Logistic Regression For Binary Classification

mcc

3/13/2020

Logistic Regression For Binary Classification

Introduction

For individuals who have studied cell biology or biochemistry, logistic regression may be familiar as dose-response curves, enzyme kinetic curves, sigmoidal curves, median lethal dose curve (LD-50) or even an exponential growth curve given limited resources.

However, in the context of predictive modeling, Logistic Regression is used as a binary classifier that toggle between the logical values of zero or one.

Logistic regression (Logit) derives its name from its similarity to linear regression, as we shall see below. The input/independent variable for Logit is the set of real numbers, ($X \in \Re$). While, the output of a Logistic Regression is not represented by $\{0, 1\}$, ($Y \notin \Re$),

$$f(x) = \begin{cases} 0 & for \ \ x < 0 \\ 1 & for \ \ x \geq 0 \end{cases} \tag{1}$$

Using Logistic Regression, we may calculate the presence or absence of a product or quality that we wish to model given a difficult situation where the transition is not clear.

In the figure below, the function's domain, $X \in \{-\infty \text{ to } \infty\}$, whereby its range is $\{0, 1\}$. In the figure, the *decision boundary* is $x = 0$, denoted by the *red dotted line*. At the inflection point the curves range changes from *zero*, absence, to *one*, the presence of quality or item.

Logistic Curve

if x < 0 then y = 0   if x >= 0 then y = 1
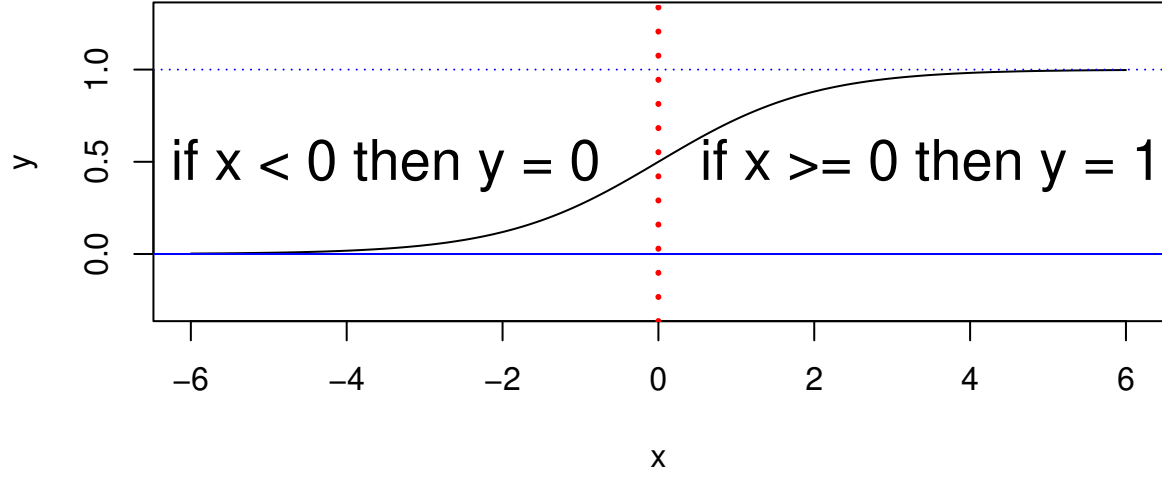
The logistic growth curve is commonly denoted by:

$$f(x) \; = \; \frac{M}{1 + Ae^{-r(x - x_0)}} \tag{2}$$

where M is the curve's maximum value, r is the maximum growth rate (also called the Malthusian parameter [1]), x_0 is the midpoint of the curve, A is the number of times that the initial population must double to reach M.[2]

In the specific case of *Logistic Regression for Binary Classification* where we have a probability between 0 and 1, M, and A take on the value one.

$$f(x) \; = \; \frac{1}{1 + e^{-(WX+b)}} \tag{3}$$

Since the logistic equation is exponential, it is easier to work with the formula in terms of its odds or *log-odds*. Odds are the probabilities of success over failure denoted as $\frac{p}{1-p}$ and more importantly, in this situation, log-odds are $ln\left(\frac{p}{1-p}\right)$.

Simply by using log-odds, logistic regression may be more easily expressed as a set of linear equations in x.[3] Hence we can now go from linear regression to logistic regression.

Step #1:

$$ln\;\left(\frac{Pr(y_i \; = \; 1|x_i)}{Pr(y_i \; = \; 0|x_i)}\right) = \; \beta_0 + \beta_1 x_1 + \; ... \; + \beta_n x_n \tag{4}$$

Step #2: Substitute (p for $Pr(y_i \; = \; 1|x_i)$) and ($1-p$ for $Pr(y_i \; = \; 0|x_i)$) and change notation to summation on the right hand side;

---

[1] https://en.wikipedia.org/wiki/Malthusian__growth_model
[2] https://en.wikipedia.org/wiki/Logistic_function
[3] http://juangabrielgomila.com/en/logistic-regression-derivation/

$$ln \left(\frac{p}{1-p} \right) = \sum_{i}^{k} \beta_i x_i \tag{5}$$

Step #3: Eliminate the natural log by taking the exponent on both sides;

$$\frac{p}{1-p} = exp \left(\sum_{i}^{k} \beta_i x_i \right) \tag{6}$$

Step #4: Substitute $u = \sum_{i}^{k} \beta_i x_i$;

$$\frac{p}{1-p} = e^u \tag{7}$$

Step #5: Rearrange to solve for p;

$$p(u) = \frac{e^u}{1+e^u} \tag{8}$$

Step #6: Incidentally, to find the probabilities, take the derivative of both sides using quotient rule;

$$p'(u) = \frac{(e^u)(1+e^u) - (e^u)(e^u)}{(1+e^u)^2} \tag{9}$$

Step #7: Simplify;

$$p'(u) = \frac{e^u}{(1+e^u)^2} \tag{10}$$

Step #8: Separate out to produce two fractions;

$$p'(u) = \left(\frac{e^u}{1+e^u} \right) \cdot \left(\frac{1}{1+e^u} \right) \tag{11}$$

Step #9: Substitute our previous success and failure variables back into place;

$$p'(u) = p(u) \cdot (1 - p(u)) \tag{12}$$

Now we can calculate the probabilities as well as the values for any given x value.

Logit-20 Training Using 20 Features

```
# Load Libraries
Libraries <- c("doMC", "knitr", "readr", "tidyverse", "caret")
for (p in Libraries) {
    library(p, character.only = TRUE)
}
```

```
# Import relevant data
c_m_TRANSFORMED <- read_csv("./00-data/02-aac_dpc_values/c_m_TRANSFORMED.csv",
                            col_types = cols(Class = col_factor(levels = c("0", "1")),
                                            PID = col_skip(),
                                            TotalAA = col_skip())))
```

```
# Partition data into training and testing sets
set.seed(1000)
index <- createDataPartition(c_m_TRANSFORMED$Class, p = 0.8, list = FALSE)
training_set.1 <- c_m_TRANSFORMED[index, ]
```

- The `test.set.1` and `Class.test` data sets are not produced since the Logit run with 20 features was not deemed useful. The reason for its dismissal was that is contained extraneous features.

- The first training run is to determine if all 20 features (amino acids) are necessary for our logistic regression model.

```
set.seed(1000)
registerDoMC(cores = 3)        # Start multi-processor mode
start_time <- Sys.time()       # Start timer

# Create model, 10X fold CV repeated 5X
tcontrol <- trainControl(method = "repeatedcv",
                         number = 10,
                         repeats = 5)

model_obj.1 <- train(Class ~ .,
                     data = training_set.1,
                     trControl = tcontrol,
                     method = "glm",
                     family = "binomial")

end_time <- Sys.time()         # End timer
end_time - start_time          # Display time
```

```
## Time difference of 3.231281 secs
```

```
registerDoSEQ()                   # Stop multi-processor mode
```

Logit-20 Summary #1

```
summary(model_obj.1)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -5.9372   -0.2835   -0.0194    0.0516    3.6884
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    8.0525     9.2156   0.874 0.382234
## A              5.0438     9.6899   0.521 0.602699
## C            -14.2228     2.6949  -5.278 1.31e-07 ***
## D            -36.2676     8.0845  -4.486 7.25e-06 ***
## E             27.6016    11.1292   2.480 0.013135 *
## F              5.6174     5.2654   1.067 0.286034
## G            -22.1970    10.3043  -2.154 0.031229 *
## H             90.1101    12.1105   7.441 1.00e-13 ***
## I             -5.9795     4.3945  -1.361 0.173610
## K             -2.8961     9.8468  -0.294 0.768669
## L             -3.7417     9.2217  -0.406 0.684926
## M             -0.1427    12.0747  -0.012 0.990570
## N              3.3478     9.6749   0.346 0.729319
## P            -39.7466    11.1010  -3.580 0.000343 ***
## Q             -5.6804    11.2516  -0.505 0.613664
## R            -83.6045    11.8104  -7.079 1.45e-12 ***
## S             -9.9745    10.0872  -0.989 0.322750
## T            -36.5980     9.2791  -3.944 8.01e-05 ***
## V             16.3411     9.7859   1.670 0.094946 .
## W              9.0169    13.8870   0.649 0.516141
## Y            -31.9282    11.1167  -2.872 0.004078 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2593.68  on 1872  degrees of freedom
## Residual deviance:  657.72  on 1852  degrees of freedom
## AIC: 699.72
##
## Number of Fisher Scoring iterations: 8
```

The Akaike information criterion (AIC)[4] for model #1 is 699.72. This will be used later to compare the models generated to rate their ability to utilize the features best. - The list of probabilities for the estimates leaves us with only **9 important features** to try re-modeling, R, H, P, C, E, Y, T, D, G.

Logit-9 Training Using 9 Features

- This test uses **ONLY** 9 features: (R, H, P, C, E, Y, T, D, G)

```
# Data import & handling
c_m_9aa <- read_csv("./00-data/02-aac_dpc_values/c_m_TRANSFORMED.csv",
               col_types = cols(Class = col_factor(levels = c("0", "1")),
                                  A = col_skip(), F = col_skip(),
                                  I = col_skip(), K = col_skip(),
                                  L = col_skip(), M = col_skip(),
                                  N = col_skip(), PID = col_skip(),
```

---

[4]https://en.wikipedia.org/wiki/Akaike_information_criterion

```r
                                     Q = col_skip(), V = col_skip(),
                                     S = col_skip(), TotalAA = col_skip(),
                                     W = col_skip())))
```

```r
# Partition data into training and testing sets
set.seed(1000)
index <- createDataPartition(c_m_9aa$Class, p = 0.8, list = FALSE)

training_set.2 <- c_m_9aa[ index, ]
test_set.2     <- c_m_9aa[-index, ]

Class_test.2 <- as.factor(test_set.2$Class)
```

```r
set.seed(1000)
registerDoMC(cores = 3)          # Start multi-core
start_time <- Sys.time()         # Start timer

# Create model, 10X fold CV repeated 5X
fitControl <- trainControl(method = "repeatedcv",
                           number = 10,
                           repeats = 5,
                           savePredictions = "final") # IMPORTANT: Saves predictions

model_obj.2 <- train(Class ~ .,
                     data = training_set.2,
                     trControl = fitControl,
                     method = "glm",
                     family = "binomial")

end_time <- Sys.time()           # End timer
end_time - start_time            # Display time
```

```
## Time difference of 2.613658 secs
```

```r
registerDoSEQ()                       # Stop multi-core
```

Logit-9 Summary

```
summary(model_obj.2)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min      1Q   Median       3Q      Max
## -6.2083  -0.2984  -0.0204   0.0601   3.5666
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    8.306      1.007    8.245  < 2e-16 ***
## C            -14.755      1.908   -7.733 1.05e-14 ***
## D            -31.411      4.949   -6.347 2.20e-10 ***
## E             21.932      5.092    4.307 1.66e-05 ***
## G            -23.259      5.071   -4.587 4.49e-06 ***
## H             94.580      8.431   11.218  < 2e-16 ***
## P            -29.394      6.264   -4.692 2.70e-06 ***
## R            -82.809      6.363  -13.015  < 2e-16 ***
## T            -40.915      5.624   -7.275 3.45e-13 ***
## Y            -37.860      6.291   -6.018 1.77e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2593.68  on 1872  degrees of freedom
## Residual deviance:  688.96  on 1863  degrees of freedom
## AIC: 708.96
##
## Number of Fisher Scoring iterations: 8
```

Logit-9 Confusion Matrix

```r
Predicted_test_vals <- predict(model_obj.2, test_set.2[, -1])

confusionMatrix(Predicted_test_vals, Class_test.2, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 235  18
##          1   8 206
##
##                Accuracy : 0.9443
##                  95% CI : (0.9195, 0.9633)
##     No Information Rate : 0.5203
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.8883
##
##  Mcnemar's Test P-Value : 0.07756
##
##             Sensitivity : 0.9196
##             Specificity : 0.9671
##          Pos Pred Value : 0.9626
##          Neg Pred Value : 0.9289
##              Prevalence : 0.4797
##          Detection Rate : 0.4411
##    Detection Prevalence : 0.4582
##       Balanced Accuracy : 0.9434
##
##        'Positive' Class : 1
##
```

- The Akaike information criterion (AIC) for model #2 is 708.96. This will be used later to compare the models generated to rate their ability to utilize the features best.
- The number of unique false-positives and false-negatives is 26.

Obtain List of False Positives & False Negatives From Logit-9

```r
fp_fn_logit <- model_obj.2 %>% pluck("pred") %>% dplyr::filter(obs != pred)

# Write CSV in R
write.table(fp_fn_logit,
            file = "./00-data/03-ml_results/fp_fn_logit.csv",
            row.names = FALSE,
            na = "",
            col.names = TRUE,
            sep = ",")

nrow(fp_fn_logit) ## NOTE: NOT UNIQUE NOR SORTED
```

```
## [1] 536
```

- The logistic regression second test produced 536 protein samples, which are either false-positives or false-negatives. The list of 536 proteins may have duplicates. Therefore they are NOT UNIQUE NOR SORTED.

Logit Conclusion

Logit is easy to implement and understand and can be used for feature selection.

Considering the table Logit Models, below, it is clear that model #2 with nine features best describes the better of the two models.

Akaike Information Criterion [5]

$$AIC \ = \ 2K \ - \ 2ln(\widehat{L}) \tag{13}$$

Where $ln(\widehat{L})$ is the log-likelihood estimate, K is the number of parameters.

Two Logit Models

Model #	Features	AIC
1	20	699.72
2	9	708.96

Logit is a common machine learning method. It is easy to understand and explain. This supervised binary classification method is very useful for determining the importance of the features which can be applied. As we saw in Model#1, there were 11 features that had probabilities of the estimates used above the 5% threshold cut-off. In Model#2, only nine features were used to describe the model, and the AIC increased by 9.24.

The nine features which best described the logistic regression model were R, H, P, C, E, Y, T, D, G. If we compare this to the Boruta test carried out in the EDA, we find the overlap interesting.

Comparison of Boruta Vs Logit: Order of Importance

Test	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Boruta	R	H	P	K	C	E	Y	T	S	A	V	U	I	F	D	G	N	L	M	Q
Logit-9	R	H	P	.	C	E	Y	T	.	.	.	.	.	.	D	G	.	.	.	.

The first 7 out of 8 amino acid features are seen in the proper order, as described by the Boruta Random Forest model. This is confirmation that Logit can pick up the importance of features similar to Boruta.

Logit produced 536 proteins, which are false-negatives or false-positives. It should be noted that the 536 are NOT UNIQUE NOR SORTED. The number of unique FN/FP from the confusion matrix is 26. These proteins will be investigated further in the Outliers chapter, which compares these FN/FP proteins to the PCA outliers.

The two tests for Logit (using 20 then 9 features) is interesting. This shows that Logit is an alternatvie way

---

[5]https://en.wikipedia.org/wiki/Akaike_information_criterion

of choosing the importance of features. As can be seen in the table "*Comparison of Boruta Vs Logit: Order of Importance*" it was seen that the first seven features lined up very closely with the Boruta Random Forest feature order of importance.