Generalized Linear models

Mark Andrews

November 14, 2018

Introduction

In generalized linear models, we model the outcome variable as a random variable whose parameters are transformed linear functions of some of more predictors variables.

```
library(dplyr)
library(magrittr)
library(readr)
library(pander)
library(tidyr)
library(tibble)
```

Logistic regression

In a binary logistic regression, we model the outcome variable as Bernoulli random variable with a parameter p, and where the log odds of p is a linear function of predictor variables. In other words, for all i,

$$y_i \sim \mathrm{dbern}(p_i),$$
$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \sum_{k=1}^{K} \beta_k x_{ki}$$

We'll load up some data about the Titanic.

```
Df <- read_csv('../data/TitanicSurvival.csv') %>%
  select(class = passengerClass,
         sex,
         age,
         survived) %>%
  mutate(survived = ifelse(survived=='yes', T, F))
```

Now, we'll model how the probability of surviving by `sex`:

```
M <- glm(survived ~ sex,
         data=Df,
         family=binomial)
```

We can look at the results as follows:

```
summary(M)
```

```
##
## Call:
## glm(formula = survived ~ sex, family = binomial, data = Df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -1.6124  -0.6511  -0.6511   0.7977   1.8196
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.9818     0.1040   9.437   <2e-16 ***
## sexmale      -2.4254     0.1360 -17.832   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1741.0  on 1308  degrees of freedom
## Residual deviance: 1368.1  on 1307  degrees of freedom
## AIC: 1372.1
##
## Number of Fisher Scoring iterations: 4
```

Predictions

As usual, we will make some data to make predictions about:

```
hypothetical_data <- tibble(sex=c('male', 'female'))
```

and then make the predictions

```
predict(M, newdata=hypothetical_data)
```

```
##         1         2
## -1.443625  0.981813
```

These predictions are in log odds units, so we can convert to probabilities using the inverse logit function, which we can make ourselves:

```
ilogit <- function(x){1/(1+exp(-x))}

logodds <- predict(M, newdata=hypothetical_data) # these are log odds
names(logodds) <- c('male', 'memale')
ilogit(logodds)
```

```
##      male    memale
## 0.1909846 0.7274678
```

We can get the same result more easily with the following:

```
predictions <- predict(M, newdata=hypothetical_data, type='response')
names(predictions) <- c('Male', 'Female')
predictions
```

```
##      Male    Female
## 0.1909846 0.7274678
```

Or better yet, we attach the predicted probabilities to the data frame of hypothetical values:

```
hypothetical_data %<>%
  mutate(prediction = predict(M, newdata = ., type = 'response'))
```

Model comparison

We will model Titanic survival using two different models, i.e. two models with different numbers of predictors:

```
# Use sex and passenger and their interaction
M_full <- glm(survived ~ sex*class, # equivalent to sex + class + sex:class
              data=Df,
              family=binomial)

# This is our comparison model, i.e. no interaction effect
M_null <- glm(survived ~ sex + class,
              data=Df,
              family=binomial)
```

We do model comparison by way of a log likelihood test:

```
ll_test <- anova(M_null, M, test='Chisq')
pander(ll_test, missing='')
```

Table 1: Analysis of Deviance Table

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
1305	1257			
1307	1368	-2	-110.9	8.366e-25

Binomial logistic regression

In binomial logistic regression, our data are counts of number of "successes" out of a total number of trials. To obtain appropriate data, we'll calculate the number of survivors and non-survivors per each `class` by `sex` combination.

```
Df_agg <- group_by(Df, class, sex) %>%
  summarize(survived = sum(survived == TRUE),
            perished = n() - survived
  )
```

Now, we do the logistic regression similarly, but not identically, to before:

```
M <- glm(cbind(survived, perished) ~ sex*class,
         family = binomial,
         data = Df_agg)
```

The results are identical to the model `M_full` above.

Poisson regression

In Poisson regression, our outcome variables are counts, i.e. discrete frequencies, and so each $y_i \in 0, 1 \ldots$, and our probabilistic model of the data is as follows:

$$y_i \sim \text{dpois}(\lambda_i),$$

$$\log(\lambda_i) = \beta_0 + \sum_{k=1}^{K} \beta_k x_{ki}$$

To explore this type of model, we will use the `affairs.csv` data-set:

```
Df <- read_csv('../data/affairs.csv')
```

And we'll model the frequencies of extra-marital affairs as a function of all the predictors:

```
M <- glm(affairs ~ gender + age + yearsmarried
         + children + religiousness + education
         + occupation + rating,
         data=Df,
         family=poisson)
```

As before, we can do model comparisons.

```
M_null <- glm(affairs ~ gender + age,
              data=Df,
              family=poisson)

# Model fit comparison of null and full based on the "Deviance"
ll.test <- anova(M_null, M, test='Chisq')
pander(ll.test, missing='')
```

Table 2: Analysis of Deviance Table

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
598	2887			
592	2360	6	527.4	1.071e-110

And we can do predictions (here using `M_null` for convenience):

```
Df_h <- tibble(gender = c('male', 'female'),
               age = median(Df$age))
Df_h %<>%
  mutate(prediction = predict(M_null, newdata = ., type = 'response'))
```