BUS 41201 Homework 3 Assignment

Veronika Rockova
Veronika.Rockova@ChicagoBooth.edu

4/22/2020

Amazon Reviews

The dataset consists of 13 319 reviews for selected products on Amazon from Jan-Oct 2012. Reviews include product information, ratings, and a plain text review. The data consists of three tables:

##Review subset.csv is a table containing, for each review, its

- ProductId: Amazon ASIN product code
- UserId: ID of the reviewer
- Score: numeric 1-5 (the number of stars)
- Time: date of the review
- Summary: review summary in words
- Nrev: number of reviews by the user
- Length: number of words in the review
- Prod Category: Amazon product category
- Prod Group: Amazon product group

Word freq.csv

is a simple triplet matrix of word counts from the review text including

- Review ID: the row index of Review subset.csv
- Word ID: the row index of words.csv
- Times Word: how many times the word occurred in the review

Words.csv

contains 1125 alphabetically ordered words that occur in the reviews.

```r
library(knitr) # library for nice R markdown output


# READ REVIEWS

data<-read.table("Review_subset.csv",header=TRUE)
dim(data)
```

[1] 13319 9

```r
# 13319 reviews
# ProductID: Amazon ASIN product code
# UserID:  id of the reviewer
# Score: numeric from 1 to 5
# Time: date of the review
# Summary: text review
# nrev: number of reviews by this user
# Length: length of the review (number of words)

# READ WORDS

words<-read.table("words.csv")
words<-words[,1]
length(words)
```

[1] 1125

```r
#1125 unique words

# READ text-word pairings file

doc_word<-read.table("word_freq.csv")
names(doc_word)<-c("Review ID","Word ID","Times Word" )
# Review ID: row of the file  Review_subset
# Word ID: index of the word
# Times Word: number of times this word occurred in the text
```

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:

$$Y = 1 \quad \text{for 5 stars} \tag{1}$$
$$Y = 0 \quad \text{for less than 5 stars.} \tag{2}$$

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)

```r
# Let's define the binary outcome
```

```r
# 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
```

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

# Cast the product category as a factor
data$Prod_Category<-as.factor(data$Prod_Category)

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)
```

```r
lasso1_summary <- summary(lasso1)
index <- which.min(lasso1_summary$aicc)
best_lambda <- lasso1_summary$lambda[index]
lambda_r2 <- lasso1_summary$r2[index]
cat("The lambda of the AICc slice is", best_lambda, "with in-sample R2 of", lambda_r2)
```

The lambda of the AICc slice is 1.464148e-05 with in-sample R2 of 0.1048737

Here, we use standardize FALSE because we have converted the data into a sparse matrix, where the zero values are not stored in memory. When we standardize a sparse matrix, we convert it into a dense format because standardization (subtracting the mean and dividing by the standard deviation) changes all zero values into non-zero values (unless all non-zero values of a feature have the same value, which is uncommon).

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)

```
# 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))

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

[1] 13319 1125

```
x_cat2<-cbind(x_cat,spm)

lasso2 <- gamlr(x_cat2, y=Y,lambda.min.ratio=1e-3,family="binomial")
```

```
lasso2_coef <- coef(lasso2)[-1,]
words_only <- tail(lasso2_coef, -142)
nonzero_words <- words_only[words_only!=0]
length(nonzero_words)
```

Out of 1125 words, there are 1022 that are selected as predictive of a 5 star review.

```
sorted_words <- sort(words_only)
tail(sorted_words, 10)
```

The 10 words that have the most positive effect on odds of a 5 star review are: worried, plus, excellently, find, grains, sliced, discount, youd, doggies

The coefficient for the word "discount" is 6.961539. It can be interpreted as: The word "discount" being included in a review multiplies odds of the product getting a 5 star review by the exponential of the coefficient, which is approximately 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)

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

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

```r
oos_coef <- coef(cv.fit, select="min")
words_only_oos <- tail(oos_coef, -142)
nonzero_words_oos <- words_only_oos[words_only_oos!=0]
cat("There are", length(nonzero_words_oos), "coefficients that are nonzero.\n")


lse_coef <- coef(cv.fit)
words_only_lse <- tail(lse_coef, -142)
nonzero_words_lse <- words_only_lse[words_only_lse!=0]
cat("Under 1se rule, there are", length(nonzero_words_lse), "coefficients that are nonzero.")
```

There are 839 coefficients that are nonzero. Under 1se rule, there are 733 coefficients that are nonzero.