
lambda.best

[1] 0.01
```

```
ridge.pred = predict(mod.ridge, newx=test.mat, s=lambda.best)
mean((College.test[, "Apps"] - ridge.pred)^2)

[1] 1135714
```

El valor de $\alpha = 0{,}01$ y un error de 1135714.

Inciso c)

En este caso ajustamos un modelo de regresión lasso sobre el conjunto de entrenamiento, donde primero encontramos el valor de α.

```
library(glmnet)
mod.lasso = cv.glmnet(train.mat, College.train[, "Apps"],
                      alpha=1, lambda=grid,
                      thresh=1e-12)
lambda.best = mod.lasso$lambda.min
lambda.best

[1] 0.01

lasso.pred = predict(mod.lasso, newx=test.mat, s=lambda.best)
mean((College.test[, "Apps"] - lasso.pred)^2)

[1] 1135659

mod.lasso = glmnet(model.matrix(Apps~., data=College),
                   College[, "Apps"],
                   alpha=1)
predict(mod.lasso, s=lambda.best, type="coefficients")

19 x 1 sparse Matrix of class "dgCMatrix"
                       s1
(Intercept) -471.39372069
```

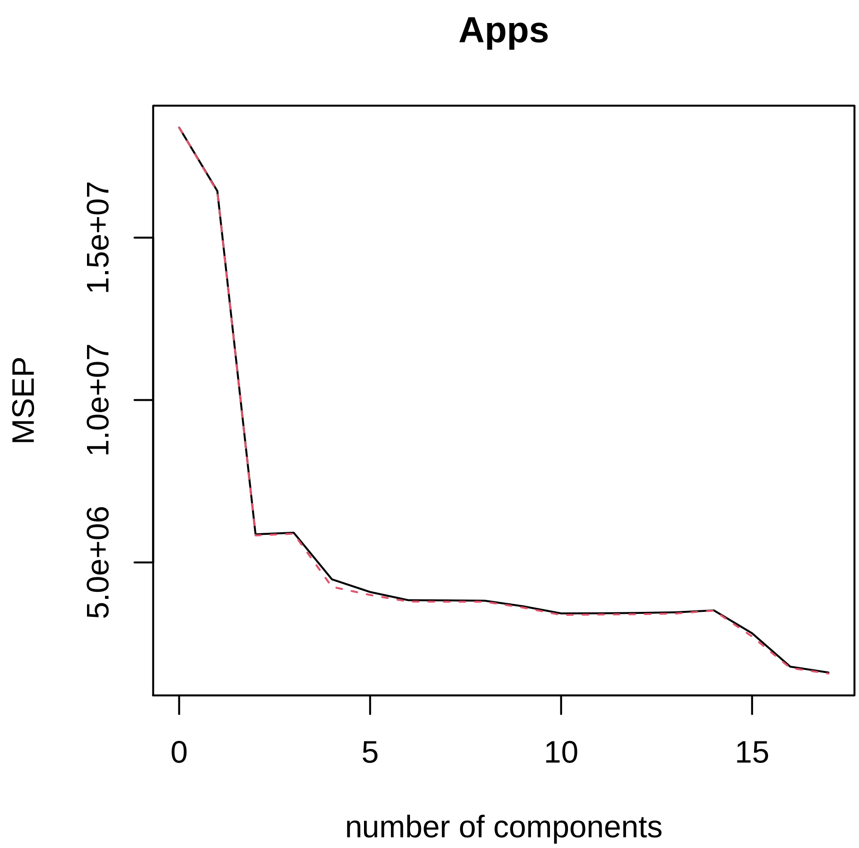
```
(Intercept)     .
PrivateYes  -491.04485135
Accept         1.57033288
Enroll        -0.75961467
Top10perc     48.14698891
Top25perc    -12.84690694
F.Undergrad    0.04149116
P.Undergrad    0.04438973
Outstate      -0.08328388
Room.Board     0.14943472
Books          0.01532293
Personal       0.02909954
PhD           -8.39597537
Terminal      -3.26800340
S.F.Ratio     14.59298267
perc.alumni   -0.04404771
Expend         0.07712632
Grad.Rate      8.28950241
```

El valor de $\alpha = 0{,}01$ y un error de 1135659 y mostramos aquellos coeficientes que son distintos de cero.

Inciso d)

Ajustamos un modelo PCR en el conjunto de entrenamiento y se muestra el gráfico.

```
library(pls)
pcr.fit = pcr(Apps~., data=College.train, scale=T, validation="CV")
validationplot(pcr.fit, val.type="MSEP")
```




```
pcr.pred = predict(pcr.fit, College.test, ncomp=10)
length(pcr.pred)

[1] 389

length(College.test[,'Apps'])

[1] 389
```

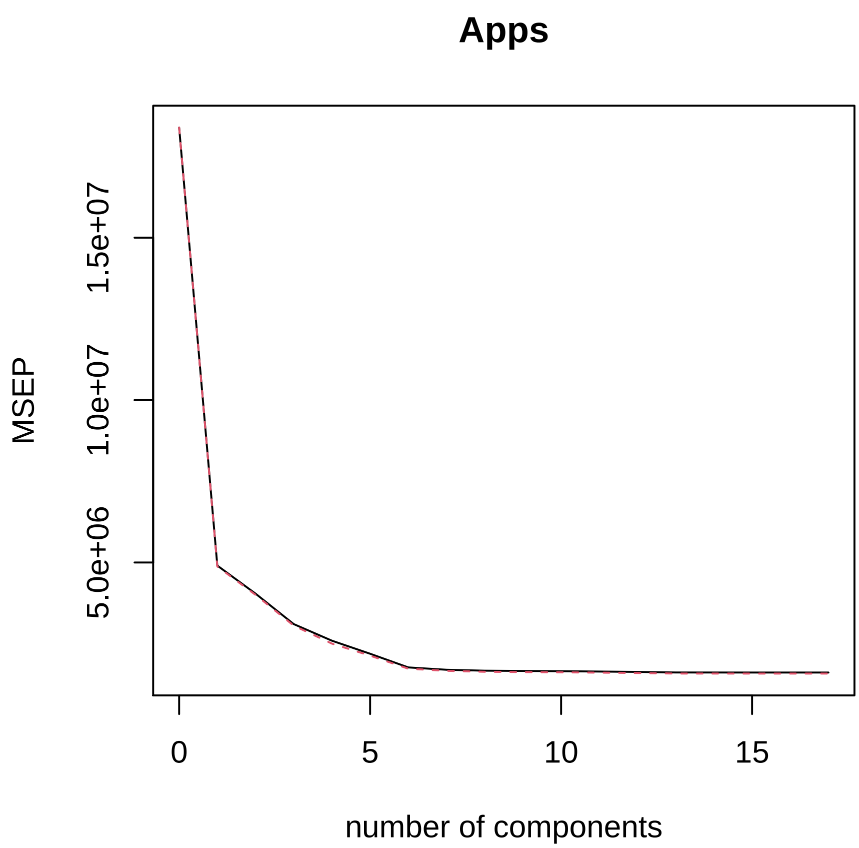
```r
mean((College.test[, "Apps"] - pcr.pred[1:389] )^2)

[1] 1723100
```

El error obtenido fue 1723100, con 10 componentes.

Inciso e)

Ajustamos un modelo PLS en el conjunto de entrenamiento y se muestra el gráfico.

```r
pls.fit = plsr(Apps~., data=College.train, scale=T, validation="CV")
validationplot(pls.fit, val.type="MSEP")
```

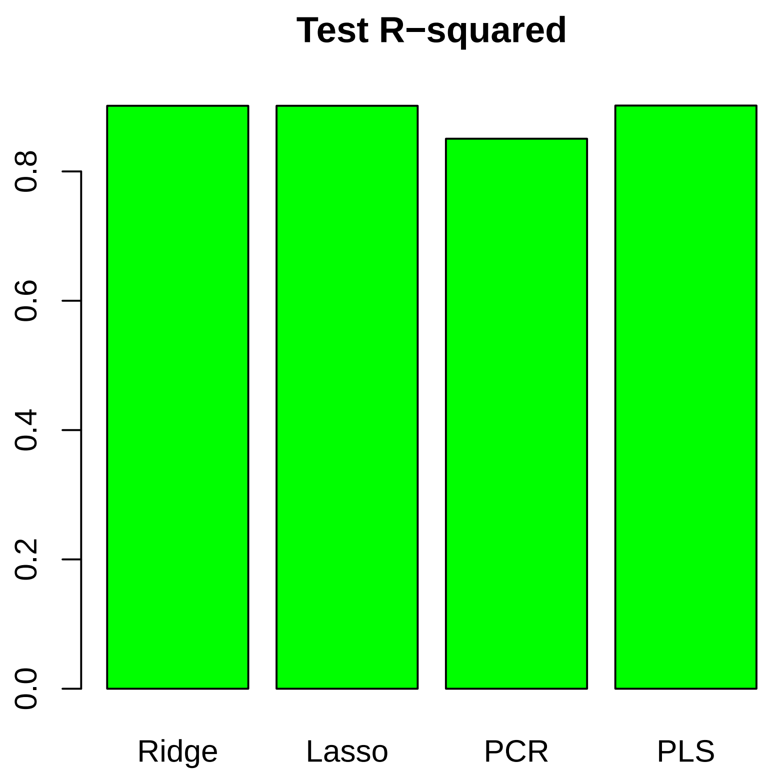



```r
pls.pred = predict(pls.fit, College.test, ncomp=10)
mean((College.test[, "Apps"] - pls.pred[1:389] )^2) #156

[1] 1131661
```

El error obtenido fue 1131661, con 10 componentes.

Inciso f)

```
test.avg = mean(College.test[, "Apps"])
#lm.test.r2 = 1 - mean((College.test[, "Apps"] - lm.pred)^2) /mean((College.test[, "
ridge.test.r2 = 1 - mean((College.test[, "Apps"] -
                  ridge.pred)^2) /mean((College.test[, "Apps"]
                      - test.avg)^2)

lasso.test.r2 = 1 - mean((College.test[, "Apps"] -
                  lasso.pred)^2) /mean((College.test[, "Apps"]
                    - test.avg)^2)

pcr.test.r2 = 1 - mean((College.test[, "Apps"] -
                  pcr.pred[1:389] )^2) /mean((College.test[, "Apps"]
                    - test.avg)^2)

pls.test.r2 = 1 - mean((College.test[, "Apps"] -
                  pls.pred[1:389] )^2) /mean((College.test[, "Apps"]
                    - test.avg)^2)

barplot(c(ridge.test.r2, lasso.test.r2, pcr.test.r2, pls.test.r2),
        col="green", names.arg=c("Ridge", "Lasso", "PCR", "PLS"),
        main="Test R-squared")
```



En el gráfico observamos que el mejor resultados dio para poder predecir el número de solicitudes de estudios universitarios es PCR ya que su R^2 fue menor que todos los demás, igual podemos observar que los valores para Ridge, Lasso y PLS se observan muy parecidas entre ellas.