2.1 Logistic Regression

Contents

1 Logistic Regression Overview **1**

1.1 Log-Odds 1

1.2 Math 2

2 Logistic Regression in R **2**

2.1 Interpretation of Results 4

2.2 Logistic Regression Example 4

1 Logistic Regression Overview

As opposed to linear regression, logistic regression applies horizontal asymptotes to create boundaries of possible values - In our case it allows us to constrict the range between 0 and 1 - The goal is to find a function of the predictor (X) variables that relates them to a 0 or 1 outcome - Predicts whether something is true or false instead of continuous numerical data - Instead of fitting a line to data, we fit an S shaped logistic function from 0 to 1 - This means we are calculating a probability - We can still use continuous data, but we don't get a continuous answer - Calculated probabilities are used for classification - We are testing to see if the variable's effect on a prediction is significantly different from 0 - Variables that don't help the prediciton can be left out - As opposed to Least Squares used in linear regression, logistic regression uses **maximum likelihood**

Process: - Initialize the coefficient and intercept of all features (X) to zero - Multiply the value of each attribute (Y) by the coefficient to obtain log-odds - Plug the log-odds into the sigmoid function to obtain the probability in the range between 0 and 1

Standard Linear Regression - $Y = a + bX$ - Because it's linear, Y could be negative to positive infinity - We calculate R^2 to determine if variables are correlated

Logistic Regression

1.1 Log-Odds

- Used to determine logistic regression ### Odds Overview
  - Same as betting odds
    - E.g. having 1:4 odds of winning means out of 5 games, you expect 1 win and 4 losses totaling 5 games
    - This is not the same as probability, in this example your probability of winning would be $\frac{1}{5}$ or 0.2

Odds can be calculated from probability as follows: $odds = \frac{P(winning)}{1-P(winning)}$

or

$odds(Y=1) = \frac{P}{1-P}$ - P is probability of positive event - Therefore odds equal the probability of event occurring over the probability that it does not occur

1.1.1 Logit Function

- In our game example above, consider that your odds of losing will always be below 1
  - E.g. 1:4
- However, if your team is good it will be above 1 and sometimes by a large amount
  - E.g. 4:2 = 2, or 32:3 = 10.7
- So losing odds are 0 to 1, but winning odds are theoretically 1 to infinity
  - The magnitude of losing odds will look dramatically smaller than the magnitude of winning odds in a way that is difficult to interpret
- Taking the log of odds makes the odds of winning and losing symmetrical, thus log-odds

You get your odds, then you take the log of them. $log(odds)$

Remembering our calculation for odds, we can calculate log-odds from probability:

$log(\frac{P}{1-P})$

This log of the ratio of the probabilities is the **Logit Function** - The logit function is useful because it generates a normal distribution - This makes classification, especially binary classification, a lot easier - In a logistic regression, we are confined to probability values between 0 and 1 - In a linear regression we can theoretically have values from -infinity to infinity - How do we achieve this in a logistic regression? - Log-odds of the probability, or the logit function boiii - A probability of 0.5 becomes 0 on the logit y-axis

1.2 Math

- Instead of using Y like linear regression, we use $logit(Y)$
- The logit can be mapped to a probability, which in turn can be mapped to a class

Obtaining the classification value P_i

$ln\frac{p}{1-p} = W * X \rightarrow \frac{p}{1-p} = e^{W*X} \downarrow p_i = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots \beta_q x_q)}}$

Depending on our threshold value we can obtain the classification. For the default threshold of 0.5: $P_i > 0.5 \rightarrow 1$ $P_i \leq 0.5 \rightarrow 0$

Or Logit: $logit(P_i) > 1 \rightarrow 1$ $logit(P_i) \leq 1 \rightarrow 0$

2 Logistic Regression in R

Helpful explainer from UCLA Statistical Consulting.

`glm()` function: > "`glm` is used to fit generalized linear models, specified by giving a symbolic description of the linear predictor and a description of the error distribution."

"A typical predictor has the form response ~ terms where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for response."

- Basic Arguments:
  - `glm(formula, data, family)`
  - Formula: description of the model
  - Data: the data frame containing the variables in the formula
  - Family: The error distribution/link function to be used

Data set: Student breakfast, sleep, and leisure time vs performance

```r
class <- read.csv("Data Sets/2.1-ClassPer.csv")
head(class)
```

```
##   Obs Breakfast Sleep Laser.time Performance
## 1   1         0     8          2           1
## 2   2         1     7          1           1
## 3   3         0     9          0           1
## 4   4         1     6          4           1
## 5   5         1     8          2           1
## 6   6         0     7          3           1
```

```r
# Change Breakfast column to factor
class$Breakfast <- factor(class$Breakfast)

# Use glm() function
myLogit <- glm(Performance~Breakfast+Sleep+Laser.time, data = class, family = "binomial")

summary(myLogit)
```

```
##
## Call:
## glm(formula = Performance ~ Breakfast + Sleep + Laser.time, family = "binomial",
##     data = class)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4861  -0.7081  -0.0453   0.4956   2.0044
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -32.071     19.299  -1.662   0.0966 .
## Breakfast1     2.450      2.077   1.180   0.2382
## Sleep          3.598      2.217   1.623   0.1045
## Laser.time     2.566      1.464   1.753   0.0795 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 20.19  on 14  degrees of freedom
## Residual deviance: 11.34  on 11  degrees of freedom
```

```
## AIC: 19.34
##
## Number of Fisher Scoring iterations: 6
```

2.1 Interpretation of Results

Reading the coefficients gives us the function:

$logit(p) = -32.07 + 2.45Breakfast + 3.598Sleep + 2.566Leisure$ - The y-intercept works similarly, if the variables are zero then log(odds of performance) are -32.07 - z-value is the estimated intercept divided by the standard error - AKA number of standard deviations away from 0 - Small standard deviation or large p-value means this is not statistically significant - Coefficients - E.g. 2.45*Breakfast - For every unit of breakfast, the log(odds of performance) increases by 2.45 - If logit(p) > 1 then it is classified as 1, otherwise classified as 0

To get probability can convert logit(P) of 1.85 to P as follows:

$P = \frac{e^{1.85}}{1+e^{1.85}}$

P = 0.86 > 0.5 therefore classified

2.2 Logistic Regression Example

```r
att <- read.csv("Data Sets/2.0-Attrition.csv")
#str(att)
att$Attrition <- factor(att$Attrition)
```

2.2.1 Splitting Data with `createDataPartition()`

```r
#install.packages("caret")
library(caret)
createDataPartition(
  y,
  times = 1,
  p = 0.5,
  list = TRUE,
  groups = min(5, length(y))
)
```

2.2.1.1 Arguments

- **y:** a vector of outcomes.
- **times:** the number of partitions to create
- **p:** the percentage of data that goes to training
- **list:** logical - should the results be in a list (TRUE) or a matrix with the number of rows equal to floor(p * length(y)) and times columns.
- **groups:** for numeric y, the number of breaks in the quantiles

2.2.2 Partitioning the Data

```r
set.seed(123)

library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.2
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.0.2
```

```r
partition <- createDataPartition(y = att$Attrition, p = 0.7, list = FALSE)
train <- att[partition,]
test <- att[-partition,]
#str(train)
```

2.2.3 Running Regression Models with `glm()`

- Logistic regression and linear models are both "Generalized Linear Models," which is why we can use this function

  "…used to fit generalized linear models, specified by giving a symbolic description of the linear predictor and a description of the error distribution."

```
glm(formula, family = gaussian, data, weights, subset,
    na.action, start = NULL, etastart, mustart, offset,
    control = list(...), model = TRUE, method = "glm.fit",
    x = FALSE, y = TRUE, singular.ok = TRUE, contrasts = NULL, ...)
```

2.2.3.1 Attributes Read the documentation, there's so much. Key points: - **formula:** the symbolic description of the model (uses the ~) - **family:** the error distribution and link function you're using. - binomial: logistical regression - **data:** the data. Not to be confused with Data.

2.2.4 Running the Model

m1 model predicts the probability of attrition based on MonthlyIncome **m2** model predicts the probability of attrition based on OverTime

```r
m1 <- glm(Attrition ~ MonthlyIncome, family = "binomial", data = att)
m2 <- glm(Attrition ~ OverTime, family = "binomial", data = att)
m1
```

```
##
## Call:  glm(formula = Attrition ~ MonthlyIncome, family = "binomial",
##     data = att)
##
## Coefficients:
##   (Intercept)  MonthlyIncome
##    -0.9291087     -0.0001271
##
## Degrees of Freedom: 1469 Total (i.e. Null);  1468 Residual
## Null Deviance:       1299
## Residual Deviance: 1253  AIC: 1257
```

```
m2
```

```
##
## Call:  glm(formula = Attrition ~ OverTime, family = "binomial", data = att)
##
## Coefficients:
## (Intercept)  OverTimeYes
##      -2.150        1.327
##
## Degrees of Freedom: 1469 Total (i.e. Null);  1468 Residual
## Null Deviance:       1299
## Residual Deviance: 1217  AIC: 1221
```

2.2.4.1 Interpreting the Logistic Model m1, monthly income - Coefficient of -0.0001271, slightly negatively correlated with attrition - Higher income means slightly less likely to leave m2, OverTimeYes - Coefficient of 1.327, highly correlated with attrition - Working overtime correlates to leaving

HOWEVER - This isn't meaningful until we put it in an exponential function - I'm not even sure the last section is accurate, he taught this badly

```
exp(coef(m1))
```

2.2.4.1.1 Exponential Function (what's your conjunction?)

```
##   (Intercept) MonthlyIncome
##     0.3949055     0.9998729
```

```
exp(coef(m2))
```

```
## (Intercept) OverTimeYes
##   0.1165254   3.7712488
```

2.2.4.2 Final Final Interpretation (Final) Verbatim from Professor for me to parse later when my brain works:

m1: The odds of employee attrition in model 1 increased by almost 1 for every $1 increase in monthly income

m2: The odds of employee attrition in model 2 increased by almost 3.77 for employees who work overtime compared to those who do not