Ridge and Lasso Regression

Süleyman Buğra Gülsoy

23 12 2021

Introduction:

The aim of the objective is predicting the number of car seat sales using the data with the best possible model

Methodology:

Statistical methods that i will be using are multiple linear regression and ridge,lasso regression Since the data has more than 1 variables multiple linear regression is suitable. Thus, to avoid overfit and have a better interpretation of data, Ridge and Lasso models is going to be used for reducing model variance that can be caused by using too many variables with shrinking estimators.

Data Set:

The data that I will be using is “Carseats” which is in the “ISLR” package The data tracks the unit sales for car seats with respect to 10 variables

Sales: unit sales in thousands CompPrice: price charged by competitor at each location Income: community income level in 1000s of dollars Advertising: local ad budget at each location in 1000s of dollars Population: regional pop in thousands Price: price for car seats at each site ShelveLoc: Bad, Good or Medium indicates quality of shelving location Age: age level of the population Education: ed level at location Urban: Yes/No US: Yes/No

The Modelling:

```
library("ISLR")
library("glmnet")

## Warning: package 'glmnet' was built under R version 4.1.3

## Zorunlu paket yükleniyor: Matrix

## Loaded glmnet 4.1-3

library("leaps")
seat_data = Carseats
nacount = c()
```

```
for(i in 1:ncol(seat_data)){

  nacount = c(nacount,sum(is.na(seat_data[,i])))
}

nacount
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0
```

The data does not have any NA values

```
sum(duplicated(seat_data))
```

```
## [1] 0
```

The data does not have any duplicated value

```
str(seat_data)
```

```
## 'data.frame':    400 obs. of  11 variables:
##  $ Sales      : num  9.5 11.22 10.06 7.4 4.15 ...
##  $ CompPrice  : num  138 111 113 117 141 124 115 136 132 132 ...
##  $ Income     : num  73 48 35 100 64 113 105 81 110 113 ...
##  $ Advertising: num  11 16 10 4 3 13 0 15 0 0 ...
##  $ Population : num  276 260 269 466 340 501 45 425 108 131 ...
##  $ Price      : num  120 83 80 97 128 72 108 120 124 124 ...
##  $ ShelveLoc  : Factor w/ 3 levels "Bad","Good","Medium": 1 2 3 3 1 1 3 2
3 3 ...
##  $ Age        : num  42 65 59 55 38 78 71 67 76 76 ...
##  $ Education  : num  17 10 12 14 13 16 15 10 10 17 ...
##  $ Urban      : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 2 2 1 1 ...
##  $ US         : Factor w/ 2 levels "No","Yes": 2 2 2 2 1 2 1 2 1 2 ...
```

All the categorical variables are in factor form thus we dont need to manipulate it Since R will handle the dummy variables automaticly

Now the data set is ready for regression

```
set.seed (8270)

split_data = sample(1:nrow(seat_data), 0.8*nrow(seat_data))

train_seat = seat_data[split_data,]

test_seat = seat_data[-split_data,]
```

The data has splitted into test and training with respect to %80,%20 rule

Firstly multiple linear regression model with using all the variables in data

```
lm_1 = lm(Sales~., data = train_seat)

summary(lm_1)

##
## Call:
## lm(formula = Sales ~ ., data = train_seat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.92800 -0.71545  0.01873  0.69185  2.70956
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.5111335  0.6617412   8.328 2.73e-15 ***
## CompPrice        0.0944238  0.0045463  20.769  < 2e-16 ***
## Income           0.0150782  0.0020562   7.333 2.00e-12 ***
## Advertising      0.1191268  0.0120177   9.913  < 2e-16 ***
## Population       0.0002695  0.0004018   0.671    0.503
## Price           -0.0947313  0.0029550 -32.058  < 2e-16 ***
## ShelveLocGood    4.9030621  0.1704880  28.759  < 2e-16 ***
## ShelveLocMedium  1.8430375  0.1381559  13.340  < 2e-16 ***
## Age             -0.0452180  0.0034889 -12.961  < 2e-16 ***
## Education       -0.0273317  0.0216803  -1.261    0.208
## UrbanYes         0.1494414  0.1240574   1.205    0.229
## USYes           -0.1623912  0.1647065  -0.986    0.325
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.008 on 308 degrees of freedom
## Multiple R-squared:  0.8739, Adjusted R-squared:  0.8694
## F-statistic:   194 on 11 and 308 DF,  p-value: < 2.2e-16

predic_linear = predict(lm_1,test_seat)

sqrt(mean((predic_linear-test_seat$Sales)^2))

## [1] 1.076357
```

The mean square error is 1.076357

Secondly, conducting a multiple linear regression model with using only better variables

To have a subset of variables we use “regsubsets” function to try to compare all possible subsets

```
subset = regsubsets(Sales~., data = train_seat, method = "backward", nvmax =
10)

summary(subset)
```

```
## Subset selection object
## Call: regsubsets.formula(Sales ~ ., data = train_seat, method =
"backward",
##     nvmax = 10)
## 11 Variables  (and intercept)
##                 Forced in Forced out
## CompPrice           FALSE      FALSE
## Income              FALSE      FALSE
## Advertising         FALSE      FALSE
## Population          FALSE      FALSE
## Price               FALSE      FALSE
## ShelveLocGood       FALSE      FALSE
## ShelveLocMedium     FALSE      FALSE
## Age                 FALSE      FALSE
## Education           FALSE      FALSE
## UrbanYes            FALSE      FALSE
## USYes               FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: backward
##          CompPrice Income Advertising Population Price ShelveLocGood
## 1  ( 1 )  " "       " "    " "         " "        " "   "*"
## 2  ( 1 )  " "       " "    " "         " "        "*"   "*"
## 3  ( 1 )  "*"       " "    " "         " "        "*"   "*"
## 4  ( 1 )  "*"       " "    "*"         " "        "*"   "*"
## 5  ( 1 )  "*"       " "    "*"         " "        "*"   "*"
## 6  ( 1 )  "*"       " "    "*"         " "        "*"   "*"
## 7  ( 1 )  "*"       "*"    "*"         " "        "*"   "*"
## 8  ( 1 )  "*"       "*"    "*"         " "        "*"   "*"
## 9  ( 1 )  "*"       "*"    "*"         " "        "*"   "*"
## 10  ( 1 ) "*"       "*"    "*"         " "        "*"   "*"
##          ShelveLocMedium Age Education UrbanYes USYes
## 1  ( 1 )  " "             " " " "       " "      " "
## 2  ( 1 )  " "             " " " "       " "      " "
## 3  ( 1 )  " "             " " " "       " "      " "
## 4  ( 1 )  " "             " " " "       " "      " "
## 5  ( 1 )  " "             "*" " "       " "      " "
## 6  ( 1 )  "*"             "*" " "       " "      " "
## 7  ( 1 )  "*"             "*" " "       " "      " "
## 8  ( 1 )  "*"             "*" "*"       " "      " "
## 9  ( 1 )  "*"             "*" "*"       "*"      " "
## 10  ( 1 ) "*"             "*" "*"       "*"      "*"

which.max(summary(subset)$adjr2)

## [1] 10

which.min(summary(subset)$cp)

## [1] 7

which.min(summary(subset)$bic)
```

```
## [1] 7
```

```
subset2 = regsubsets(Sales~., data = train_seat, method = "forward", nvmax =
10)
```

```
which.max(summary(subset2)$adjr2)
```

```
## [1] 10
```

```
which.min(summary(subset2)$cp)
```

```
## [1] 7
```

```
which.min(summary(subset2)$bic)
```

```
## [1] 7
```

Both forward and backward methods suggest using the same 7 variables with respect to cp and bic minimum

```
coef(subset,7) == coef(subset2,7)
```

```
##     (Intercept)       CompPrice          Income     Advertising
Price
##           FALSE           FALSE           FALSE            TRUE
FALSE
##   ShelveLocGood ShelveLocMedium             Age
##           FALSE            TRUE           FALSE
```

Conducting our second multiple linear regression with best subset of our variables

```
lm_2 = lm(Sales~ CompPrice + Income + Advertising + Price + ShelveLoc +
Age,data = train_seat)
summary(lm_2)
```

```
##
## Call:
## lm(formula = Sales ~ CompPrice + Income + Advertising + Price +
##     ShelveLoc + Age, data = train_seat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7965 -0.6689 -0.0160  0.7171  2.8292
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.295652   0.553685   9.564  < 2e-16 ***
## CompPrice        0.093930   0.004519  20.786  < 2e-16 ***
## Income           0.014996   0.002047   7.327 2.03e-12 ***
## Advertising      0.113420   0.008455  13.415  < 2e-16 ***
## Price           -0.094550   0.002953 -32.014  < 2e-16 ***
```

```
## ShelveLocGood    4.888267   0.170256  28.711  < 2e-16 ***
## ShelveLocMedium  1.842115   0.137932  13.355  < 2e-16 ***
## Age             -0.045317   0.003480 -13.024  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.01 on 312 degrees of freedom
## Multiple R-squared:  0.872,  Adjusted R-squared:  0.8691
## F-statistic: 303.6 on 7 and 312 DF,  p-value: < 2.2e-16
```

```
predic_linear_2 = predict(lm_2,test_seat)

sqrt(mean((predic_linear_2-test_seat$Sales)^2))
```

```
## [1] 1.07292
```

The mean square error is 1.07292

Now conducting a ridge regression model with using all variables

```
x_train = model.matrix(Sales~.,train_seat,)[,-1]

y_train = train_seat$Sales
```

To decide best lambda value for our model we use cross validation

```
optimal = cv.glmnet(x_train,y_train, alpha = 0)

optimal$lambda.min
```

```
## [1] 0.1417235
```

```
ridge_model = glmnet(x_train, y_train, alpha = 0, lambda = optimal$lambda.min
)


x_test = model.matrix(Sales~.,test_seat,)[,-1]

y_test = test_seat$Sales


predic_ridge = predict(ridge_model, newx = x_test)

sqrt((mean(predic_ridge - y_test)^2))
```

```
## [1] 0.01823733
```

The mean square error with ridge regression is 0.01823733

Now conductin our second ridge regression model with using only the best subset of variables

```
x_train_2 = model.matrix(Sales~CompPrice + Income + Advertising + Price +
ShelveLoc + Age ,train_seat,)[,-1]

y_train_2 = train_seat$Sales

optimal_2 = cv.glmnet(x_train_2,y_train_2, alpha = 0)

optimal_2$lambda.min

## [1] 0.1417235

ridge_model_2 = glmnet(x_train_2, y_train_2, alpha = 0, lambda =
optimal_2$lambda.min )


x_test_2 = model.matrix(Sales~CompPrice + Income + Advertising + Price +
ShelveLoc + Age,test_seat,)[,-1]

y_test_2 = test_seat$Sales


predic_ridge_2 = predict(ridge_model_2, s= optimal_2$lambda.min, newx =
x_test_2)

sqrt((mean(predic_ridge_2 - y_test_2)^2))

## [1] 0.0159226
```

The mean square error with second ridge regression model is 0.0159226

Lasso Regression with using all of the variables

```
optimal_3 = cv.glmnet(x_train,y_train, alpha = 1)

optimal_3$lambda.min

## [1] 0.006426991

lasso_model = glmnet(x_train, y_train, alpha = 1, lambda =
optimal_3$lambda.min )

sum(coef(lasso_model) != 0)

## [1] 12

predic_lasso = predict(lasso_model, newx = x_test)

sqrt((mean(predic_lasso - y_test)^2))

## [1] 0.01651172
```

The mean square error with lasso model is 0.01699377

Conducting the lasso model with using only the best subset of variables

```
optimal_4 = cv.glmnet(x_train_2,y_train_2, alpha = 1)

optimal_4$lambda.min

## [1] 0.004861783

lasso_model_2 = glmnet(x_train_2, y_train_2, alpha = 1, lambda =
optimal_4$lambda.min )

sum(coef(lasso_model_2) != 0)

## [1] 8

predic_lasso_2 = predict(lasso_model_2,newx = x_test_2)

sqrt((mean(predic_lasso_2 - y_test_2)^2))

## [1] 0.01361432
```

The mean square error with second lasso model is 0.01361432

To Conclude:

We have conduected in total 6 models: And the error for each model is

Multiple linear regression with all variables : 1.076357

Multiple linear regression with best subset of variables : 1.07292

Ridge regression with all variables : 0.01823733

Ridge regression with best subset of variables : 0.0159226

Lasso regression with all variables : 0.01699377

Lasso regression with best subset of variables : 0.01361432

As it seen the best model for prediction is Lasso regression model Which has only the best subset of 7 variables since it has the lowest error between all the models

References:

https://www.statology.org

https://rstatisticsblog.com

https://rpubs.com