17 - Logistic Regression and Classification

Junvie Pailden

SIUE, F2017, Stat 589

October 30, 2017

Binary Response (Target) Variable

Analysis of whether or not business firms have an industrial relations department, according to size of firm.

- Response: firm has industrial relations $(Y = 1)$ or firm does not have industrial relations $(Y = 0)$
- Explanatory: size of firm

Study of labor force participation of married women, as a function of age, number of children, and husbands income.

- Response: married woman in the labor force $(Y = 1)$ and married woman not in the labor force $(Y = 0)$
- Explanatory: age, $\#$ of children, husbands income

Simple Logistic Regression Model

- Response variable Y_i ,$i = 1, \ldots, n$, are independent Bernoulli r.v. with expected value $E(Y_i) = p_i$ and Z_i predictor,

$$E(Y_i|Z_i = z_i) = p(z_i) = \frac{\exp(\beta_0 + \beta_1 z_i)}{1 + \exp(\beta_0 + \beta_1 z_i)}$$

  is called the *logistic response function*. Equivalently,

$$\ln\left(\frac{p(z_i)}{1 - p(z_i)}\right) = \beta_0 + \beta_1 z_i$$

- The ratio $p(z_i)/(1 - p(z_i))$ is called the odds and $\log\left(\frac{p(z_i)}{1-p(z_i)}\right)$ is called the logit.

  It is assumed that the logit transformation of the response prob'y of succes has a linear relationship with the predictor variable(s).

Recall Credit Scoring on German Bank

The German credit data set was obtained from the UCI Machine Learning Repository. The data set, which contains attributes and outcomes on 1000 loan applications.

This dataset classifies people described by a set of attributes/predictors as good or bad credit risks.

- Number of rows = 1000
- Number of attributes = 20

Load the German Credit Data

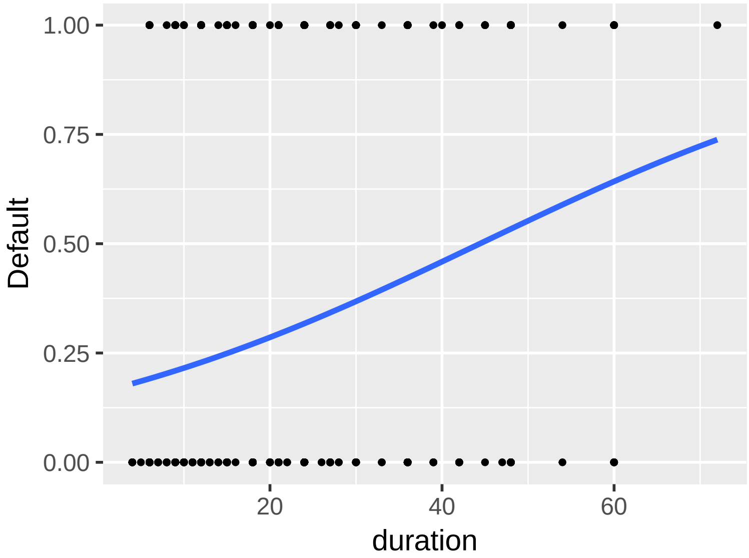
```
credit <- read.csv("germancredit.csv")
library(dplyr)
names(credit)
```

```
#  [1] "Default"          "checkingstatus1" "duration"
#  [4] "history"          "purpose"         "amount"
#  [7] "savings"          "employ"          "installment"
# [10] "status"           "others"          "residence"
# [13] "property"         "age"             "otherplans"
# [16] "housing"          "cards"           "job"
# [19] "liable"           "tele"            "foreign"
```

Fit a Simple Logistic Regression Model

- Response Variable: $y =$ Default,
- Predictor Variable: $z =$ Duration of Loan

```r
library(ggplot2)
credit %>%
  ggplot(aes(x = duration, y = Default)) +
  geom_point() +
  stat_smooth(method = "glm",
        # logistic reg falls under general linear model
        method.args = list(family = "binomial"),
        # use binomial family under glm
        se = F) # remove confidence bands
```


1.00
0.75
0.50
0.25
0.00
Default
20
40
60
duration

Using glm function to fit logistic regression

```
fit.logr1 <- glm(Default ~ duration,
                 data = credit,
                 family = binomial)
# use binomial family for logistic reg
fit.logr1
```

```
#
# Call:  glm(formula = Default ~ duration, family = binomi
#
# Coefficients:
# (Intercept)     duration
#     -1.6664       0.0375
#
# Degrees of Freedom: 999 Total (i.e. Null);  998 Residual
# Null Deviance:        1220
# Residual Deviance: 1180   AIC: 1180
```

Logistic Regression Likelihood Function

Since each $Y_i = 0, 1$,$i = 1, \ldots, n$, is a Bernoulli r.v., its prob'y dist'n

$$f_i(Y_i) = p_i^{Y_i}(1 - p_i)^{1-Y_i}$$

The joint prob'y dist'n of $Y_1, \ldots, Y_n$ independent r.v.s. is

$$g(Y_1, \ldots, Y_n) = \prod_{i=1}^{n} f_i(Y_i) = \prod_{i=1}^{n} p_i^{Y_i}(1 - p_i)^{1-Y_i}$$

Log-likelihood Function on the Coefficients

For convenience,

$$\ln\, g(Y_1,\ldots,Y_n) = \sum_{i=1}^{n}\left[Y_i \cdot \ln\left(\frac{p_i}{1-p_i}\right)\right] + \sum_{i=1}^{n} \ln\left(1-p_i\right) \quad \text{(Why?)}$$

Thus, the log-likelihood function of (β_0, β_1) can be written as

$$\begin{aligned}\ln\, L(\beta_0,\beta_1) &= \ln\, g(Y_1,\ldots,Y_n)\\ &= \sum_{i=1}^{n} Y_i(\beta_0+\beta_1 X_i) - \sum_{i=1}^{n}\ln\left[1+\exp(\beta_0+\beta_1 X_i)\right]\end{aligned}$$

Maximum Likelihood Estimation

- No closed-form solution exists for the MLEs of β_0 and β_1.
- Computer-intensive numerical search procedures are required to find the MLE estimates b_0 and b_1. Use built-in `glm` function in R to find these estimates.
- Once the MLE estimates b_0 and b_1 are found, we can compute the fitted logistic response function

$$\hat{p}(z_i) = \frac{\exp(b_0 + b_1 z_i)}{1 + \exp(b_0 + b_1 z_i)}$$

- We can also compute the fitted logit response function (log-odds)

$$\text{logit}(\hat{p}(z_i)) = \ln\left(\frac{\hat{p}(z_i)}{1 - \hat{p}(z_i)}\right) = b_0 + b_1 z_i$$

Interpretation of b_1

At $Z = z_j$ and $Z = z_j + 1$,

$$\begin{aligned}\text{logit}[\hat{p}(z_j)] &= b_0 + b_1 z_j \\ \text{logit}[\hat{p}(z_j + 1)] &= b_0 + b_1(z_j + 1)\end{aligned}$$

The difference between the two fitted values,

$$\begin{aligned}\text{logit}[\hat{p}(z_j + 1)] - \text{logit}[\hat{p}(z_j)] &= \log(odds_2) - \log(odds_1) \\ &= \log(\text{odds ratio}) = b_1 \\ \text{odds ratio} &= \hat{OR} = \exp(b_1)\end{aligned}$$

The estimated odds is multiplied by $\exp(b_1)$ for any unit increase in z.

Model Summary

```r
model1 <- summary(fit.logr1)
names(model1)
```

```
#  [1] "call"           "terms"          "family"         '
#  [5] "aic"            "contrasts"      "df.residual"    '
#  [9] "df.null"        "iter"           "deviance.resid" '
# [13] "aliased"        "dispersion"     "df"             '
# [17] "cov.scaled"
```

```r
# coefficients
model1$coefficients
```

```
#             Estimate Std. Error z value Pr(>|z|)
# (Intercept)  -1.6664     0.1466  -11.37 6.21e-30
# duration      0.0375     0.0057    6.58 4.63e-11
```

Multiple Logistic Regression

The multiple logistic response function with r predictors is

$$E\{Y|\mathbf{z}_i\} = p(\mathbf{z}_i) = \frac{\exp(\mathbf{z}_i'\beta)}{1+\exp(\mathbf{z}_i'\beta)} = \frac{1}{1+\exp(-\mathbf{z}_i'\beta)}$$

where

$$\mathbf{z}_i'\beta = \beta_0 + \beta_1 z_{i,1} + \ldots + \beta_r z_{i,r}.$$

Note that the r predictors can be either continuous or discrete.

Using `glm` function to fit multiple logistic regr (full data)

```r
fit.logr2 <- glm(Default ~ duration + amount + installment
                 data = credit,
                 family = binomial)

summary(fit.logr2)$coefficients
```

```
#              Estimate Std. Error z value Pr(>|z|)
# (Intercept) -1.54e+00   0.334509   -4.59 4.42e-06
# duration     2.67e-02   0.007698    3.47 5.29e-04
# amount       6.83e-05   0.000034    2.01 4.47e-02
# installment  2.00e-01   0.072288    2.76 5.75e-03
# age         -2.08e-02   0.006771   -3.08 2.08e-03
```

Using Logistic Regression for Classification (Performance)

Randomly split data into 70% training and 30% testing.

```r
set.seed(1) # fix seed to get same training set
train <- sample(nrow(credit), size = 0.7*nrow(credit))
cred.train <- credit[train, ]
dim(cred.train)
```

```
# [1] 700  21
```

```r
cred.test <- credit[-train,]
dim(cred.test)
```

```
# [1] 300  21
```

Multiple Logistic Regression on Train/Test Credit Data

```r
# build model on training data
fit.train <- glm(Default ~ duration + amount +
                            installment +  age,
                 data = cred.train,
                 family = binomial)
problr.test <- predict(fit.train,
                  newdata = cred.test,
                  type = "response") #predicted probabilit
table(cred.test$Default) # ratio of 1's and 0's
```

```
#
#   0   1
# 205  95
```

Determining Optimal Cutoff for Classification

The default cutoff prediction probability score is 0.5 or the ratio of 1's and 0's in the training data ($95/205 = 0.463$).

But sometimes, tuning the probability cutoff can improve the accuracy in both the development and validation samples.

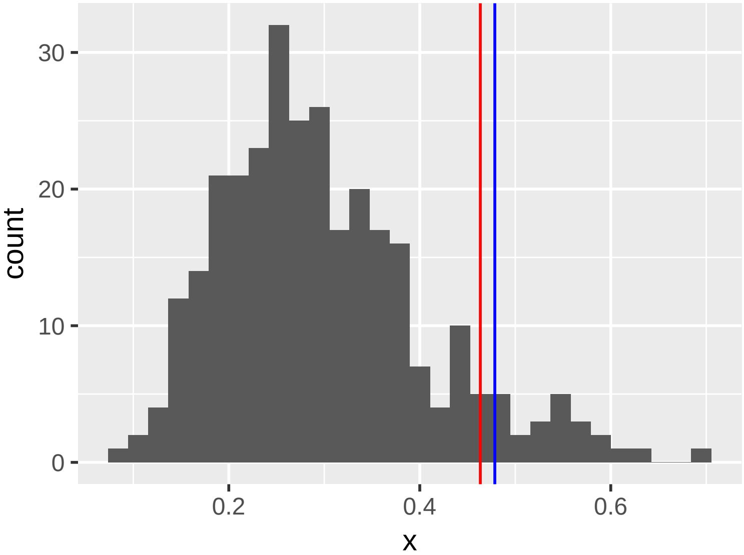
The `InformationValue::optimalCutoff` function provides ways to find the optimal cutoff to improve the prediction of 1's, 0's, both 1's and 0's that reduces the misclassification error.

```
library(InformationValue)
(optCutOff <- optimalCutoff(cred.test$Default,
                            predictedScores = problr.test))
```

```
# [1] 0.479
```

Histogram of Predicted Probabilities

```r
data.frame(x = problr.test) %>%
  ggplot(aes(x)) +
  geom_histogram() +
  geom_vline(xintercept = c(95/205, optCutOff),
             colour = c("red", "blue"))
```


30
20
10
0
count
0.2
0.4
0.6
x

Model Performance using Default Cut-off

```
confusionMatrix(cred.test$Default,
                predictedScores = problr.test,
                threshold = 95/205)
```

```
#     0  1
# 0 194 80
# 1  11 15
```

```
misClassError(cred.test$Default,
              predictedScores = problr.test,
                threshold = 95/205)
```

```
# [1] 0.303
```

Model Performance using Optimal Cut-off

```
confusionMatrix(cred.test$Default,
                predictedScores = problr.test,
                threshold = optCutOff)
```

```
#     0  1
# 0 198 80
# 1   7 15
```

```
misClassError(cred.test$Default,
                predictedScores = problr.test,
                threshold = optCutOff)
```

```
# [1] 0.29
```

Note that the misclassification rate under the logistic regression model is slightly better than either LDA or QDA (see lecture 15).

Receiver Operating Characteristics (ROC) Curve

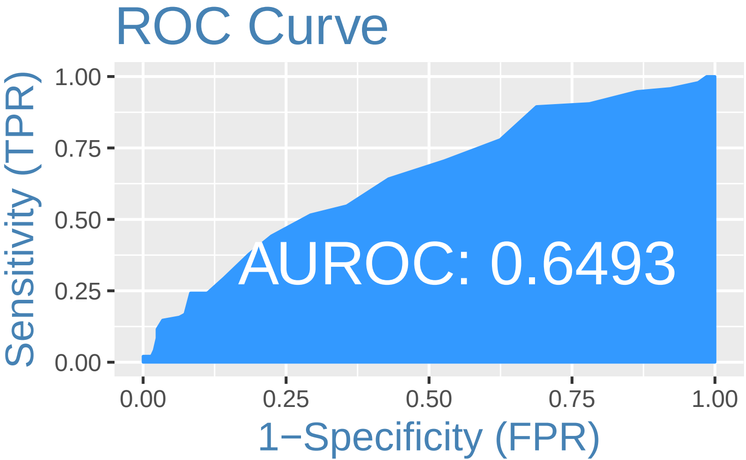
Receiver Operating Characteristics Curve traces the percentage of true positives accurately predicted by a given logit model as the prediction probability cutoff is lowered from 1 to 0.

For a good model, as the cutoff is lowered, it should mark more of actual 1's as positives and lesser of actual 0's as 1's. So for a good model, the curve should rise steeply, indicating that the true positive rate TPR (Y-Axis) increases faster than the false positive rate FPR (X-Axis) as the cutoff score decreases.

Larger area under the ROC curve (depends on application) implies better predictive ability of the model.

ROC Curve of Logistic Reg Model for Credit Data

```
plotROC(cred.test$Default, predictedScores = problr.test)
```


ROC Curve
Sensitivity (TPR)
AUROC: 0.6493
1−Specificity (FPR)