

Lista 10

Matheus Cougias

12/03/2021

Leitura dos dados

```
dados <- read.delim("boston_corrected.txt")
dados <- subset(dados, select = -c(OBS., TOWN, TOWN., TRACT, LON, LAT, CMEDV))
dt <- data.frame(MEDV=log(dados$MEDV), CRIM=log(dados$CRIM), ZN=log(dados$ZN+1),
                INDUS=log(dados$INDUS), CHAS=log(dados$CHAS+1), NOX=log(dados$NOX),
                RM=log(dados$RM), AGE=log(dados$AGE), DIS=log(dados$DIS),
                RAD=log(dados$RAD), TAX=log(dados$TAX), PTRATIO=log(dados$PTRATIO),
                B=log(dados$B), LSTAT=log(dados$LSTAT))
```

Modelo de regressão linear múltipla Através do modelo de regressão linear múltipla, tivemos um R^2 preditivo de 75,48254%.

```
require(car)
```

```
## Loading required package: car
```

```
## Loading required package: carData
```

```
modelo <- lm(MEDV ~ CRIM + ZN + INDUS + CHAS + NOX + RM + AGE + DIS + RAD +
            TAX + PTRATIO + B + LSTAT, data=dt)
modeloSTEP <- step(modelo)

1 - 1/vif(modeloSTEP)

Y <- as.matrix(dt$MEDV)
SQT <- sum((Y - mean(Y))^2)

X <- model.matrix(modeloSTEP)

H <- X%*%solve(t(X)%*%X)%*%t(X)
res <- Y - H%*%Y
aux <- 1 - diag(H)
res_del <- res/(1-diag(H))
SQRes <- sum(res_del^2)
(R2pred <- 1 - SQRes/SQT)
```

Modelo com aplicação de Ridge Regression Através do modelo com aplicação de Ridge Regression, tivemos um R^2 preditivo de 75,51781%.

```

beta <- solve(t(X)%*%X)%*%t(X)%*%Y
Identidade <- diag(c(0,rep(1, nrow(beta)-1)))

R2predFun <- function(lmbd)
{
  Identidade <- diag(c(0,rep(1, nrow(beta)-1)))
  beta <- solve(t(X)%*%X + lmbd*Identidade)%*%t(X)%*%Y
  H <- X%*%solve(t(X)%*%X + lmbd*Identidade)%*%t(X)
  res <- Y - H%*%Y
  aux <- 1 - diag(H)
  res_del <- res/(1-diag(H))
  SQRes <- sum(res_del^2)
  R2pred <- 1 - SQRes/SQT
  return(R2pred)
}

saida <- optimize(R2predFun, lower=0.001, upper=1.2, maximum = TRUE)
lambda <- saida$maximum
(saida$objective)

```

Modelo com aplicação de LASSO Através do modelo com aplicação de LASSO, o valor ótimo de d encontrado foi de 2,470599 e com ele tivemos um R^2 preditivo de 75,59047%.

```

invXtX <- solve(t(X)%*%X)
beta.hat <- invXtX%*%t(X)%*%Y
d.max <- sum(abs(beta.hat[-1]))

R2predFun2 <- function(d.abs)
{
  C <- matrix(sign(beta.hat), ncol=length(beta.hat))
  C[,1] <- 0
  D <- as.matrix(d.abs)
  beta <- beta.hat - invXtX%*%t(C)%*%solve(C%*%invXtX%*%t(C))%*(C%*%beta.hat - D)
  apply(cbind(beta.hat[-1], beta[-1]), 2, function(x) sum(abs(x)))

  while( sum( abs(beta[-1]) ) - d.abs > 1e-3){
    C <- rbind(C, as.numeric(sign(beta)))
    C[,1] <- 0 # Restricao nao se aplica ao beta_0
    D <- rbind(D, d.abs)
    beta <- beta.hat - invXtX%*%t(C)%*%solve(C%*%invXtX%*%t(C))%*(C%*%beta.hat - D)
  }

  apply(cbind(beta.hat[-1], beta[-1]), 2, function(x) sum(abs(x)))

  H <- X%*%invXtX%*%(t(X) - t(C)%*%solve(C%*%invXtX%*%t(C))%*%C%*%invXtX%*%t(X))

  yhat <- X%*%beta
  res <- Y - yhat
  press <- sum( (res/(1-diag(H)))^2 )
  R2pred <- 1 - press/SQT
}

saida2 <- optimize(R2predFun2, lower=0.001, upper=d.max, maximum = TRUE)

```

```
d.abs <- saida2$maximum
(d.abs)
(saida2$objective)
```

Modelo utilizando o pacote glmnet

```
require(glmnet)
```

```
## Loading required package: glmnet
```

```
## Warning: package 'glmnet' was built under R version 4.0.4
```

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

```
## Loaded glmnet 4.1-1
```

```
print('Ridge Regression')
```

```
## [1] "Ridge Regression"
```

```
modelo <- cv.glmnet(X[,-1], Y, nfolds=20, alpha=0)
coef(modelo)
```

```
## 13 x 1 sparse Matrix of class "dgCMatrix"
```

```
##              1
## (Intercept)  4.159972213
## CRIM        -0.015568422
## ZN          -0.003953024
## CHAS        0.163092575
## NOX         -0.171368767
## RM          0.648450068
## AGE        -0.001708924
## DIS        -0.043066981
## RAD         0.017549346
## TAX        -0.118299243
## PTRATIO    -0.472507698
## B          0.052292119
## LSTAT      -0.267262891
```

```
modelo$lambda.min
```

```
## [1] 0.0336058
```

```
print('LASSO')
```

```
## [1] "LASSO"
```

```
modelo <- cv.glmnet(X[,-1], Y, nfolds=20, alpha=1)
coef(modelo)
```

```
## 13 x 1 sparse Matrix of class "dgCMatrix"
##              1
## (Intercept)  4.811711581
## CRIM         .
## ZN           .
## CHAS         0.110313898
## NOX          .
## RM           0.391453264
## AGE          0.002359847
## DIS         -0.003940686
## RAD          .
## TAX         -0.111285326
## PTRATIO     -0.391422969
## B           0.043640299
## LSTAT       -0.403705807
```

```
modelo$lambda.min
```

```
## [1] 0.000377502
```

```
print('Elastic Net')
```

```
## [1] "Elastic Net"
```

```
modelo <- cv.glmnet(X[,-1], Y, nfolds=20)
coef(modelo)
```

```
## 13 x 1 sparse Matrix of class "dgCMatrix"
##              1
## (Intercept)  4.947561198
## CRIM         .
## ZN           -0.001595342
## CHAS         0.127065484
## NOX          -0.073403935
## RM           0.401452176
## AGE          0.016421406
## DIS         -0.041467765
## RAD          0.004561131
## TAX         -0.133095256
## PTRATIO     -0.422799719
## B           0.049017059
## LSTAT       -0.416357004
```

```
modelo$lambda.min
```

```
## [1] 0.000377502
```

Análise de resultados Se compararmos os resultados dos betas acima gerados, percebemos que o Ridge Regression impossibilita o corte de variáveis do modelo, fazendo com que algumas sejam mantidas ainda que com seu coeficiente bem abaixo das demais. Possivelmente esse não corte de variáveis sem bom poder preditivo faz com que o valor do R^2 preditivo seja prejudicado na utilização do Ridge Regression.

Por outro lado, ao utilizarmos o método de LASSO, percebemos que as variáveis CRIM, ZN, NOX, AGE, DIS e RAD foram retiradas do modelo, provavelmente por serem redundantes ou então diminuírem sua capacidade preditiva. Essa ideia é suportada a partir do momento em que a capacidade preditiva do modelo LASSO é levemente superior à capacidade do modelo de Ridge Regression.

Por último, se observarmos o modelo do Elastic Net, percebemos que o método LASSO interferiu no seu corte de variáveis, mantendo as mesmas variáveis que anteriormente restaram no modelo. Porém, podemos também perceber que os coeficientes das variáveis mantidas foram levemente alterados.

Análise do comportamento do LASSO Como descrito anteriormente, o método LASSO possui a capacidade de zerar o coeficiente de algumas variáveis, fazendo com que aquelas que não alteram a capacidade preditiva do modelo sejam cortadas da regressão, então ela funciona basicamente como um selecionador de variáveis para o modelo. Por isso, acredito que em modelos onde não seja viável o corte de variáveis, o método LASSO possa sair em desvantagem em relação à Ridge Regression.