HW3

2024-04-08

```
knitr::opts_chunk$set(echo = TRUE)
```

```
# ***** AMAZON REVIEWS

# 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

```r
# 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 relevel it for the LASSO. We want each coefficient to

#levels(data$Prod_Category)

data$Prod_Category<-naref(data$Prod_Category)

#levels(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
# We removed intercept so that each category is standalone, not a contrast relative to the baseline cat

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)

null_deviance <- deviance(glm(Y ~ 1, family = binomial(link = "logit")))
min_aicc_index <- which.min(AICc(lasso1))
best_lambda <- lasso1$lambda[min_aicc_index]
best_deviance <- lasso1$deviance[min_aicc_index]
R_2 <- 1 - (best_deviance / null_deviance)
R_2
```

```
##     seg91
## 0.1048737
```

Based on product categories as predictors and utilizing the AICc-LASSO method, accounts for approximately 10.49% of the variability observed in the consumer ratings.

When standardize is set to FALSE, it means that the predictors are not adjusted to have a mean of 0 and a standard deviation of 1. This adjustment, known as standardization or scaling, is commonly applied to continuous predictors to ensure they are on a comparable scale. However, for categorical predictors like product categories, standardization is typically not applied because it can alter the interpretation of the coefficients.

Question 2

```
library(gamlr)

spm<-sparseMatrix(i=doc_word[,1],j=doc_word[,2],x=doc_word[,3],dimnames=list(id=1:nrow(data),words=words

dim(spm)
```

```
## [1] 13319  1125
```

```
# 13319 reviews using 1125 words

x_cat2<-cbind(x_cat,spm)

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

```
## Warning in gamlr(x_cat2, y = Y, lambda.min.ratio = 0.001, family = "binomial", :
## numerically perfect fit for some observations.
```

```
best_lambda2 <- log(lasso2$lambda[which.min(AICc(lasso2))])
best_lambda2
```

```
##     seg89
## -8.334091
```

The optimal value of the regularization parameter lambda chosen by the LASSO model using AICc is approximately -8.334.

```
coefficients <- coef(lasso2, lambda = best_lambda2)
num_predictive_words <- sum(coefficients[-1][-(1:(ncol(x_cat) - 1))] != 0)
num_predictive_words
```

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

Out of the total words considered in the analysis, 1022 words were selected as predictive of a 5-star review by the LASSO model.

```
words <- rownames(coefficients)[(ncol(x_cat) + 1):length(coefficients)]
coefficients_word <- coefficients[(ncol(x_cat) + 1):length(coefficients)]
top_10_indices <- head(order(coefficients_word, decreasing = TRUE), 10)
top_10_words <- words[top_10_indices]
top_10_words
```

```
##  [1] "worried"     "plus"        "excellently" "find"        "grains"
##  [6] "hound"       "sliced"      "discount"    "youd"        "doggies"
```

```
coefficient_discount <- coefficients["discount", ]
coefficient_discount
```

```
## [1] 6.961539
```

The coefficient of 6.961539 associated with the word 'discount' in a review suggests that the presence of this word positively influences the odds of the review being rated 5 stars. In other words, it contributes positively to the predictive capability of the LASSO model.

Question 3

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

```
## Warning in gamlr(x, y, ...): numerically perfect fit for some observations.
```

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

```
coefficients_best_lambda <- coef(cv.fit, select = 'min')
nonzero_coef_best_lambda_count <- sum(coefficients_best_lambda[-1] != 0)
nonzero_coef_best_lambda_count
```

```
## [1] 987
```

```
coefficients_1se <- coef(cv.fit, select = '1se')
nonzero_coef_1se_count <- sum(coefficients_1se[-1] != 0)
nonzero_coef_1se_count
```

```
## [1] 810
```