BIOST 546 HW 3

My-Anh Doan

2023-02-16

```
# set global options for code chunks
knitr::opts_chunk$set(message = FALSE, warning = FALSE, collapse = TRUE)
knitr::opts_knit$set(root.dir = rprojroot::find_rstudio_root_file())

library(dplyr)
library(ggplot2)
library(caret)
library(pROC)
library(knitr)
library(cowplot)
library(MASS)
library(class)
```

As in HW2, we will perform binary classification on the Breast Cancer Wisconsin (Diagnostic) Data Set in the csv file `wdbc.data`. The dataset describes characteristics of the cell nuclei present in n (sample size) images. Each image has multiple attributes, which are described in detail in `wdbc.names`. This time, however, you will predict the attribute in column 2, which we denote by Y, given the columns $\{3, 4, \ldots, 32\}$, which we denote by $X_1, ..., X_30$. The variable Y represents the diagnosis (M = malignant, B = benign).

1. Data exploration and Simple Logistic Regression

- 1a. Describe the data: sample size n, number of predictors p, and the number of observations in each class.

```
# load data
wdbc <- read.csv("./dataset/wdbc.data", header = FALSE, stringsAsFactors = TRUE)
wdbc_set <- wdbc[, -1] %>%
    rename(diagnosis = 1)

str(wdbc_set)
## 'data.frame':    569 obs. of  31 variables:
##  $ diagnosis: Factor w/ 2 levels "B","M": 2 2 2 2 2 2 2 2 2 2 ...
##  $ V3       : num  18 20.6 19.7 11.4 20.3 ...
##  $ V4       : num  10.4 17.8 21.2 20.4 14.3 ...
##  $ V5       : num  122.8 132.9 130 77.6 135.1 ...
##  $ V6       : num  1001 1326 1203 386 1297 ...
##  $ V7       : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
##  $ V8       : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
##  $ V9       : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
##  $ V10      : num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
##  $ V11      : num  0.242 0.181 0.207 0.26 0.181 ...
##  $ V12      : num  0.0787 0.0567 0.06 0.0974 0.0588 ...
##  $ V13      : num  1.095 0.543 0.746 0.496 0.757 ...
##  $ V14      : num  0.905 0.734 0.787 1.156 0.781 ...
##  $ V15      : num  8.59 3.4 4.58 3.44 5.44 ...
##  $ V16      : num  153.4 74.1 94 27.2 94.4 ...
##  $ V17      : num  0.0064 0.00522 0.00615 0.00911 0.01149 ...
##  $ V18      : num  0.049 0.0131 0.0401 0.0746 0.0246 ...
##  $ V19      : num  0.0537 0.0186 0.0383 0.0566 0.0569 ...
##  $ V20      : num  0.0159 0.0134 0.0206 0.0187 0.0188 ...
##  $ V21      : num  0.03 0.0139 0.0225 0.0596 0.0176 ...
##  $ V22      : num  0.00619 0.00353 0.00457 0.00921 0.00511 ...
##  $ V23      : num  25.4 25 23.6 14.9 22.5 ...
##  $ V24      : num  17.3 23.4 25.5 26.5 16.7 ...
##  $ V25      : num  184.6 158.8 152.5 98.9 152.2 ...
##  $ V26      : num  2019 1956 1709 568 1575 ...
##  $ V27      : num  0.162 0.124 0.144 0.21 0.137 ...
##  $ V28      : num  0.666 0.187 0.424 0.866 0.205 ...
##  $ V29      : num  0.712 0.242 0.45 0.687 0.4 ...
##  $ V30      : num  0.265 0.186 0.243 0.258 0.163 ...
##  $ V31      : num  0.46 0.275 0.361 0.664 0.236 ...
##  $ V32      : num  0.1189 0.089 0.0876 0.173 0.0768 ...

# check for missing values; if returns 0, no missing data in data set
which(complete.cases(wdbc_set) == FALSE)
## integer(0)

wdbc_summary <- wdbc_set %>%
  count(diagnosis)

kable(wdbc_summary, caption = "Total observations by diagnosis class")
```

Table 1: Total observations by diagnosis class

diagnosis	n
B	357
M	212

The `wdbc` data set has a sample size of $n = 569$ and $p = 30$ predictors. There are `n wdbc_summary[1, 2]` observations in the benign class and `n wdbc_summary[2, 2]` observations in the malignant class (as shown in *Table 1*).

- 1b. Divide the data into a training set of 400 observations and a test set.

```
set.seed(2)
random_sample <- sample(1:nrow(wdbc_set), size = 400, replace = FALSE)

# split data into training and test sets
wdbc_train <- wdbc_set[random_sample, ]
wdbc_test <- wdbc_set[-random_sample, ]
```

- 1c. Normalize your predictors, i.e. for each variable X_j remove the mean and make each variable's standard deviation 1. Explain why you should perform this step separately in the training set and test set.

- 1d. Compute the correlation matrix of your training predictors (command `cor`) and plot it (e.g. command `ggcorrplot` in the library `ggcorrplot`). Inspect the correlation matrix and explain what type of challenges this data set may present?

- 1e. Fit a simple logistic regression model to predict Y given $X_1, ..., X_3$0. Inspect and report the correlation between the variables X_1 and X_3; and the magnitude of their coefficient estimates $\hat{\beta}_1$, $\hat{\beta}_3$ with regard to the other coefficients of the model. Comment on their values and relate this to what we have seen in class.

- 1f. Use the glm previously fitted and the Bayes rule to compute the predicted outcome $\hat{Y}$ from the associated probability estimates (computed with `predict`) both on the training and the test set. Then compute the confusion table and prediction accuracy (rate of correctly classified observations) both on the training and test set. Comment on the results.

2. Ridge Logistic Regression

- 2a.From the normalized training set and validation set, contruct a data matrix X (numeric) and an outcome vector y (factor).
- 2b. On the training set, run a ridge logistic regression model with `glmnet` (with the argument `family = "binomial"`) to predict Y given $X_1, ..., X_3$0. Use the following grid of values for lambda: `10^seq(5, -18, length = 100)`
- 2c. Plot the values of the coefficients β_1, β_3 (y-axis) in function of `log(lambda)` (x-axis). Comment on the result.
- 2d. Apply 10-fold cross-validation with the previously defined grid of values for lambda. Report the value of lambda that minimizes the CV misclassification error. We will refer to it as the optimal lambda. Plot the misclassification error (y-axis) in function of `log(lambda)` (x-axis). Use `cv.glmnet` with the arguments `family = "binomial"` and `type.measure = "class"`.
- 2e. Report the number of coefficients β_j that are different from 0 for the ridge model with the optimal lambda. Comment on the results.
- 2f. Use the regularized glm previously gitted (with the optimal lambda) and the Bayes rule to compute the predicted outcome $\hat{Y}$ from the associated probability estimates on both the training and test sets. Then compute the confusion table and prediction accuracy both on the training and test set. Comment on the results. Use the command `predict` with argument ‘type = “response”.
- 2g. Plot the ROC curve, computed on the test set.
- 2h. Compute an estimate of the area under the ROC curve (AUC).

3. Lasso Logistic Regression

Repeat the sub-problems 2b to 2h using a lasso regression model instead of a ridge logistic regression model.

4. Discussion

Discuss the performance of the simple glm, ridge glm, and lasso glm on the Breast Cancer Wisconsin Data Set in terms of prediction accuracy (on the training and test set) and model interpretability.