

GLMM

Nick Syring

2022-11-12

Contents

1	Introduction	5
2	Introduction	7
3	Poisson Regression	9
3.1	Children Ever Born Data	9
3.2	Defining GLMs	10
3.3	Fitting GLMs	11

Chapter 1

Introduction

Chapter 2

Introduction

Chapter 3

Poisson Regression

3.1 Children Ever Born Data

The “Children Ever Born” (CEB) dataset consists of grouped data on the number of births of Fijian women. The women are described according to their marriage duration in ordinal levels: (1=0-4, 2=5-9, 3=10-14, 4=15-19, 5=20-24, 6=25-29); their place of residence (Suva, the capital city; Urban; Rural); and, their level of education (none, lower primary, upper primary, secondary or greater). The count, mean, and variance of the number of children ever born, and the group size, is given for each group of women by cross-classified factorial level. These summaries are sufficient to model counts of children ever born by a Poisson distribution (each individual woman’s count is not needed).

The CEB data is an example of an observational dataset — the characteristics of the individuals are inherent rather than set by experimenters as in an experimental/controlled trial, that should be clear from the context. Several interesting questions may be answered using this data, such as: are more or fewer born children associated with higher or lower education among Fijian women; does an urban versus rural living location influence the number of children ever born; and, do the number of children ever born steadily increase with marriage duration, or tend to plateau?

```
ceb <- read.table('ceb.dat')
head(ceb)
```

```
##   dur   res educ mean  var  n    y
## 1 0-4  Suva  none 0.50 1.14  8  4.00
## 2 0-4  Suva lower 1.14 0.73 21 23.94
## 3 0-4  Suva upper 0.90 0.67 42 37.80
## 4 0-4  Suva  sec+ 0.73 0.48 51 37.23
```

```
## 5 0-4 urban none 1.17 1.06 12 14.04
## 6 0-4 urban lower 0.85 1.59 27 22.95
```

A statistician (or student statistician) familiar with multiple linear regression and/or ANOVA for factorial experiments may instinctively choose to fit a Gauss-Markov linear model to the CEB data, treating the responses as independent normal random variables. However, since the responses are counts, a Poisson model is more reasonable. But, just how does one perform *Poisson regression*? — as opposed to the familiar multiple linear regression described by the Gauss-Markov model:

$$Y = X\beta + \epsilon, \quad \epsilon \sim N_n(0_{n \times 1}, \sigma^2 I_n).$$

That is the motivation for this chapter, in which we will explore the family of *Generalized Linear Models* from defining and fitting the model, to performing inference and model diagnostics, all within the context of the CEB example.

3.2 Defining GLMs

For the CEB data we naturally want to model the CEB grouped counts as realizations of Poisson r.v.'s with means $n_j x_j^\top \beta$ where n_j is the number of women in the j^{th} factorial group, x_j is the vector of their common covariates, and β is the common regression coefficient vector. Then, the likelihood of the model is

$$L(\beta; \text{data}) = \prod_{j=1}^{70} \frac{(n_j x_j^\top \beta)^{y_j} e^{-n_j x_j^\top \beta}}{y_j!}$$

and the loglikelihood is given by

$$\ell(\beta; \text{data}) = \text{constant} + \sum_{j=1}^{70} y_j \log(n_j x_j^\top \beta) - n_j x_j^\top \beta.$$

The Poisson likelihood is a member of the Exponential Family, which contains all distributions with PDFs that may be expressed as

$$f(y; \theta, \phi) = \exp\{[y\theta - b(\theta)]/a(\phi) + c(y, \phi)\}.$$

Looking ahead, we will apply the exponential family model above to independent but not identically distributed responses, similar to the data we encounter in multiple linear regression and the Gauss-Markov model, so we will allow θ as well as the forms of the a , b , and c functions to vary over observations, but we will fix ϕ , so that the loglikelihood for a sample of size n may be written as follows:

$$\ell(\beta; \text{data}) = \sum_{i=1}^n \{[y_i \theta_i - b_i(\theta_i)]/a_i(\phi) + c_i(\phi, y_i)\}.$$

The Poisson regression model is a fairly simple member of this family, having $\theta = \log(n_j x_j^\top \beta)$, $\phi = a(\phi) = 1$, and $b(\theta) = \exp(\theta) = n_j x_j^\top \beta$. In fact, it is very often the case that GLMs satisfy $a(\phi) \propto \phi$ up to a known constant. In general, GLMs satisfy

$$E(Y) = b'(\theta) \quad \text{and} \quad V(Y) = b''(\theta)a(\phi).$$

For the Poisson regression model, in particular, we have

$$E(Y) = b'(\theta) = \frac{\partial}{\partial \theta} \exp(\theta) = \exp(\theta) = n_j x_j^\top \beta; \text{ and,}$$

$$V(Y) = b''(\theta)a(\phi) = \frac{\partial^2}{\partial \theta^2} \exp(\theta) = \exp(\theta) = n_j x_j^\top \beta.$$

3.3 Fitting GLMs

Like any other model defined by a likelihood, GLMs may be fit by maximizing the (log)likelihood. But, it is generally not the case that the maximizers (MLEs) are available in closed form. Instead, they are computed iteratively using Newton's method or a similar iterative procedure. Refer again to the exponential family loglikelihood using the usual representation $a_i(\phi) = \phi/w_i$ where w_i are known constants:

$$\ell(\beta; \text{data}) = \sum_{i=1}^n \{w_i[y_i \theta_i - b_i(\theta_i)]/\phi + c_i(\phi, y_i)\}.$$

Write $g(\mu_i) = x_i^\top \beta$ where μ_i is the mean of response Y_i and g is the “link function” which determines a transformation of the mean that is linear in the covariates; for example, for the Poisson regression model, we have $\log \mu_j = n_j x_j^\top \beta$ so the link function is the logarithm. Since $b'_i(\theta_i)$ is also equal to μ_i in the exponential family, we may differentiate the loglikelihood with respect to the regression parameter β using the chain rule:

$$\frac{\partial \ell}{\partial \beta_j} = \sum_{i=1}^n \left\{ \frac{w_i}{\phi} \left[y_i \frac{\partial \theta_i}{\partial \beta_j} - \frac{\partial b_i(\theta_i)}{\partial \beta_j} \right] + c_i(\phi, y_i) \right\}$$

using

$$\frac{\partial \theta_i}{\partial \beta_j} = \frac{\partial \theta_i}{\partial \mu_i} \frac{\partial \mu_i}{\partial \beta_j}.$$

Since $\mu_i = b'_i(\theta_i)$ we have $\partial \theta_i / \partial \mu_i = 1/b''_i(\theta_i)$. And, since $\mu_i = g^{-1}(x_i^\top \beta)$ we have $\partial \mu_i / \partial \beta_j = x_{ij} / g'[g^{-1}(x_i^\top \beta)]$. Substituting, we can write the score function using only μ_i as follows:

$$\frac{\partial \ell}{\partial \beta_j} = \frac{1}{\phi} \sum_{i=1}^n \frac{y_i - \mu_i}{g'(\mu_i)V(\mu_i)} x_{ij}.$$

The second (mixed partial) derivative may be written

$$\frac{\partial^2 \ell}{\partial \beta_j \partial \beta_k} = -\frac{1}{\phi} \sum_{i=1}^n \frac{x_{ij} x_{ik} h(\mu_i)}{g'(\mu_i)^2 V(\mu_i)}$$

where $h(\mu_i) = 1 + (y_i - \mu_i)\{V'(\mu_i)/V(\mu_i) + g''(\mu_i)/g'(\mu_i)\}$. The expectation of the second derivative (which when multiplied by -1 appears in the Fisher information matrix) is the same quantity with $h(\mu_i)$ replaced by $E[h(\mu_i)]$, which simply equals 1 because $E(Y_i - \mu_i) = 0$. The Hessian of the loglikelihood is clearly a quadratic form $\phi^{-1} X^\top W X$ where X is the $n \times p$ design matrix of covariates and $W = [h(\mu_i)/\{g'(\mu_i)^2 V(\mu_i)\}]$ is an $n \times n$ diagonal matrix of “weights”. Less obvious, we may define $G = \text{diag}\{g'(\mu_i)/h(\mu_i)\}$ so that the gradient of the loglikelihood equals $\phi^{-1} X^\top W G(y - \mu)$. With this clever rewriting, Newton’s method updates take on the form of a weighted least squares solution:

$$\begin{aligned} \beta^{[k+1]} &= \beta^{[k]} + (X^\top W X)^{-1} X^\top W G(y - \mu) \\ &= (X^\top W X)^{-1} X^\top W \{G(y - \mu)X + \beta^{[k]}\} \\ &= (X^\top W X)^{-1} X^\top W z \end{aligned}$$

where $z := G(y - \mu)X + \beta^{[k]}$ is sometimes referred to as the “pseudo-data”. Repeating the weighted least squares update, iteratively, until convergence, is termed *iteratively re-weighted least squares* (IRLS) since, of course, the weights in W are updating with each iteration.

For our Poisson regression based on the grouped CEB data we have the following likelihood, gradient, and Hessian:

$$\begin{aligned} \ell(\beta; \text{data}) &= \sum_{j=1}^{70} [y_j x_j^\top \beta - n_j e^{x_j^\top \beta}] \\ \nabla_s \ell &= \sum_{j=1}^{70} [y_j x_{js} - n_j x_{js} e^{x_j^\top \beta}] \\ \nabla_{s,t}^2 \ell &= - \sum_{j=1}^{70} n_j x_{js} x_{jt} e^{x_j^\top \beta}. \end{aligned}$$

Rewriting the Hessian and gradient as above for the general exponential family GLM we have

$$W_{k,k} = n_k \mu_k \quad \text{and} \quad G_k = (n_k \mu_k)^{-1}$$

so that the IRLS updates are given by

$$(X^\top W X)^{-1} X^\top W z$$

with $z_k = (n_k \mu_k)^{-1}(y_k - n_k \mu_k) + x_k^\top \beta$.

3.3.1 IRLS for the CEB data

```

n <- nrow(ceb)
group.sizes <- ceb$n
Y <- ceb$y
# IRLS - factor coding
# initialize with mu = Y/group.sizes
options(contrasts = c('contr.treatment', 'contr.treatment'))
X <- model.matrix(y~dur+res+educ, data = ceb)
mu <- Y/group.sizes
XB <- log(mu)
W <- diag(as.numeric(mu))
z <- -(1/mu)*(Y/group.sizes-mu) + XB
beta <- solve(t(X)%*%W%*%X)%*%t(X)%*%W%*%z
tol <- 0.0001
difference <- 1
maxiter <- 100
iter <- 1
while((difference > tol) & (iter < maxiter)){
  XB <- X%*%beta
  mu <- exp(XB)
  W <- diag(as.numeric(group.sizes*mu))
  z <- (Y/diag(W) - rep(1,n)) + XB
  beta.old <- beta
  beta <- solve(t(X)%*%W%*%X)%*%t(X)%*%W%*%z
  difference <- max(abs(beta - beta.old))
  iter<-iter+1
}
beta

```

```

##                [,1]
## (Intercept)  0.05695417
## dur10-14      1.37053208
## dur15-19      1.61423104
## dur20-24      1.78548879
## dur25-29      1.97679469
## dur5-9        0.99765038
## resSuva      -0.15121728
## resurban     -0.03895822
## educnone     -0.02308034
## educsec+     -0.33266474
## educupper    -0.12474575

```

```
## the glm function can be used with offset equal to logarithm of the group sizes
summary(glm(round(y)~dur+res+educ, family = poisson(link = 'log'), data = ceb, offset = log(n)))

##
## Call:
## glm(formula = round(y) ~ dur + res + educ, family = poisson(link = "log"),
##      data = ceb, offset = log(n))
##
## Deviance Residuals:
##      Min        1Q      Median        3Q      Max
## -2.2960  -0.6641   0.0725   0.6336   3.6782
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.05754    0.04803   1.198   0.231
## dur10-14     1.36940    0.05107  26.815 < 2e-16 ***
## dur15-19     1.61376    0.05119  31.522 < 2e-16 ***
## dur20-24     1.78491    0.05121  34.852 < 2e-16 ***
## dur25-29     1.97641    0.05003  39.501 < 2e-16 ***
## dur5-9       0.99693    0.05274  18.902 < 2e-16 ***
## resSuva     -0.15166    0.02833  -5.353 8.63e-08 ***
## resurban    -0.03924    0.02463  -1.594  0.111
## educnone    -0.02297    0.02266  -1.014  0.311
## educsec+    -0.33312    0.05390  -6.180 6.41e-10 ***
## educupper   -0.12425    0.03000  -4.142 3.44e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 3731.852  on 69  degrees of freedom
## Residual deviance:  70.665  on 59  degrees of freedom
## AIC: 522.14
##
## Number of Fisher Scoring iterations: 4
```