LASSO multinomial logistic regression

Variable selection for high dimensional data

Miao Cai

9/22/2019

Contents

1 Introduction to penalized regression

Penalized logistic regression imposes a penalty to the logistic model for having too many variables. This results in shrinking the coefficients of the less contributive variables toward zero. This is also known as regularization.

The most commonly used penalized regression include:

1. **Ridge regression**: variables with minor contribution have their coefficients close to zero. However, all the variables are incorporated in the model. This is useful when all variables need to be incorporated in the model according to domain knowledge.
2. **LASSO regression** (Least Absolute Shrinkage and Selection Operator): the coefficients of some less contributive variables are forced to be exactly zero. Only the most significant variables are kept in the final model.
3. **Elastic net regression**: the combination of ridge and lasso regression. It shrinks some coefficients toward zero (like ridge regression) and set some coefficients to exactly zero (like lasso regression).

Above contents sourced from http://www.sthda.com/english/articles/36-classification-methods-essentials/149-penalized-logistic-regression-essentials-in-r-ridge-lasso-and-elastic-net/.

2 Theories of Penalized logistic regression

2.1 Logistic regression theory

A logistic regression assumes the binary outcome variable Y has Bernoulli distribution with the parameter p. The log odds of p equals a linear combination of predictor variables $x_1, x_2, \cdots, x_k$ and parameters $\beta_1, \beta_2, \cdots, \beta_k$.

$$
\begin{aligned}
Y &\sim \text{Bernoulli}(p) \\
\log \frac{p}{1-p} &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k
\end{aligned}
$$

After a little bit of algebra, we have:

$$p_i = \frac{\exp(\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_k x_{ki})}{1 + \exp(\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_k x_{ki})} \tag{1}$$

To get the parameter estimates for $\beta_1, \beta_2, \cdots, \beta_k$, we need to maximize the likelihood function L:

$$L = \prod_{i=1}^{N} p_i^{y_i} (1 - p_i)^{(1-y_i)}$$

or the log likelihood function LL:

$$\begin{aligned} LL &= \log\left(\prod_{i=1}^{N} p_i^{y_i} (1 - p_i)^{(1-y_i)}\right) \\ &= \sum_{i=1}^{N} \Big(y_i \log p_i + (1 - y_i) \log(1 - p_i)\Big) \end{aligned} \tag{2}$$

Where the p_i can be substituted by the expression in Equation 1. The purpose of MLE is to maximize the log likelihood in Equation 2 and obtain MLE estimates for $\beta_1, \beta_2, \cdots, \beta_k$, or equivalently minimizing the negative log likelihood $-LL$.

2.2 Penalized logistic regression

- For LASSO, also known as $L1$-regularization, it minimizes

$$-LL + \lambda \sum_{i=1}^{K} |\beta_i|$$

- For Ridge regression, also known as $L2$-regularization, it minimizes

$$-LL + \lambda \sum_{i=1}^{K} \beta_i^2$$

- For elastic net, which is a combined version of ridge and LASSO, it minimizes the

$$-LL + \lambda\left((1 - \alpha) \sum_{i=1}^{K} \beta_i^2 + \alpha \sum_{i=1}^{K} |\beta_i|\right)$$

For all the above three penalized likelihoods, they are a balance between the likelihood function LL and penalty term. On one hand, the parameters $\beta_1, \beta_2, \cdots, \beta_k$ should be maximized to maximize the likelihood function. On the other hand, the parameters $\beta_1, \beta_2, \cdots, \beta_k$ should be minimized to minimize the penalty term. λ is a hyperparameter that controls the extent of regularization, and it could be determined by cross validation (CV).

3 LASSO syntax with `glmnet` package

```r
x <- model.matrix(diabetes ~ ., train_data)[,-1]
y <- ifelse(train_data$diabetes == "pos", 1, 0)

glmnet(x, y, family = "binomial", alpha = 1, lambda = NULL)
```

- x: matrix of predictor variables

- `y`: the response or outcome variable, which is a binary variable.
- `family`: the response type. Use "binomial" for a binary outcome variable
- `alpha`: the elasticnet mixing parameter. Allowed values include:
  - "1": for lasso regression
  - "0": for ridge regression
  - a value between 0 and 1 (say 0.3) for elastic net regression.
- `lamba`: a numeric value defining the amount of shrinkage. Should be specify by analyst.

4 Application using `glmnet`

4.1 Data importing

```r
library(data.table)
library(dplyr)
d = fread("pd_speech_features/pd_speech_features.csv", skip = 1) %>% # multilogit_LASSO/
  mutate(y_cat = case_when(PPE < 0.762 ~ 0,
                             PPE >= 0.762 & PPE < 0.809 ~ 1,
                             PPE >= 0.809 & PPE < 0.834 ~ 2,
                             PPE >= 0.834  ~ 3),
         PPE = NULL,
         app_entropy_shannon_10_coef = NULL)
```

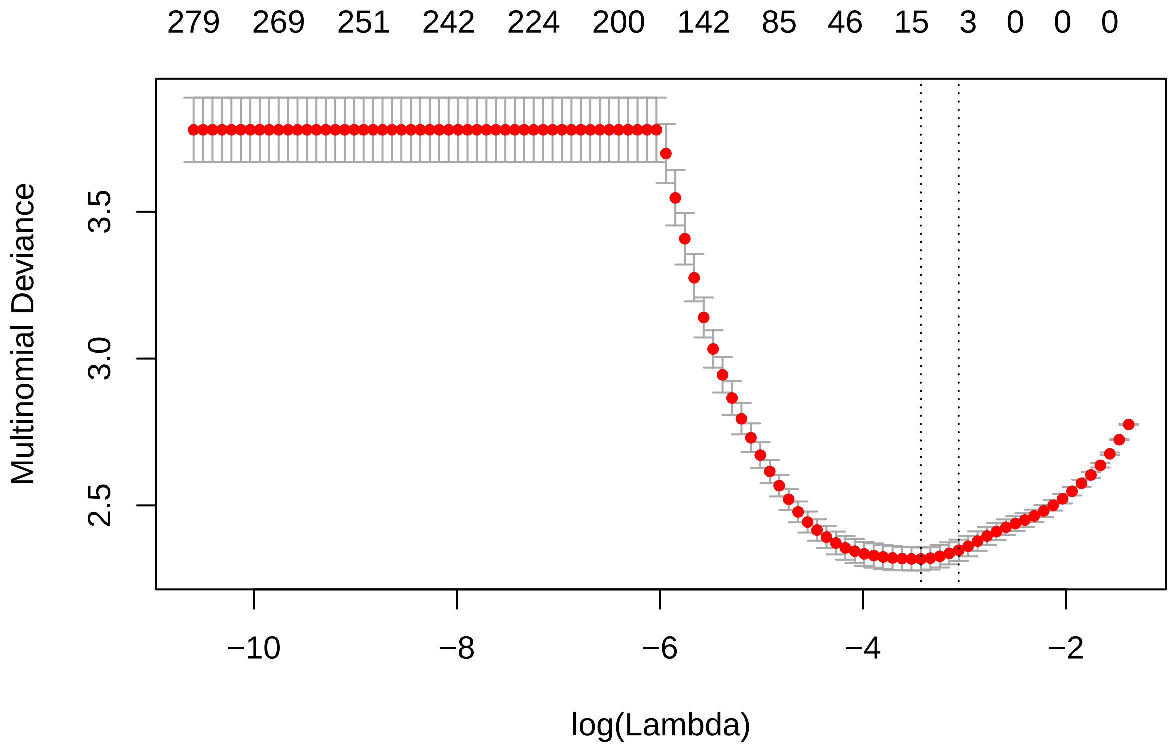
4.2 Cross-validation (CV) for the optimal α

```r
library(glmnet)
set.seed(123)

X <- as.matrix(d[, 1:(ncol(d) - 1)])
Y <- d$y_cat

# cross validation to search for the optimal
fit_cv = cv.glmnet(X, Y, alpha = 1, family = "multinomial")

plot(fit_cv)
```



4.3 Applying the optimal λ provided by CV

```
# should specify lambda = fit_cv$lambda.min
fit1 = glmnet(X, Y, alpha = 1, family="multinomial", lambda = 0.3)
zz = coef(fit_cv, 0.03)
```

Here we should specify `lambda = fit_cv$lambda.min` in `glmnet()` function. However, it does not really work for this specific data. So I just changed it to a random value of 0.3.

```
knitr::kable(cbind(zz[[1]][1:10,], zz[[2]][1:10,], zz[[3]][1:10,]),
             caption = "Penalized regression coefficients for different levels of Y")
```

Table 1: Penalized regression coefficients for different levels of Y

(Intercept)	2.338918	0.9670057	-2.133138
id	0.000000	0.0000000	0.000000
gender	0.000000	0.0000000	0.000000
DFA	0.000000	0.0000000	0.000000
RPDE	2.831181	0.0000000	0.000000
numPulses	0.000000	0.0000000	0.000000
numPeriodsPulses	0.000000	0.0000000	0.000000
meanPeriodPulses	0.000000	0.0000000	0.000000
stdDevPeriodPulses	214.061976	0.0000000	0.000000
locPctJitter	0.000000	0.0000000	0.000000

Since we have four different levels of y, there are three sets of regression coefficients. I only listed the first 10 coefficients here.