Generalized Linear models

Mark Andrews

April 5, 2017

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(pander)
```

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,

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

We'll load up some data about extra marital affairs.

```
load('../data/affairs.Rda')
```

Let's take a look:

```
head(Affairs)
```

```
##    affairs gender age yearsmarried children religiousness education
## 4        0   male  37        10.00       no             3        18
## 5        0 female  27         4.00       no             4        14
## 11       0 female  32        15.00      yes             1        12
## 16       0   male  57        15.00      yes             5        18
## 23       0   male  22         0.75       no             2        17
## 29       0 female  32         1.50       no             2        17
##    occupation rating
## 4           7      4
## 5           6      4
## 11          1      4
## 16          6      5
## 23          6      3
## 29          5      5
```

We create a new variable that indicates if someone cheats or not:

```
Affairs$cheater <- Affairs$affairs > 0
```

Now, we'll model how the probability of cheating varies by gender:

```
M <- glm(cheater ~ gender,
         data=Affairs,
         family=binomial)
```

Predictions

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

```
hypothetical.data <- data.frame(gender=c('male', 'female'))
```

and then make the predictions

```
predict(M, newdata=hypothetical.data)
```

```
##          1          2
## -0.9808293 -1.2163953
```

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', 'Female')
ilogit(logodds)
```

```
##      Male    Female
## 0.2727273 0.2285714
```

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
## 0.2727273 0.2285714
```

Model comparison

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

```
### Using all predictors
M <- glm(cheater ~ gender + age + yearsmarried
         + children + religiousness + education
         + occupation + rating,
         data=Affairs,
         family=binomial)

# This is the "null" model, i.e. no predictors
M.null <- glm(cheater ~ 1,
              data=Affairs,
              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)
600	675.4			
592	609.5	8	65.87	3.252e-11

Poisson regression

Instead of modelling the probability of cheating, we can model the number of affairs people have, using a Poisson regression model:

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

M.null <- glm(affairs ~ 1,
              data=Affairs,
              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)
600	2925			
592	2360	8	565.9	4.977e-117