Logistic regression

Dr Juan H Klopper

Introduction

Logistic regression is similar to linear regression, but is used when the outcome dependent variable, YY, is categorical. Recall the equation for linear regression, written below for a single variable, XX, with an intercept given as β_0β0 and an error term, ϵ_iϵi:

$$Y_i = \beta_0 + b_1 X_i + \epsilon_i$$
Yi=β0+b1Xi+ϵi

The best fit for the parameters β_0β0 and β_iβi were found by minimizing the square of the errors (difference between y_i yi and $\hat{y}_i$ y^i).

Since YY is now a categorical variable, the effort shift to calculating the probability of finding one of the data point values in the sample space of YY, i.e. $P(Y)$P(Y). It is denoted by:

$$P(Y) = \frac{1}{1 + e^{-(b_0+\beta_1 X)}}$$
P(Y)=11+e−(b0+β1X)

The extends naturally to more than a single independent variable.

This post will explain logistic regression in the case of a dichotomous dependent variable, coded as $0$0 when an outcome does not occur and $1$1 if it does. Finding the optimum parameter values is done by the process of *maximum-likelihood estimation*.

Log-likelihood statistic

The result of the probability, $P(Y)$P(Y) above, is between $0$0 and $1$1 (inclusive). The difference between this probability value and the actual value (being either $0$0 or $1$1) is expressed in terms of the log-likelihood:

$$\sum_{i=1}^{n} [Y_i \ln(P(Y_i)) + (1 - Y_i) \ln(1 - P(Y_i))]$$

∑i=1n[Yiln(P(Yi))+(1−Yi)ln(1−P(Yi))]

This fulfills the same role as the sum of square errors in linear regression.

Deviance

Another way of assessing how well the parameters fit the data is *deviance*. It is calculated as:

$$\text{deviance} = -2 \times \log - \text{likelihood}$$

deviance=−2×log−likelihood

It is more useful as it has a χ^2χ2 distribution allowing for hypothesis testing.

It also allows for the comparison of one logistic regression model against another. It is common to use a baseline model. One such model uses only the dependent variable dichotomous data point value that occurs most commonly. In terms of logistic regression, this is a model that only contains the intercept. The deviance of the baseline minus the deviance of the *new* model is then the χ^2χ2 statistic. Degrees of freedom is calculated by subtracting $1$1 from the number of parameters in the new model (which is $1$1 plus the number of independent variables).

Wald statistic

In linear regression the probability of finding a coefficient for any of the independent variables can be calculated. The concurrent statistic in logistic regression is the *z* statistic and it follows a normal distribution. It calculates the probability of finding a coefficient value other than $0$0. It is calculated as follows:

$$z = \frac{b}{SE_b}$$

z=bSEb

Here bb is the coefficient under consideration and SE_bSEb is its standard error. The *p* value calculated from this must be viewed with caution. As the value for bb increases, its standard error increases even more, leading to a small *z* statistic and underestimating the significance of the particular independent variable.

R statistic

The *R* statistic is the partial correlation between the dependent variable and each of the predictor variables. It ranges form -1 −1 to $+1$ +1 (inclusive), with the latter indicating a rise in the likelihood of the dependent variable as the independent variable under consideration increases, etc.. It is calculated as follows:

$$R = \sqrt{\frac{z^2 - 2df}{defiance_{baseline}}}$$

R=z2−2dfdefiancebaseline

With the Wald statistic as part of the calculation of the *R* statistic, it must be viewed with caution. Even more so for the R^2 value. There are various ways of calculating the latter. The Hosmer-Lemeshow R^2 value is calculated as:

$$R^2 = \frac{deviance_{model}}{deviance_{new}}$$

R2=deviancemodeldeviancenew

The Cox-Snell R^2 value is calculated as:

$$R^2 = 1 - e^{\frac{deviance_{new} - deviance_{baseline}}{n}}$$

R2=1−edeviancenew−deviancebaselinen

Information criteria

Yet more ways of assessing how well the model performs is the Akaike information criterion (AIC) and the Bayes information criteria (BIC). Both aim to solve the problem of an increase in R^2 value whenever more independent variables are added. They penalize the assessment for adding more variables.

$$AIC = deviance - 2k$$
$$BIC = deviance + 2k \times \ln(n)$$

AIC=deviance−2kBIC=deviance+2k×ln(n)

Here nn is the sample size and kk is the number of independent variables.

Odds ratio

This is an important measure of a logistic regression model. Each of the independent variables in the model increases or decreases the odds of the outcome (11) of the dependent variable. This change has a threshold of 11 and refers to a single unit of change in the independent variable and its effect. Below this, the independent variable under consideration decreases the odds of the outcome. Above 11, it increases the odds.

The odds ratio (OR) is calculated from the coefficients.

$$OR = e^{b_i}$$

OR=ebi

This OR can be expressed as a percentage. In the case of the independent variable being categorical, we have the following. If the OR is less than 1, subtract it from 1. For example, with an OR = $0.4$0.4 it follows that $1.0 - 0.4 = 0.6$ 1.0−0.4=0.6. Multiplying this by 100 gives rise to the statement that a unit rise in the particular independent variable reduces the odds of the outcome by 60%.

If the value is larger than 1, subtract 1 from it. For example, given and OR of $1.12$1.12 it follows that $1.12 - 1.0 = 0.12$ 1.12−1.0=0.12. Multiplying this by 100 gives rise to the statement that a unit increase in the particular independent variable reduces increases the odds of the outcome by 12%.

Building the model

There are various ways of building a model. In the first, all the independent variables are entered. This is termed the *forced entry method*.

The *stepwise method* on the other hand, has a forward and backward method. In the forward method only the constant term is used. A single variable is then added to the model at a time. It is kept only if it improves the AIC or the BIC. In the backward method, the model is initiated as with the forced entry method, but independent variables are removed one at a time. If it increases the AIC or BIC it is brought back.

There are also hybrids of the two stepwise methods. In the forward-backward approach the model starts with the forward method, but each time a variable is added all the entered variables are tested for possible removal.

Assumptions

The first assumption is that there is a linear relationship between continuous independent variables and the logit of the outcome variable.

Independence of errors requires that there be no dependence between samples in the model, i.e. using the same individuals at different times.

Multicollinearity refers to a relationship (correlation) between the independent variables and must be excluded from logistic regression models.

Building the model

The datasheet for this post is a spreadsheet file in *comma seperated values* format with the first rows containing the variable names (column headers). The code snippet below imports the data file as a `list` in the computer variable `df`.

```
df = read.csv("LR.csv", header = TRUE)
```

A look at the variables is achieved by the `names()` command.

```
names(df)
```

```
## [1] "Gender"  "Hours"   "Prelim"  "Outcome"
```

In this case, *Ouctome* is the dependent variable. The other three can be taken as independent variables. The first six rows can be viewed using the `head()` command.

```
head(df)
```

```
##   Gender Hours Prelim  Outcome
## 1   Male  1305     11    Alive
## 2   Male  1537     12 Deceased
## 3   Male  1670     13    Alive
## 4   Male  1754     11    Alive
## 5   Male  1834     10 Deceased
## 6   Male  1848     15 Deceased
```

Note that gender is a categorical variable with a sample space containing two values, `Male` and `Female`. A dichotomous sample space was chosen for the sake of explanation. The dependent variable *Outcome* is also categorical. `R` will automatically change these to numerical values, i.e. `0` and `1`. This will be done alphabetically, though, and needs attention to set up properly. This is achieved using the `relevel()` command. In the models required for this post, it would make sense to have `Deceased` as the base data point value and `Alive` as `1`. For the sake of argument `Male` is set as the base data point value. This model will the test the independent variable's' effect on an `Alive` outcome.

```
df$Gender <- relevel(df$Gender, "Male")
df$Outcome <- relevel(df$Outcome, "Deceased")
```

A model can be created with just the intercept as predictor variable. The `glm()` command creates the model. The code snippet below shows `Outcome` as dependent variable. The `~` symbol acts as equal sign. the `family =` argument is set to `binomial()` indicating that the outcome is dichotomous.

```
model_0 <- glm(Outcome ~ 1, data = df, family = binomial())
```

A summary can be printed using the `summary()` command.

```
summary(model_0)
```

```
##
## Call:
## glm(formula = Outcome ~ 1, family = binomial(), data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6006  -1.6006   0.8067   0.8067   0.8067
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.9555     0.1664   5.742 9.36e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 212.7  on 179  degrees of freedom
## Residual deviance: 212.7  on 179  degrees of freedom
## AIC: 214.7
##
## Number of Fisher Scoring iterations: 4
```

Of note is the *null-deviance* that will act as a baseline. Adding and removing independent variables must lower the *residual-deviance* (which is equal to the null-deviance for now).

In keeping with the backward stepwise method, all the independent variables are added to `model_1` below.

```
model_1 <- glm(Outcome ~ Gender + Hours + Prelim, data = df, family = binomial())
```

The summary is as follows:

```
summary(model_1)
```

```
##
## Call:
## glm(formula = Outcome ~ Gender + Hours + Prelim, family = binomial(),
##     data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1937  -0.9966   0.4974   0.7954   1.4223
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.9670448  1.6369114   1.813  0.06990 .
## GenderFemale -1.2758823  0.5232272  -2.438  0.01475 *
## Hours         0.0006937  0.0005203   1.333  0.18243
## Prelim       -0.2536125  0.0939686  -2.699  0.00696 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 212.70  on 179  degrees of freedom
## Residual deviance: 184.93  on 176  degrees of freedom
## AIC: 192.93
##
## Number of Fisher Scoring iterations: 4
```

The residual deviance has indeed decreased to 184.93, which shows an improvement of the model.

Note that `Gender` shows `Male` as baseline, as was encoded above. The model then shows the effect of being female.

Still in keeping with the backward stepwise method, in `model_2` below, `Gender` is removed. The `.` can be used in place of *ALL* the independent variable listed by addition in `model_1`.

```
model_2 <- glm(Outcome ~ . - Gender, family = binomial(), data = df)
summary(model_2)
```

```
##
## Call:
## glm(formula = Outcome ~ . - Gender, family = binomial(), data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1785  -1.1237   0.6083   0.8245   1.3822
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.376e+00  1.327e+00   4.051 5.10e-05 ***
## Hours       -1.632e-05  4.159e-04  -0.039    0.969
## Prelim      -3.406e-01  8.709e-02  -3.911 9.19e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 212.70  on 179  degrees of freedom
## Residual deviance: 191.38  on 177  degrees of freedom
## AIC: 197.38
##
## Number of Fisher Scoring iterations: 4
```

The residual deviance has increased to 191.38. `Gender` must be returned to the model. In `model_3`, `Hours` are removed.

```
model_3 <- glm(Outcome ~ . - Hours, data = df, family = binomial())
```

The residual deviance can be expressed on its own.

```
model_3$deviance
```

```
## [1] 186.7366
```

This is still higher than the model with all the independent variables. In `model_4` `Prelim` is removed.

```
model_4 <- glm(Outcome ~ . - Prelim, data = df, family = binomial())
model_4$deviance
```

```
## [1] 192.8
```

Higher again. The `model_1` is chosen as the best model. It can be evaluated as per the discussion above.

The function below extracts the relevant data from the model that is passed as argument and expresses the re;event R^2 values.

```
rsquared <- function(created_model) {
  dev <- created_model$deviance
  null_dev <- created_model$null.deviance
  model_n <- length(created_model$fitted.values)
  R_l <- 1 - dev / null_dev
  R_cs <- 1 - exp(-(null_dev - dev) / model_n)
  R_n <- R_cs / (1 - exp(-(null_dev / model_n)))
  cat("Pseudo R-squared for logistic regression model\n\n")
  cat("Hosmer and Lemeshow R-squared\t", round(R_l, 3), "\n")
  cat("Cox and Snell R-squared\t\t\t", round(R_cs, 3), "\n")
  cat("Nagelkerke R-squared\t\t\t\t", round(R_n, 3), "\n")
}
```

```
rsquared(model_1)
```

```
## Pseudo R-squared for logistic regression model
##
## Hosmer and Lemeshow R-squared     0.131
## Cox and Snell R-squared           0.143
## Nagelkerke R-squared              0.206
```