BUS 41201 Homework 3 Assignment

Group 24: Shihan Ban, Yi Cao, Shri Lekkala, Ningxin Zhang

9 April 2024

Setup

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.

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


# READ REVIEWS

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

[1] 13319 9

```
# 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

```
#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.

```
# 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)

source("naref.R")

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

class(data$Prod_Category)
```

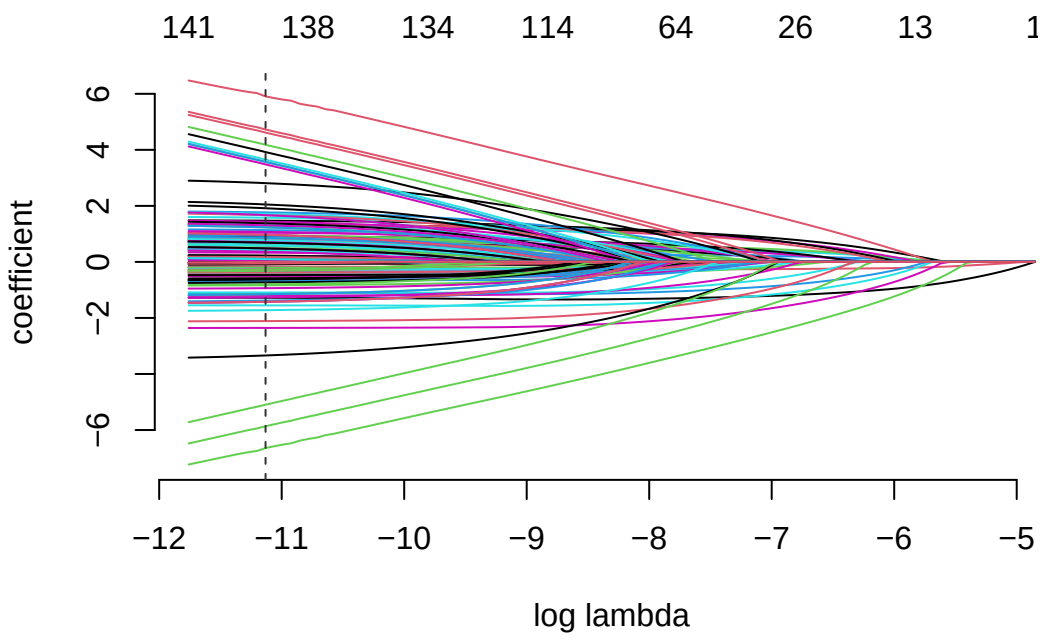
[1] "factor"

```
# 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)
summary_output <- summary(lasso1)
```

binomial gamlr with 142 inputs and 100 segments.

```
plot(lasso1)
```



What is the in-sample R2 for the AICc slice of the LASSO path?

```
# Find the lowest AICc
best_aicc_index <- which.min(summary_output$aicc)
# corresponding r^2
best_r2 <- summary_output$r2[best_aicc_index]

# print it
print(best_r2)
```

```
## [1] 0.1048737
```

The in-sample R2 for the AICc slice of the LASSO path is 0.1048737.

Why did we use standardize FALSE?

We did not standardize as the predictors in the model are binary indicators representing product categories. Since each product category is either present (1) or absent (0), their scale is already normalized in a sense that they are on the same scale. Standardizing would not add value and could potentially mislead the model fitting process by artificially inflating the variance of these binary predictors.

Question 2

Fit a LASSO model with both product categories and the review content (i.e. the frequency of occurrence of words).

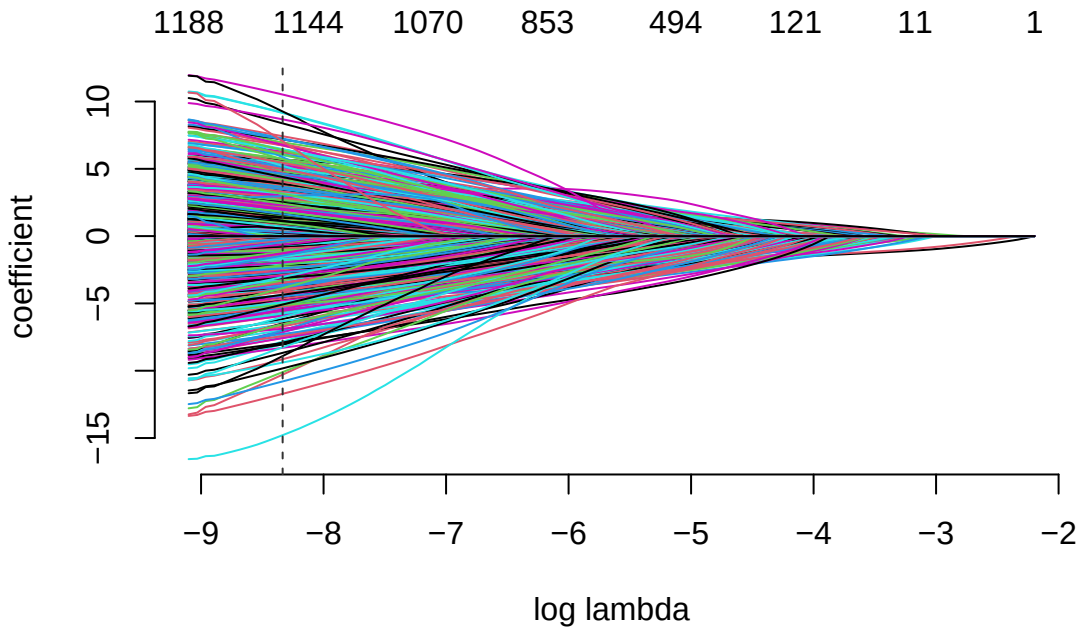
```
# Fit a LASSO with all 142 product categories and 1125 words

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")
summary_output_2 <- summary(lasso2)
```

binomial gamlr with 1267 inputs and 100 segments.

```
plot(lasso2)
```



Use AICc to select lambda.

```r
# Find the index of the best AICc
best_aicc_index_2 <- which.min(summary_output_2$aicc)

# Find the optimal lambda using the best AICc index
optimal_lambda <- lasso2$lambda[best_aicc_index_2]
optimal_lambda
```

```
##        seg89
## 0.0002401874
```

So the optimal lambda is 0.0002401874, selected via AICc.

How many words were selected as predictive of a 5 star review?

```r
# Get the coefficients at the optimal lambda
coefficients_lasso2 <- lasso2$beta[, best_aicc_index_2]

# Since the first set of predictors are the product categories
# we only want to count the non-zero coefficients associated with words.
word_coefficients <- coefficients_lasso2[-(1:142)]

# Count the non-zero coefficients for words
num_predictive_words <- sum(word_coefficients != 0)

# Output the number of words
num_predictive_words
```

```
## [1] 1022
```

So there are 1022 words out of the 1125 possible ones that were selected as predictive of a 5 star review

Which 10 words have the most positive effect on odds of a 5 star review?

```r
# Sort the coefficients to find the largest positive values
sorted_word_coefs <-sort(word_coefficients, decreasing = TRUE)

# Select the top 10 words with the most positive effect
top_10_positive_words <-head(sorted_word_coefs, 10)

# Output the result
top_10_positive_words
```

```
##     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
```

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

What is the interpretation of the coefficient for the word 'discount'?

The coefficient value of 6.96 is relatively large compared to other coefficients in the model.
This indicates a strong correlation between mentioning the word "discount" in reviews and the likelihood of receiving a 5-star rating. A positive coefficient signifies a positive correlation, suggesting that as the frequency of the word "discount" increases in reviews, the probability of receiving a 5-star rating also increases.
Practically, this coefficient implies that reviews mentioning "discount" are more likely to receive a 5-star rating compared to those that do not mention "discount," assuming all other factors remain constant. This could reflect customer satisfaction with discounted products or a tendency to give positive reviews when discounts are involved.
While many factors influence the likelihood of receiving a 5-star rating, mentioning discounts emerges as a significant contributing factor.

Question 3

Continue with the model from Question 2. Run cross-validation to obtain the best lambda value that minimizes OOS deviance.

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

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

```
# Find the best lambda that minimizes the OOS deviance
best_lambda_cv <- cv.fit$lambda.min

# Output the best lambda
best_lambda_cv
```

```
## [1] 0.00137444
```

So the best lambda that minimizes OOS deviance is 0.001473767, obtained via cross-validation

How many coefficients are nonzero then?

```
# Extract the coefficients at the best lambda from the cross-validation
coefficients_cv <- coef(cv.fit, s = "min")

# Count the number of non-zero coefficients
nonzero_coefficients <- sum(coefficients_cv != 0)

# Output the number of non-zero coefficients
nonzero_coefficients
```

```
## [1] 974
```

So there are 974 nonzero coefficients.

How many are nonzero under the 1se rule?

```
# Find the best lambda via the 1se rule
best_lambda_1se <- cv.fit$lambda.1se

# Output the best lambda
best_lambda_1se
```

```
## [1] 0.001948234
```

```r
# Extract the coefficients via the 1se rule
coefficients_1se <- coef(cv.fit, s = "1se")

# Count the number of non-zero coefficients
nonzero_coefficient_1se <- sum(coefficients_1se != 0)

# Output the number of non-zero coefficients
nonzero_coefficient_1se
```

```
## [1] 831
```

So there are 831 nonzero coefficients under the 1se rule.