HW3_Mengdi

Mengdi Hao

2024-04-07

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)

ANSWER:

The in-sample R2 is 0.1049, which means around 10.49% of variations in dependent variable is captured by the LASSO model.

"standardize=FALSE" indicates that the variable should not be standardized before the LASSO regularization is applied. Standardization transforms variables to have a mean of zero and a standard deviation of one. However, here our design matrix consists entirely of categorical variables, standardization is unnecessary or meaningful, as these variables are already on a comparable scale. We have already set a reference level of "NA".

```
# ***** AMAZON REVIEWS

# READ REVIEWS

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

# 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")
```

```r
words <- words[,1]
# length(words)
#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

# 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
```

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

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

# look inside naref.R; it applies to every factor variable:
#   > factor(x,levels=c(NA,levels(x)),exclude=NULL)
# Since product category is a factor, we want to re-level it for the LASSO.
# We want each coefficient to be an intercept for each factor level rather than a contrast.

# In R, categorical variables are often converted to factor variables for use in models.
# A factor has levels, which represent the different categories.
# The reference level is the baseline category against which the other categories are compared.
# In regression models, coefficients for factor levels are typically represented as the change
# relative to this baseline category.
# levels(data$Prod_Category)
data$Prod_Category <- naref(data$Prod_Category)
# use NA as reference
# levels(data$Prod_Category)

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

# Sparse matrix, storing 0's as .'s
# We removed intercept so that each category is standalone, not a contrast relative to the baseline cat
x_cat <-sparse.model.matrix(~., data=products)[,-1]

# let's call the columns of the sparse design matrix as the product categories
colnames(x_cat) <- levels(data$Prod_Category)[-1]
```

```r
# Let's fit the LASSO with just the product categories
# "100 segments" in the result means there are 100 lambda values constructed
lasso1 <- gamlr(x_cat,y=Y,standardize=FALSE,family="binomial", lambda.min.ratio=1e-3)

# obtain the optimal lambda index
optimal_lambda_index <- which.min(AICc(lasso1))

# extract optimal R2 based on AICc
lasso1_summary <- summary(lasso1)
```

```
##
## binomial gamlr with 142 inputs and 100 segments.
```

```r
in_sample_R2 <- lasso1_summary$r2[optimal_lambda_index] # R2=0.1049
cat("In-Sample R2:", in_sample_R2, "\n")
```

```
## In-Sample R2: 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)

ANSWER:

The optimal log(lambda) using AICc chosen for the LASSO model with both product categories and the review content is -8.334.

The number of words selected as predictive of a 5-star review is 1022. The top 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".

The top 10 words identified as predictive of a 5-star review in the LASSO logistic regression model suggest positive correlations with high customer satisfaction. Words like "excellently" and "plus" directly enhance a review's positive tone, while others like "grains" or "hound" may relate to specific product qualities appreciated in certain categories like pet food or health foods. "Discount" indicates that price reductions positively impact review ratings, enhancing perceived value. Each word's presence increases the log-odds of a review being 5 stars, reflecting nuanced consumer sentiments and expectations.

The coefficient for the word "discount" is 6.962. This shows that comments with the word "discount" in it is more likely to be a 5-star rating. For each additional time "discount" appears, the odds of a 5-star review is multiplied by 1055.26. However, such a large coefficient may be unusual and could suggest over-fitting. This could be due to the result of LASSO selection, which keep variables that have a large effect even if they are not reliable predictors.

```r
# (2) Fit a LASSO with all 142 product categories and 1125 words

library(gamlr)
spm <- sparseMatrix(i=doc_word[,1],j=doc_word[,2],x=doc_word[,3],dimnames=list(id=1:nrow(data),words=wo
# dim(spm) # 13319 reviews using 1125 words

x_cat2 <- cbind(x_cat,spm) # totally 1125+142=1267 covariates
lasso2 <- gamlr(x_cat2,
```

```
                    y=Y,
                    lambda.min.ratio=1e-3,
                    family="binomial")
# summary(lasso2)

# optimal log(lambda) based on AICc
optimal_log_lambda <- log(lasso2$lambda[which.min(AICc(lasso2))])
cat("Optimal log(lambda) Under AICc:", optimal_log_lambda, "\n")
```

```
## Optimal log(lambda) Under AICc: -8.334091
```

```
# extract coefs
lasso2_beta <- coef(lasso2) # extract the coefficients from the lasso2 model

# Number of non-zero coefficients for words only
num_words_selected <- sum(lasso2_beta[-(1:143)] != 0)  # Exclude intercept and product categories
cat("The number of words selected as predictive:", num_words_selected, "\n")
```

```
## The number of words selected as predictive: 1022
```

```
# Extracting word coefficients and sorting them to find the top 10 words
word_coefficients <- lasso2_beta[-(1:143)]  # Exclude intercept and product categories

# assign names to the coefficients: this is feasible because the order of coefficients
# in "word_coefficients" is the same as the order in "words"
names(word_coefficients) <- words

top_words <- word_coefficients[names(sort(word_coefficients, decreasing = TRUE)[1:10])]  # Get top 10 w
cat("Top 10 words with the most positive effect on odds of a 5 star review:\n")
```

```
## Top 10 words with the most positive effect on odds of a 5 star review:
```

```
print(top_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
```

```
# Find the coefficient for the word 'discount'
discount_coef <- word_coefficients["discount"] # 6.961539
odds_multiplier <- exp(discount_coef)

cat("Coefficient for the word 'discount':", discount_coef, "\n")
```

```
## Coefficient for the word 'discount': 6.961539
```

```
cat("Odds multiplier for the word 'discount':", odds_multiplier, "\n")
```

```
## Odds multiplier for the word '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)

ANSWER:

The best log(lambda) is around -6.5. With CV LASSO, the number of non-zero coefficients selected is around 950-980. Under 1se rule, the number drops to around 810-850.

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

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

```
## CV min deviance selection
# cv.min gives us the coefficients under the optimally chosen lambda
# this gives us the refitted regression coefficients on the entire dataset with
# the optimal lambda
cv.min <- coef(cv.fit, select="min")
num_select_1 <- sum(cv.min!=0) ## around 950-1000 with log(lam) around -6.5 (its random!)
# compute the logarithm of the optimal lambda value
optimal_lambda_1 <- log(cv.fit$lambda.min)

cat("Optimal log(lambda):", optimal_lambda_1, "\n")
```

```
## Optimal log(lambda): -6.659484
```

```
cat("Number of non-zero coefficients selected:", num_select_1, "\n")
```

```
## Number of non-zero coefficients selected: 988
```

```
## CV 1se selection (the default)
cv.1se <- coef(cv.fit)
# CV 1se selection usually gives us the null model
num_select_2 <- sum(cv.1se!=0) ## 897, 831, 811
# compute log(lambda:1se)
optimal_lambda_2 <- log(cv.fit$lambda.1se) # -6.31, -6.10, -6.03

cat("Optimal log(lambda) under 1se rule:", optimal_lambda_2, "\n")
```

```
## Optimal log(lambda) under 1se rule: -6.031506
```

```
cat("Number of non-zero coefficients selected:", num_select_2, "\n")
```

```
## Number of non-zero coefficients selected: 811
```