

LASSO multinomial logistic regression

Variable selection for high dimensional data

Miao Cai

9/22/2019

1 Introduction

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**: 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 LASSO syntax with glmnet package

```
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.
- **lambda**: a numeric value defining the amount of shrinkage. Should be specify by analyst.

3 Application using glmnet

3.1 Data importing

```
library(data.table)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.1
```

```
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)
```

```
## Warning in require_bit64(): Some columns are type 'integer64' but
## package bit64 is not installed. Those columns will print as strange
## looking floating point data. There is no need to reload the data. Simply
## install.packages('bit64') to obtain the integer64 print method and print
## the data again.
```

3.2 Cross-validation (CV) for the optimal α

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.6.1
```

```
set.seed(123)
```

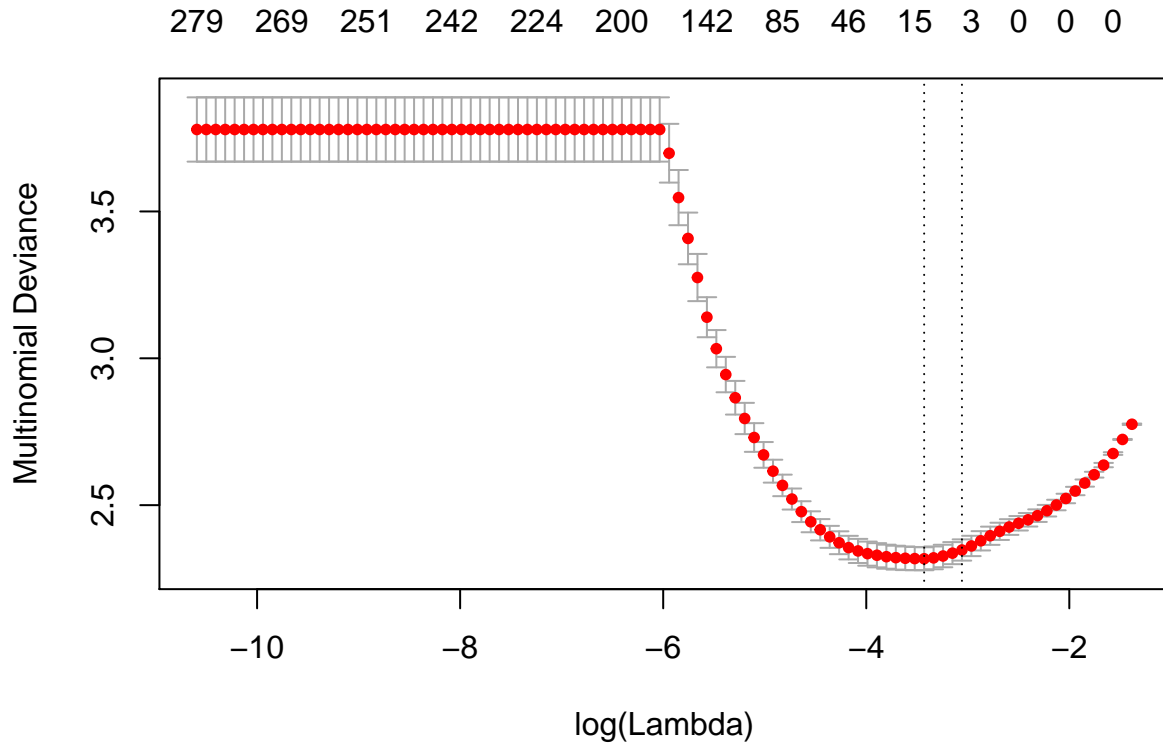
```
X <- as.matrix(d[, 1:(ncol(d) - 1)])
```

```
Y <- d$y_cat
```

```
# cross validation to search for the optimal
```

```
fit_cv = cv.glmnet(X, Y, alpha = 1, family = "multinomial")
```

```
plot(fit_cv)
```



3.3 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.