BUS 41201 Homework 3 Assignment

4/22/2020

Group 29: Hye-Min Jung, Jeong Lim Kim, Jayoung Kang

"I pledge my honor that I have not violated the Honor Code during this assignment"

Question 1

We want to build a predictor of customer ratings from product reviews and product attributes. For these questions, you will fit a LASSO path of logistic regression using a binary outcome:

Fit a LASSO model with only product categories. The start code prepares a sparse design matrix of 142 product categories. What is the in-sample R2 for the AICc slice of the LASSO path? Why did we use standardize FALSE? (1 point)

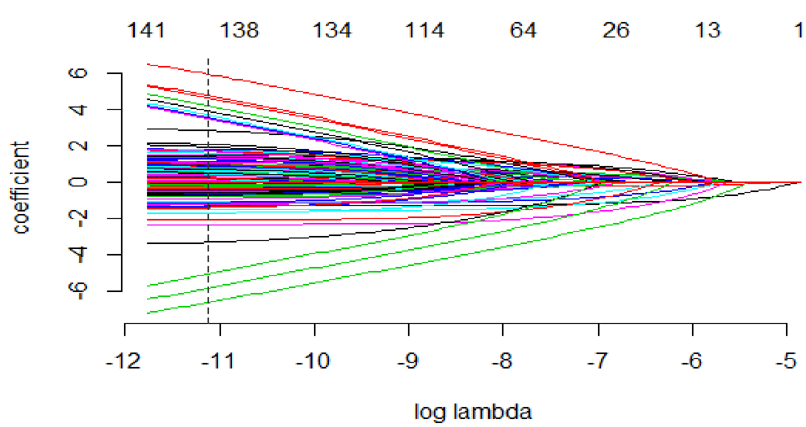
- In-sample R^2: 0.1048737
- Rationale for using standardize=FALSE: Standardization allows us to have different variables scaled so that they are more interpretable and features with larger scales do not dominate another. However, in this case we only have dummy variables regarding the category of goods. Using standardization would put more penalty on common categories and less penalty on rare categories, which might be undesirable for this case.

```
# Let's define the binary outcome

# Y=1 if the rating was 5 stars

# Y=0 otherwise

Y<-as.numeric(data$Score==5)

# (a) Use only product category as a predictor

library(gamlr)

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following object is masked from 'package:tidyr':
##
##     expand
```

```r
source("naref.R")

class(data$Prod_Category)
```

[1] “factor”

```r
# Since product category is a factor, we want to relevel it for the LASSO.
# We want each coefficient to be an intercept for each factor level rather than a
contrast.
# Check the extra slides at the end of the lecture.
# look inside naref.R. This function relevels the factors for us.

data$Prod_Category<-naref(data$Prod_Category)

# Create a design matrix using only products
products<-data.frame(data$Prod_Category)

x_cat<-sparse.model.matrix(~., data=products)[,-1]

# Sparse matrix, storing 0's as .'s
# Remember that we removed intercept so that each category
# is standalone, not a contrast relative to the baseline category

colnames(x_cat)<-levels(data$Prod_Category)[-1]

# let's call the columns of the sparse design matrix as the product categories
# Let's fit the LASSO with just the product categories
lasso1<- gamlr(x_cat, y=Y, standardize=FALSE, family="binomial", lambda.min.ratio=
1e-3)
plot(lasso1)
```




```r
# AICc selected coef
beta <- coef(lasso1)
nrow(beta)
```

[1] 143

```r
# lambda
log(lasso1$lambda[which.min(AICc(lasso1))])

seg91
```

-11.13165

```r
# No. of non-zero coef
sum(beta!=0)
```

[1] 139

```r
# find R2 (method 1)
source("deviance.R")
pred <- predict(lasso1, newdata = x_cat, type="response")
R2(Y, pred, family = "binomial")
```

[1] 0.1048737

```r
# find R2 (method 2)
summary(lasso1)$r2[which.min(AICc(lasso1))]
```

binomial gamlr with 142 inputs and 100 segments.

[1] 0.1048737

Question 2

Fit a LASSO model with both product categories and the review content (i.e. the frequency of occurrence of words). Use AICc to select lambda. How many words were selected as predictive of a 5 star review? Which 10 words have the most positive effect on odds of a 5 star review? What is the interpretation of the coefficient for the word `discount'? (3 points)

- AICc lambda: -8.334091
- Number of words selected: 1022
- Top 10 words: worried, plus, excellently, find, grains, hound, sliced, discount, youd, doggies
- Interpretation for 'discount' coefficient: A unit increase in the frequency of the word 'discount' in the review increases the odds of receiving 5 stars by 1055.256 times.

```
# Fit a LASSO with all 142 product categories and 1125 words
spm<-sparseMatrix(i=doc_word[,1],
                  j=doc_word[,2],
                  x=doc_word[,3],
                  dimnames=list(id=1:nrow(data),
                  words=words))

# 13319 reviews using 1125 words
dim(spm)
```

[1] 13319 1125

```
# new matrix with category and words
x_cat2<-cbind(x_cat,spm)

# Lasso with product category and words
lasso2 <- gamlr(x_cat2, y=Y,lambda.min.ratio=1e-3,family="binomial")
log(lasso2$lambda[which.min(AICc(lasso2))])

seg89
```

-8.334091

```
# AICc selected coef
beta2 <- coef(lasso2)
sum(beta2!=0)
```

[1] 1154

```
# Number of words selected
sum(beta2[(ncol(x_cat)+1):nrow(beta2)]!=0)
```

[1] 1022

```r
# Top 10 words
beta3<-coef(lasso2)[(ncol(x_cat)+1):nrow(beta2),]
beta3[order(beta3,decreasing=TRUE)[1:10]]

worried        plus  excellently        find       grains        hound
```

10.516545 9.175674 8.375464 7.422606 7.250390 7.179146 sliced discount youd doggies 7.045506 6.961539 6.842082 6.766085

```r
# 'discount' coef
beta3['discount']
```

discount 6.961539

```r
exp(6.961539)
```

[1] 1055.256

```r
exp(beta3['discount'])
```

discount 1055.256

Question 3

Continue with the model from Question 2. Run cross-validation to obtain the best lambda value that minimizes OOS deviance. How many coefficients are nonzero then? How many are nonzero under the 1se rule? (1 point)

- No. of nonzero coefficients for OOS deviance min: 974

- No. of nonzero coefficients for 1se rule: 831

```
set.seed(123)

cv.fit <- cv.gamlr(x_cat2,
                   y=Y,
                   lambda.min.ratio=1e-3,
                   family="binomial",
                   verb=TRUE)
```

fold 1,2,3,4,5,done.

```
beta4<-coef(cv.fit, select="min") ## min cv selection
beta5<-coef(cv.fit) ## 1se rule; see ?cv.gamlr

sum(beta4!=0)
```

[1] 988

```
sum(beta5!=0)
```

[1] 831

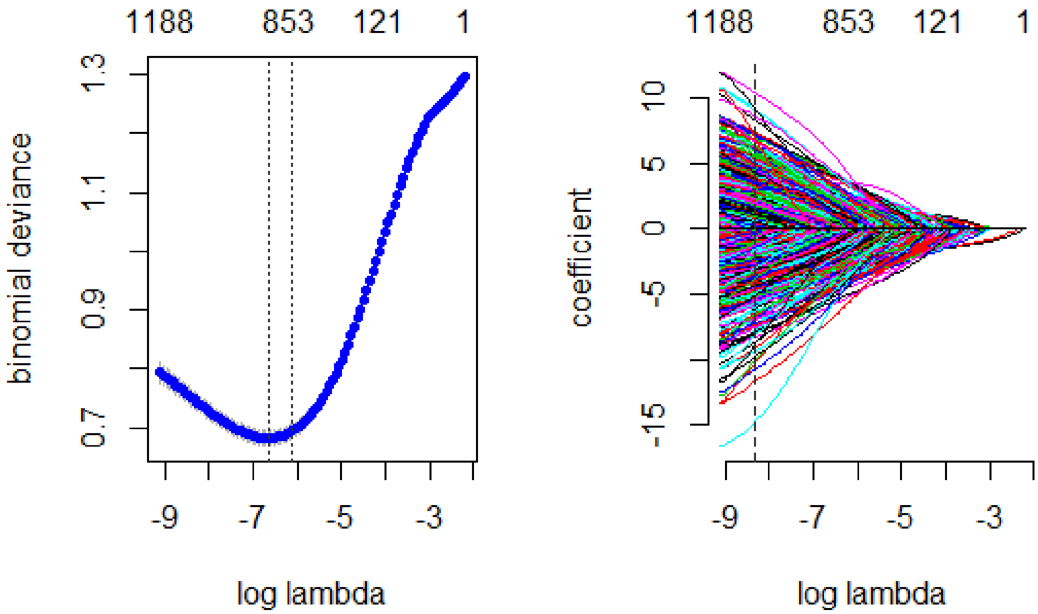
```
log(cv.fit$lambda.min)
```

[1] -6.659484

```
log(cv.fit$lambda.1se)
```

[1] -6.101282

```
## plot them together
par(mfrow=c(1,2))
plot(cv.fit)
plot(cv.fit$gamlr)
```


1188
853
121
1
binomial deviance
0.7
0.9
1.1
1.3
-9
-7
-5
-3
log lambda
1188
853
121
1
coefficient
-15
-5
0
5
10
-9
-7
-5
-3
log lambda