Design Document for Negative Binomial Regression: R Package `glmm`

Sydney Benson

August 2019

Abstract

This design document describes the process of creating a generalized linear mixed model with a response variable having a negative binomial distribution.

1 Introduction

The R package `glmm` is expanding to include the ability to create a generalized linear mixed model with a negative binomial response variable. Variables having a negative binomial distribution contain observations which count the number of trials until a specified number of successes. This is similar to the Poisson distribution which counts the number of successes within a given time frame; however, the Poisson distribution requires that the mean and variance of the variable be the same. This leads to the issue of overdispersion, greater variability in a data set than would be expected in the statistical model. The negative binomial distribution, which does not require the variable's mean and variance to be the same, does not risk overdispersion making it optimal for use with count data that do not have equal mean and variance.

2 Data

2.1 NMES1988

One data option to illustrate the use of these improvements is the NMES1988 data set from the R package `AER`. This is cross sectional data from the U.S. National Medical Expenditure Survey conducted in 1987 and 1988. The data is a sample of individuals age 66 and older who are covered by Medicare. The data set includes information on the number of visits made to various categories

of medical offices and a breakdown of how many visits were made to each category of office. The data also includes information on each of the individuals personal characteristics, including gender and race, their economic well-being, and any additional insurance that the individual is covered by.

2.2 grouseticks

The second data option to illustrate the use of this improvement to the R package `glmm` is the grouseticks data set from the R package `lme4`. This is a simpler data set which observes the counts of the number of ticks on the heads of red grouse chicks sampled from various fields. The data set also contains information on the individual chicks' brood and geographical information about the fields the chicks were sampled from.

3 The Negative Binomial Family

Let g be the canonical link function and μ be a vector of length n such that

$$g(\mu) = X\beta + Zu \tag{1}$$

The choice of the link function is related to the distribution of the data, $f_\theta(y|u)$. If the data have a negative binomial distribution, the link is $\log(\mu)$. Following the method discussed in the online source from the University of California Berkeley, X is defined as a variable following a negative binomial distribution with number of failures r and probability of success μ. Then, the density function is

$$p(x) = \binom{x+r-1}{x} (1-\mu)^r \mu^x \tag{2}$$

$$= \binom{x+r-1}{x} \exp\left(r \log\left(1-\mu\right) + x \log\left(\mu\right)\right) \tag{3}$$

Thus, $h(x) = \binom{x+r-1}{x}$, $T(x) = x$, $\eta = \log(\mu)$ and the cumulant function, $c(\eta) = -r\log(1-\mu)$ or $c(\eta) = r \log\left(\frac{1}{1-e^\eta}\right)$.

For simplicity of future notation, let $\eta = g(\mu) = X\beta + Zu$. Let $c(\eta)$ denote the cumulant function such that the log of the data density can be written as

$$y'\eta - c(\eta) = \sum_i \left[y_i\eta_i - c(\eta_i)\right] \tag{4}$$

The user is required to specify the family in the model-fitting function. Once the family is specified, many family-specific functions are called. They are contained in an S3 class called

"glmm.family". The negative binomial family function outputs a list including the family name ("neg.binomial.glmm"), a function that calculates the value of the cumulant function $c(\eta)$, a function that calculates the cumulant's first derivative $c'(\eta)$ with the derivative taken with respect to η, and a function that calculates the cumulant's second derivative $c''(\eta)$.

The users provide the family in the model-fitting function by either enter the character string ("neg.binomial.glmm") or the function (`neg.binomial.glmm()`). Then, calculating $c(\eta_i), c'(\eta_i)$, and $c''(\eta_i)$ is as simple as:

```
neg.binomial.glmm()$cum(args)
neg.binomial.glmm()$cp(args)
neg.binomial.glmm()$cpp(args)
```

For the negative binomial distribution, we calculate these values (`cum`, `cp`, and `cpp`) as follows.

$$c(\eta_i) = r \log\left(\frac{1}{1 - e^{\eta_i}}\right) \tag{5}$$

$$c'(\eta_i) = r\frac{e^{\eta_i}}{1 - e^{\eta_i}} \tag{6}$$

$$c''(\eta_i) = r\frac{e^{\eta_i}}{(1 - e^{\eta_i})^2} \tag{7}$$

Then we use these pieces to create the scalar $c(\eta)$, the vector $c'(\eta)$ and the matrix $c''(\eta)$. We calculate

$$c(\eta) = \sum_i c(\eta_i). \tag{8}$$

The vector $c'(\eta)$ has components $c'(\eta_i)$. The matrix $c''(\eta)$ is diagonal with diagonal elements $c''(\eta_i)$.

Also, this family of functions contains a check to ensure the data are valid given the family type. For `neg.binomial.glmm`, the data should be nonnegative. If the data set does not pass the check, the check returns an error message.

3.1 The Likelihood

If Y_i are negative binomially distributed then $E[Y_i|U = u] = \mu_i$ and $Var[Y_i|U = u] = \mu + \frac{\mu^2}{\alpha}$. We can then give the response vector's density given the random effects as

$$f_{\alpha,\beta}(y|u) = exp(y^T(X\beta + Zu) + \sum_{i=1}^{n} \log \frac{\Gamma(y_i + \alpha) \cdot \alpha^\alpha}{\Gamma(\alpha) \cdot (exp(X\beta + Zu) + \alpha)^{y_i + \alpha}}) \tag{9}$$

and the density of the random effects distribution as $f_\nu(u)$. Thus, the likelihood can be represented by

$$L(\theta|y) = \log \int f_{\alpha,\beta}(y|u) f_\nu(u) du \tag{10}$$

$$= \log \int \Big[exp(y^T(X\beta + Zu) + \sum_{i=1}^{n} \log \frac{\Gamma(y_i + \alpha) \cdot \alpha^\alpha}{\Gamma(\alpha) \cdot (exp(X\beta + Zu) + \alpha)^{y_i + \alpha}} \Big] f_\nu(u) du. \tag{11}$$

4 Checks

4.1 Checking the cumulant finite differences

To check that the first and second derivatives of the cumulant are calculated correctly, we use finite differences. To do this, we chose a value of $\theta = (\beta, \sigma)$, α, and a relatively small value of δ. We can check that the value and first derivative of the cumulant function are calculated correctly by making sure the following approximation holds

$$\nabla c(\theta, \alpha) \cdot \delta \approx c(\theta + \delta, \alpha) - c(\theta, \alpha). \tag{12}$$

Then, we can check the first and second derivatives of the cumulant are calculated correctly by checking for the following approximation:

$$\nabla^2 c(\theta, \alpha) \cdot \delta \approx \nabla c(\theta + \delta, \alpha) - \nabla c(\theta, \alpha) \tag{13}$$