Paper 2694-2018

Alternative Variance Parameterizations in Count Data Models with the NLMIXED Procedure

Robin High, University of Nebraska Medical Center, Omaha, NE

SAS®/STAT and SAS/ETS software have several procedures for working with count data based on the Poisson or negative binomial distributions. In particular, the GENMOD and GLIMMIX procedures offer the most conventional approaches for estimating model coefficients and assessing goodness of fit and also for working with correlated data. In addition, the COUNTREG procedure includes the Conway-Maxwell Poisson distribution with a statement to model dispersion and the negative binomial distribution with two different variance functions. The FMM procedure includes a generalized Poisson distribution as a method to deal with over-dispersion in count data. However, programming statements entered into the NLMIXED procedure in SAS/STAT can also model count data with other distributions and in particular to estimate the dispersion parameter as a function of covariates, and as a result modify the variance function to improve the fit.

INTRODUCTION

A statistical model based on the normal distribution is the only data analysis situation where the variance of an observation is independent of the mean. With response data having non-negative integer values (count data) the variance of an observation is a function of the mean of the distribution and a dispersion parameter often labeled in the probability density functions with various Greek symbols (ν, α, θ, k, φ, τ which in the SAS code are usually given the variable names nu, alpha, theta, k, phi, or tau). Estimation of this dispersion parameter with several count data distributions having alternative variance functions will be illustrated in this paper. Several of these distributions are not included among the "conventional" count data models usually applied with the GENMOD or GLIMMIX procedures. A few of these models are available in other SAS procedures, yet are not applied with count data as frequently as they could be, and reasons for choosing them and their interpretation are likely unknown. The objective of this paper is to introduce parameter estimation methods for these models for consideration of when they may be relevant in count data analyses and how to interpret the dispersion parameter. Under- or over-dispersion due to zero-inflation and truncation of the distribution are not covered here, but are direct extensions of these techniques. Also, a brief overview of how to assess the fit of these models is provided. Many of these procedures can be programmed with statements written in the NLMIXED procedure (and in situations with one linear predictor, programming statements entered in PROCs GENMOD or GLIMMIX); they can also be estimated by accessing routines in the R statistical package through the IML procedure (see Appendix for instructions how to run R code with the IML procedure).

THE NLMIXED PROCEDURE

This paper is an extension of "Fitting Statistical Models with the NLMIXED and MCMC Procedures" (High, ElRayes, 2017) which illustrates how to write statements which the NLMIXED procedure needs to estimate parameters for statistical models: the initial parameter estimates (PARMS) followed by equations for the linear predictor eta, the mean (computed with the inverse link), and the log-likelihood equation for a parametric distribution. PREDICT statements compute means and dispersion to assess model fit.

The NLMIXED procedure requires several programming statements to estimate count data models with alternative variance functions. The motivating data set contains two explanatory variables: one categorical variable (group coded as A/B) and one continuous variable x. The linear predictor eta is written such that group level B is the reference category (corresponding to reference category coding). The statements to be entered into the NLMIXED procedure for the examples are:

```
PROC NLMIXED DATA = indat(rename=(response = y)) ;
PARMS b0 2 b1 1 b2 .1 phi .5;       * initial parameter estimates;
 eta = b0 + b1*(group='A') + b2*x;  * linear predictor for the mean;
  mu = exp(eta);                    * inverse of the LOG link;
lglk =  < log-likelihood equation for the pdf with parameters mu and phi > ;
MODEL y ~ GENERAL( lglk ) ;
PREDICT mu OUT=prd(KEEP= < variables in the model, pred > rename(pred=mu) ) ;
TITLE "NLMIXED: < model type > ";
RUN;
```

In the examples demonstrated with PROC NLMIXED, the outcome or response variable always has the name y. Other variable names selected for this variable should be changed to y on the PROC statement with the `rename=(<response>=y)` option attached to the input data set name and also to be sure no other variables in the data set have the name y. The MODEL statement and especially the log-likelihood equation contain the variable name y, which may appear multiple times in the latter, and thus it does not need to be edited. The PARMS statement, linear predictor eta, and the inverse link for these count data models have the same contents for all distributions and will be omitted from most examples. Initial parameter estimates entered into the PARMS statement are often crucial to avoid computational errors. The magnitude and sign of the intercept term is usually the most important estimate to enter. Since the log link is present in most of these models, coefficient values are not large, so initial estimates close to 0 often are preferred (e.g., -0.1 or 0.1 rather than the default value of 1). A method to provide initial estimates entered with an external data set are shown in one of the examples. Any of these count data models having a log link function for the mean (i.e., having the inverse link of the exponential function) can provide incidence rate ratios by exponentiating the parameter estimates for the predictors, thus unifying an approach to interpreting these models.

MODELS BASED ON THE POISSON DISTRIBUTION

With response data collected as non-negative integers, the Poisson distribution may first be considered as a distribution option with either the GENMOD or GLIMMIX procedures:

$$P(Y = y) = (\mu^{y} * e^{-\mu}) / y! \quad \text{for } y = 0, 1, 2, ..$$

However, a severe restriction imposed by the Poisson distribution is the equality of the mean and the variance which is frequently unrealistic due to under- or over-dispersion present in the data. The Poisson distribution is a special case of the distributions to be described here. The primary difference is these distributions include a dispersion parameter which leads to alternative variance functions that are able account for over-dispersion in the data and in a few cases under-dispersion as well. The negative binomial distribution is the most common choice when over-dispersion is present; however, it cannot analyze data that exhibit under-dispersion which is not encountered as frequently as over-dispersion, yet alternatives are available when it does.

THE CONWAY-MAXWELL POISSON REGRESSION MODEL

Two forms of the Conway-Maxwell Poisson (COM) regression model are available in PROC COUNTREG (parameter=mu and parameter=lambda). The mu formulation of the COM is:

$$P(Y=y) = [1/S(\mu,\nu)] * [(\mu^{y} /y!)^{\nu}] \text{ for } y = 0,1,2,3 ..$$

where the index of the function, $S(\mu,\nu) = \Sigma(\mu^{n} /n!)^{\nu}$ ranges from n=0 to ∞

The Poisson distribution is a special case for both the lambda and mu parameterizations when the dispersion parameter ν=1. In the pdf for the COM Poisson distribution, ν represents the rate of decay of the probabilities compared with the Poisson distribution. In the COM distribution models data that exhibit under-dispersion $\nu > 1$ and those with over-dispersion $\nu < 1$. The two types of parameterizations of the COM distribution can be programmed in the NLMIXED procedure. An application of the “lambda” parameterization is described by Morris (2017) as derived by Sellers (2010). The “mu” parameterization is the default in COUNTREG with its derivation given in Guikema (2008). The estimates from the mu parameterization can be compared with the estimated coefficients from a Poisson regression model with the GENMOD and GLIMMIX procedures. An illustration of under-dispersion is encountered with the air-freight data with the number broken as a function of the number of transfers (Neter, et. al. exercise 1.21):

```
DATA freight;
LABEL transfers='Number of Transfers' y="No. of broken items";
INPUT y transfers @@;
CARDS;
16 1  9 0  17 2  12 0  22 3  13 1  8 0  15 1  19 2  11 0
;

PROC GENMOD DATA=freight;
MODEL broken = transfers / dist=poisson;
RUN;
```

Two criteria for assessing goodness of fit (Pearson Chi-Square=0.222 and Deviance=0.227) indicate under-dispersion, since the ratio of Value/DF should be close to 1. The estimated parameter values include the intercept (b0=2.35, se=0.132) and the coefficient for transfers (b1=0.26, se=0.079).

The COM regression model can be run with the COUNTREG procedure:

```
PROC COUNTREG DATA=freight;
MODEL broken = transfers / dist=compoisson parameter=mu;
RUN;
```

The parameter estimates include the intercept (b0=2.39, se=0.054) and the coefficient for transfers (b1=0.26, se=0.032). The parameter estimates are nearly the same with the Poisson and COM models; however, the standard errors for the COM model are considerably smaller due to under-dispersion (implied by the negative value for _lnNu= -1.757). The COM model can be programmed for both the mu and lambda parameterizations with NLMIXED statements where the dispersion parameter ν is called "nu":

```
%LET nitr = 50; * upper limit on number of iterations for under-dispersed data;

* In the NLMIXED code with statements for parameterizations lambda and mu
  choose one of the pairs of statements defined by parameter = ;

PROC NLMIXED data=freight(rename=(broken=y));
PARMS b0 -2 b1 .1
      bn0 1 ;
eta = b0 + b1*transfers;    * linear predictor for the mean;
mean = EXP(eta);            * inverse log link;
etaNu = bn0;                * linear predictor for dispersion;
nu = exp(etaNu);            * inverse log link;
S_ = 1;
DO i_ = 1 to &nitr. ;       * Number of iterations for S_ to converge;
f_ = 1;
 DO n_ = 1 to i_ ;
   *f_ = f_ * ( mean / (n_**nu)); * parameter = lambda;
   f_ = f_ * (( mean /n_)**nu);   * parameter = mu;
 END;
 S_ = S_ + f_;
END;

*lglk =      y*log(mean) - nu*lgamma(y+1)  - log(S_); * lglk for parameter = lambda;
lglk  = nu*( y*LOG(mean) -    lgamma(y+1)) - log(S_); * lglk for parameter = mu;

MODEL y ~ general(lglk);
ESTIMATE '-log(nu)' -bn0;
ESTIMATE 'nu'       EXP(bn0);
TITLE 'NLMIXED: COM Poisson';
RUN;
```

NLMIXED estimates the dispersion parameter, ν, as 5.78 (-LOG(nu)=-1.757 which is printed in the COUNTREG output) thus indicating under-dispersion (a feature of the data when -LOG(nu) < 0). The DO loop in this code specifies 50 iterations for each row of the data file with a macro variable which shows the intensive computational nature of this count data model. As the index i_ increases, the computations for S_ are based on ratios of two extremely large numbers. Although the mathematical formula indicates the value f_ will eventually converge, a few conditions are necessary for it do so with a reasonable upper limit. With data that are clearly under-dispersed (e.g., as previously determined with PROC GENMOD where ν will be greater than 1) convergence occurs quickly, so the upper limit for the number of iterations (i_) may be less than 50 as Figure 1 indicates.

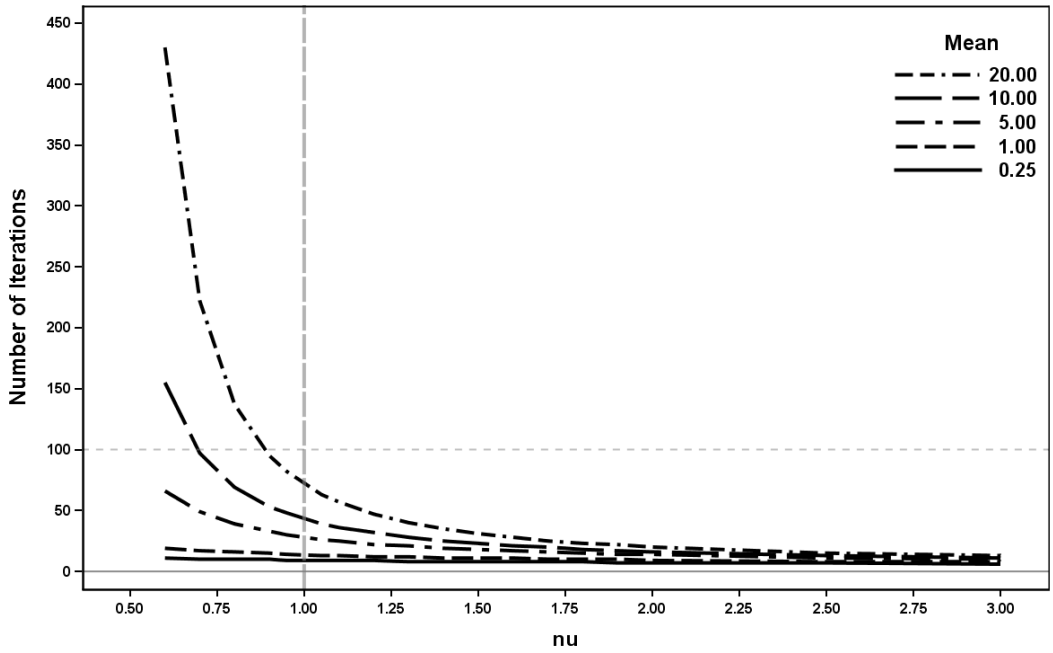

Figure 1. Number of Iterations of COM Loop to Converge within 0.000000001

For observations having relatively large means (e.g., greater than 20) or with data sets having substantial over-dispersion ($\nu < 1$), the dash-dash-dot line indicates the upper limit on the number of iterations for the sum functions to converge may be very large. The rate of convergence for this log-likelihood can be observed by reviewing the computations in a DATA step for a few observations (i.e., place an OUTPUT statement after `S_ = S_ + f_ ;` and change PARMS to RETAIN).

Note these special cases of the COM:

- With $\nu = 1$ the COM distribution is the Poisson.
- With the mean mu less than 1 and as the dispersion parameter ν gets arbitrarily close to 0, the COM converges to the geometric distribution with parameter p = 1 / (1+muCOM), where muCOM is the mean of the COM distribution.
- With the lambda parameterization and binary data, the dispersion parameter ν will be very large (extreme under-dispersion) and the COM Poisson regression model closely approximates a logistic regression model.

The final condition relates to the situation where the response variable is considered a “count” and the data are typically 0 or 1 (however, not necessarily a dichotomous outcome) with the rare possibility the response can still be 2 or more.

COM: MODEL THE MEAN AND DISPERSION

The COUNTREG procedure also has a DISPMODEL statement to estimate the dispersion parameter as a function of covariates (heterogeneous model):

```
PROC COUNTREG DATA=indat ;
CLASS group;
MODEL y = group x / dist=compoisson parameter=mu;
DISPMODEL y ~ group ;
OUTPUT out=prdCR pred=prd nu=nu dispersion=dsprsn variance=vrnce;
RUN;
```

The necessary modifications to the NLMIXED code given above to estimate the dispersion parameter ν as a function of the covariates is to enter a second linear predictor. This approach to estimate the dispersion

parameter with a second linear predictor and the LOG link function for models with overdispersion (by entering the exponential function as the inverse link) applies in general to the models that follow.

GPS: GENERALIZED POISSON REGRESSION

The generalized Poisson distribution was described by Consul and Jain (1973). Consul (1989) provides an extensive description of this distribution and Consul and Famoye (1992) give an example of its use with count data. Like the COM Poisson, its derivation begins with the Poisson distribution and then adds a dispersion parameter which is based on the lognormal distribution thus allowing it to be applied with data having either under- or over dispersion. Two types of GPR distributions exist for count data:

GPS-1: Unrestricted Generalized Poisson

The GPS-1 model is the parameterization defined by Consul (1989) and Harris, Yang, and Hardin (2012). It includes a dispersion parameter δ (dlt) in the unrestricted form.

$$P(Y=y) = (\mu * (\mu + \delta y)^{(y-1)} * e^{-(\mu + \delta y)}) / y! \quad \text{for } y = 0, 1, 2, ..$$

The formula shows how the GPS-1 pdf reduces to the Poisson distribution when δ approaches 0. The log-likelihood can be programmed with PROC NLMIXED where the dispersion parameter δ = dlt:

```
PROC NLMIXED DATA=indat(rename=(rsp=y)) ;
 < enter statements for PARMS, eta, and mu >;
lglk = log(mu) + (y-1)*log(mu + (dlt*y)) - (mu + (dlt*y)) - lgamma(y+1);
MODEL y ~ general( lglk ) ;
ESTIMATE "dispersion" 1 / (1-dlt)**2
PREDICT mu OUT=prd1(keep=y pred < covariates > rename=(pred=mu));
TITLE "GPR-1: Generalized Poisson Regression with dispersion parameter dlt";
RUN;
```

The sign and magnitude of δ indicates the type of dispersion present in the data. When δ=0 the model is equidispersed (i.e., it is Poisson). When $\delta > 0$ the data are overdispersed; when $\delta < 0$ the data exhibit underdispersion. The estimate of dispersion relative to the Poisson model (assuming the mean equals the variance) computed by the ESTIMATE statement should be close to ValueDf for the chi-square or deviance printed in the model fit table of PROC GENMOD.

The FMM procedure fits the generalized Poisson with the dist=gpoisson option on the MODEL statement. The coefficients of the predictor variables from FMM match the coefficients from the GPS-1 output with NLMIXED, though the parameterization of FMM computes a different intercept term. The correspondence between the predicted values, variances, and dispersion parameter of the two approaches is illustrated in the supplementary SAS code file.

GPS-2: Restricted Generalized Poisson

Famoye (1993) demonstrates applications of the restricted generalized Poisson model:

$$P(Y=y)=((\mu/(1+\alpha\mu))^{y} * (1+\alpha y)^{(y-1)} * e^{[(-\mu(1 + \alpha y))/(1+\alpha y)]}) / y! \quad y=0,1,2, ..$$

The GPS-2 pdf reduces to the Poisson distribution when the dispersion parameter α approaches 0. The log-likelihood can be programmed with PROC NLMIXED (with alpha as the dispersion):

```
PROC NLMIXED DATA=indat(rename=(rsp=y));
< enter statements for PARMS, eta, and mu >;
lglk = y*log(mu/(1 + (alpha*mu))) + (y-1)*log(1+(alpha*y))
       + ((-mu*(1+(alpha*y))) / (1 + (alpha*mu))) - lgamma(y+1);
MODEL y ~ general(lglk) ;
PREDICT mu OUT=prd2(keep=y pred < covariates > rename=(pred=mu));
TITLE1 'GPR-2: Restricted Generalized Poisson with dispersion alpha";
RUN;
```

Hilbe (2011, p. 340) incorrectly prints the log likelihood equation for this distribution. Based on the pdf (which is correct at the top of p. 340), the sign of the part of the equations 10.65 and 10.66 with the EXP function is incorrect, which should be addition (not subtraction).

The case of $\alpha = 0$ the model corresponds to the Poisson distribution. When $\alpha < 0$, the data are under-dispersed; if $\alpha > 0$, the data are over-dispersed. Both versions of the GPS models have limitations to the amount of under-dispersion it can evaluate. For the parameters mu and α the restrictions are

$$\text{GPS-1: } \max(-1, -\mu/4) \le \alpha < 1$$
$$\text{GPS-2: } \max(-1/\mu, -1/4) \le \alpha \le 1/\mu$$

There is no theoretical limitation on the amount of over-dispersion with either version of the generalized Poisson model; it is quite effective in dealing with right skewed count data distributions having long tails, though results are often comparable with the negative binomial NB-2 model described below. However, that the range of the response variable depends on the respective dispersion parameter violates one of the standard conditions for consistency and asymptotic normality of maximum likelihood estimation (Cameron and Trivedi, 2013, p. 171).

An important difference between the two generalized Poisson regression models is described in the goodness of fit section below.

THE NEGATIVE BINOMIAL DISTRIBUTION

The negative binomial model is one of several discrete and continuous distributions available as an option with the MODEL statement of PROC NLMIXED:

```
MODEL y ~ NEGBIN(n, p);
```

The two parameters n and p are defined from the negative binomial probability density function:

$$f(y \mid n, p) = \frac{\Gamma(n + y)}{\Gamma(n) * \Gamma(y+1)} * p^{n} * (1-p)^{y}$$

Though the parameter n is typically chose as an integer to demonstrate its connection with the binomial distribution, “n” can assume any positive real number. The parameterization invoked with the negative binomial option on the MODEL statement in PROC NLMIXED differs from that in the FMM, GLIMMIX, and GENMOD procedures. For maximum likelihood estimation in PROC NLMIXED equivalent results to these SAS procedures can be obtained with these statements:

```
PROC NLMIXED DATA = indat(rename(rsp=y));
< enter statements for PARMS, eta, and mu (LOG link) >;
p = 1 /(1 + (mu*phi));
MODEL y ~ NEGBIN(1/phi, p);
TITLE 'NLMIXED: Negative Binomial with MODEL statement';
RUN;
```

The models in this section are considered extensions or alternatives to the conventional negative binomial distribution. They are described in greater detail in chapter 10 of “Negative Binomial Regression” (2011) and also chapters 5 and 6 of “Modeling Count Data” (2014) by Joseph Hilbe where computations are primarily illustrated with code from R, STATA, and LIMDEP. SAS/STAT or SAS/ETS software does not currently include some of them as modeling options, especially in the GENMOD and GLIMMIX procedures. However, all of them can be estimated with SAS through programming statements entered into the NLMIXED procedure.

NB2: THE NEGATIVE BINOMIAL MODEL (P=2)

Computations of the negative binomial regression model with the NLMIXED procedure defined above are equivalent to PROCs GENMOD, FMM, and GLIMMIX by directly entering the log-likelihood equation for the negative binomial with dispersion parameter phi:

```
PROC NLMIXED DATA=indat(rename=(rsp=y));
< enter statements for PARMS, eta, and mu (LOG link) >;
lglk = y*log(phi*mu) - (y+(1/phi))*log(1+(phi*mu))
      + lgamma(y+(1/phi)) - lgamma(1/phi) - lgamma(y+1) ;
MODEL y ~ general(lglk);
TITLE 'NLMIXED: Negative Binomial with log-likelihood equation';
RUN;
```

where mu is the mean (a function of the parameter estimates in the linear predictor) and phi is the dispersion parameter for the pdf for the negative binomial distribution. The NB-2 distribution has a variance function $Var(Y) = \mu + \varphi*\mu^2$, identified with the exponent 2 in the variance function. With the log link [the inverse link is EXP(eta)], the Poisson distribution is a limiting case of the negative binomial distribution as φ approaches 0 from the right (Hilbe, p. 221); that is, when φ is computationally close to 0, results from the Poisson and negative binomial distributions, both having a log link, are nearly the same. For count data with under-dispersion, the Poisson and negative binomial results will be the same with dispersion=0 (if the NB-2 model converges).

GEOM: GEOMETRIC REGRESSION MODEL

The geometric distribution is a special case of the negative binomial NB-2 model where the pdf is parameterized with the mean μ:

$$Prob(Y=y) = [\ 1/(1+\mu)\ ] * [\mu/(1+\mu)]^y \quad \text{for } y = 0, 1, 2, 3, ..$$

The log-likelihood equation entered into NLMIXED for the geometric distribution is thus:

```
lglk = y*log(mu) - (y+1)*log(1+mu);
```

To derive this equation from the NB-2 distribution, let phi=1 in the log-likelihood equation:

```
lglk = y*log(phi*mu) - (y+(1/phi))*log(1+(phi*mu))
     + lgamma(y+(1/phi)) - lgamma(1/phi) - lgamma(y+1) ;
```

The lgamma functions cancel each other since lgamma(1/1) = 0 and lgamma(y+(1/1)) = lgamma(y+1), so the sum of the items printed on the second line of the equation is 0, again resulting in the lglk for the geometric.

When a negative binomial model has a dispersion parameter close to 1, the geometric distribution may be an alternative. To estimate the geometric distribution, substitute this log-likelihood in the programming statements in PROC NLMIXED for the negative binomial distribution. The choice between the negative binomial or geometric distributions can be evaluated by comparing the -2 Log Likelihood values or the fit statistics (e.g., AIC or BIC) of the two models. A quick test can also be done with an ESTIMATE statement in NLMIXED with the negative binomial model which will test whether phi is significantly difference from 1:

```
ESTIMATE "HO: phi = 1?" phi - 1;
```

The geometric count data model is available in the FMM procedure:

```
PROC FMM DATA=indat;
CLASS group ;
MODEL y = group x / dist=geometric;
TITLE 'FMM: Geometric';
RUN;
```

Even if the geometric model fits well, the sources of over-dispersion should be evaluated with a heterogeneous negative binomial NB-2 model (described below and in the examples).

NB1: THE LINEAR NEGATIVE BINOMIAL MODEL (P=1)

The NB-1 distribution is defined by the form of its variance function $Var(Y)=\mu+\varphi*\mu^1$ where the exponent of 1 placed on μ produces a linear function for the variance . The COUNTREG procedure from SAS/ETS software has an option to estimate the linear negative binomial model:

```
PROC COUNTREG DATA=indat;
CLASS group;
MODEL y = group x / dist=negbin(p=1);
RUN;
```

The log-likelihood for the NB-1 regression model is derived from the log-likelihood for the NB-2 model as presented above with phi replaced by phi/mu (Cameron and Trevedi, 2013).

```
PROC NLMIXED DATA = indat(rename=(rsp=y));
< enter statements for PARMS, eta, and mu >;
lglk = y*log(phi) - (y+(1/(phi/mu)))*log(1+phi)
     + lgamma(y+(1/(phi/mu))) - lgamma(1/(phi/mu)) - lgamma(y+1) ;
MODEL y ~ GENERAL(lglk);
TITLE "NB-1: Linear Negative Binomial";
RUN;
```

NB-P: GENERALIZED NEGATIVE BINOMIAL

The generalized negative binomial model contains three parameters to be estimated: mu, k, and Q (from which P is derived as shown in the NLMIXED code). The generalized negative binomial model (NB-P) allows more flexibility in estimating the variance than is available with either the NB-1 and NB-2 models. The generalized formula allows the estimation of the exponent placed on the second mu in the variance function:

$$Var(Y) = \mu + \varphi*\mu^{P}$$

The pdf for NB-P is obtained from the NB-2 distribution with φ replaced with $\varphi*\mu^{(2-P)}$. In the derivation, the NLMIXED log-likelihood equation, the exponent 2-P is substituted with Q so that the pdf for NB-2 has Q=0 and the pdf for NB-1 has Q=1 (see Greene, 2008, p. 586 for details concerning the derivation of the three NB pdfs). The log-likelihood equation entered into PROC NLMIXED is the revised equation 10.47 from Hilbe (2011, the eq. on the bottom of p. 324 contains an error corrected in the errata list available on the book's website).

```
 PROC NLMIXED DATA = indat(rename=(rsp=y)) technique=trureg;
 PARMS Q 1 / data=NB2Prms;
 < enter statements for eta and mu >;
lglk = ( (1/phi)*(mu**Q) * log( ((1/phi)*(mu**Q))/(((1/phi)*(mu**Q)) + mu) ))
       + (y*log(1 - (((1/phi)*(mu**Q)) / (((1/phi)*(mu**Q)) + mu))) )
       + lgamma(y + ((1/phi)*(mu**Q))) - lgamma((1/phi)*(mu**Q)) - lgamma(y+1);
MODEL y ~ general(lglk) ;
ESTIMATE 'P'  (2 - Q) ;
RUN;
```

Getting initial estimates for the parameters can be obtained by running PROC COUNTREG or GENMOD for the NB-2 model and then either insert them directly following the coefficient names listed in the PARMS statement or read them in from an external file (the names listed under the variable name "Parameter" must match coefficient names included in the linear predictor). Also, entering `technique=trureg` option on the PROC statement may improve estimation results.

One application of the NB-P model is to determine a preference for the NB-1 or NB-2 model (Hilbe, 2014) (since the NB-1 and NB-2 models are not nested, a comparisons of -2LL is not appropriate). Comparisons of the two models can be evaluated with a likelihood ratio test by extracting the -2 Log Likelihood values from the fitstatistics tables. PROC COUNTREG offers one approach to extract the log-likelihoods for both the NB-1 and NB-2 models. Comparisons with the -2 LL value from NLMIXED for the NB-P model are demonstrated in the supplementary SAS code. Simulations indicate a comparison of the log-likelihoods is very sensitive to detect even small departures from the NB-1 or NB-2 models.

With count data from applied projects the generalized negative binomial model (NB-P) may not converge when run in PROC NLMIXED, especially if they have small sample sizes or when the NB-1 or NB-2 models do not fit well. The usefulness of the NB-P as a stand-alone model has potential, but is yet to be explored.

NB-H: HETEROGENEOUS NEGATIVE BINOMIAL REGRESSION

The heterogeneous model is not currently available in SAS procedures. This approach extends the applications of negative binomial models discussed thus far (in particular, NB-1 & NB-2) by allowing the ancillary dispersion parameter phi to be estimated with one or more predictor variables in order to study the source of the overdispersion. The additional code includes initial parameter estimates in the PARMS statement for a second linear predictor, etaD, to estimate the dispersion parameter (rather than enter it on the PARMS statement) and its inverse link (for data with overdispersion, a log link is usually chosen):

```
PROC NLMIXED DATA=indat(rename=(rsp=y));
PARMS  b0 .1  b1 .1  b2 .1
      bd0 .1 bd1 .1 ;  * initial coef estimates for linear predictor of dispersion;

 * linear predictor for the mean;
 eta = b0 + b1*(group='A') + b2*x;
  mu = EXP(eta);

 * linear predictor for the dispersion parameter;
 etaD = bd0 + bd1*(group='A'); * dispersion depends on group;
  phi = EXP(etaD);             * the inverse EXP function assures phi > 0;

 * enter the log-likelihood for either NB-1 or NB-2 ;

 * NB-1; lglk = y*log(phi) - (y+(1/(phi/mu)))*log(1+phi)
               + lgamma(y+(1/(phi/mu))) - lgamma(1/(phi/mu)) - lgamma(y+1);

 * NB-2; lglk = y*log(phi*mu) - (y+(1/phi))*log(1+(phi*mu))
               + lgamma(y+(1/phi)) - lgamma(1/phi) - lgamma(y+1);
```

The heterogeneous negative binomial model is a valuable tool to differentiate sources of variation influencing the parameter estimates and also to assess sources influencing overdispersion in count data.

P-IG: POISSON – INVERSE GAUSSIAN

The negative binomial distribution is derived as a mixture of the Poisson and gamma distributions; thus, the model defined above as NB-2 could also be called the Poisson-inverted gamma distribution. Applying the inverted Gaussian distribution for the mean of the Poisson distribution results in the Poisson-inverted Gaussian (P-IG) model. This model is especially relevant to work with extremely over-dispersed count data, well beyond the situations the negative binomial model (NB-2) can handle. The P-IG model should especially be considered whenever the count data have a distribution with an initial peak at low values (say, with many counts less than 5) and then tapers off with a smaller number of much larger values resulting in a highly skewed distribution that is greatly Poisson over-dispersed. Unlike the negative binomial distribution, the P-IG probability density function does not have a concise mathematical formula (see Zha, 2016, p. 21). The formula includes a modified Bessel function of the third kind (currently not available in SAS; Wicklin, 2015) and for that reason has not be widely implemented in statistical software. Another derivation of the pdf is shown in Guo and Trivedi (2002, p. 68) which has a summation in the equation that can be programmed in NLMIXED; the gamma function in the numerator gives a computational error (i.e., missing value) for values of the response, y, greater than 76 (i.e., missing values result for `(y+i) GE 172` in the gamma function where i ranges from 0 to y-1).

However, the pdf for the Poisson-Inverse Gaussian also has a recursive form that computes the probability of a given y sequentially using the probabilities from 0 to y (Zha, 2016, p. 23 and Dean, 1989, p. 173).

$$P(Y=0) = \exp\left(\tau^{-1} * \left[1 - \left(1 + (2\tau\mu) \right)^{1/2} \right] \right)$$

$$P(Y=1) = \left[\mu * \left(1 + (2\tau\mu) \right)^{-1/2} \right] * Prob(Y=0)$$

$$P(Y=y) = \left[(2\tau\mu) / (1 + (2\tau\mu)) \right] * [1 - (3/(2y))] * Prob(Y=y-1)$$
$$+ \left[\mu^2 / (1+(2\tau\mu)) \right] * \left[1 / (y*(y-1)) \right] * P(Y=y-2) \quad y=2,3,4..$$

where τ is the dispersion parameter. Probability calculations with these equations match the results from R using the dPIG function from the gamlss package, as illustrated in the supplementary file of SAS code.

Also note that as values of τ approach 0 from the right, the P-IG distribution approaches the Poisson distribution. Estimation of regression coefficients for the P-IG can be achieved with these formulas programmed in the NLMIXED procedure by storing the individual probabilities in an ARRAY. The dimension of the ARRAY is defined by the maximum value of the response data +1 saved in a macro variable.

```
%LET maxY = 125; * enter the maximum value of the response variable;

PROC NLMIXED DATA =indat (rename=( <rsp> =y));
 < enter statements for PARMS (tau as dispersion), eta, and mu (LOG link) >;
ARRAY prby(%EVAL(&maxY +1) ) py0 - py&maxY; * ARRAY dimension is max value of y +1;
py0 = EXP( (1/tau)*(1 - SQRT(1 + (2*tau*mu))) );
IF y = 0 then py = py0;
py1 = py0 * mu * (1/SQRT(1 + (2*tau*mu))) ;
IF y EQ 1 then py = py1;
IF y GE 2 then
DO;
  DO k = 2 to y ;
    prby{k+1} = ((2*tau*mu/(1+ (2*tau*mu))) * (1 - (3/(2*k))) * prby{k})
              + ((mu**2)/(1 + (2*tau*mu))) * (1/(k*(k-1))) * prby{k-1};
  END;
  py = prby{k};
END;
py = max(min(py,1-1E-9),1E-9);
lglk = log(py);
MODEL y ~ general(lglk);
RUN;
```

The estimation process requires a DO loop to calculate the probability of the value of the response given the mean computed with the coefficients and the dispersion parameter, in this case storing probabilities of all values of y in an array for each record as the data is processed (mu changes with the values of the respective covariates for each row). It first computes Prob(Y=0) and then Prob(Y=1). For probabilities of the response y greater than or equal to 2 a DO loop is utilized where the index stops at the value of y, each iteration accessing the previous two probability calculations. The need for convergence of the COM model is not present here, since the number of required iterations is finite. The main requirement is the upper bound for the ARRAY needs to be 1 more than the largest value of the response (as y starts at 0). The upper bound can be found externally and entered into a macro variable, where the max value is determined before NLMIXED is run. Also, realize that computing the probability in this manner before computing the log likelihood in this manner is generally not recommended, as the logs of the individual components of the pdf are computed separately from each other and then added, as demonstrated in the lglk equations for other distributions.

Initial parameter estimates for this model that are close to the solution may be necessary for this model to produce results; that is, a computing error reported in the log window or lack of convergence is possible. The Poisson or the generalized Poisson (GPR-1) models may provide close initial estimates for the parameters. The PARMS statement is able to read parameter names and their initial values from an external SAS data set (an example for the P-IG model is provided in the accompany SAS code with initial estimates provided by a Poisson model produced with PROC GENMOD and read into NLMIXED with a data set). The ODS OUTPUT parameter estimates file for NLMIXED provides an example of the necessary contents and structure of this input data set for the PARMS statement; however, with GENMOD, the addition of the NLMIXED parameter names is necessary, which if chosen as b0, b1, b2, and so forth, is relatively easy to add enter in a DATA step. Be aware the P-IG model works best with data having extreme overdispersion and may not converge if the dispersion is relatively small. The P-IG model is also available in R with the gamlss package which can be run with PROC IML within a SAS program as shown in the supplementary SAS file.

EXAMPLES

Applications of most of these distributions with real and synthetic data sets are demonstrated in the supplementary file of SAS code accompanying this document. Analysis of count data for the various distributions are based on the examples demonstrated with Stata or R in chapters 5, 6, and 8 of "Modeling Count Data" (Hilbe, 2014) and chapter 10 "Negative Binomial Regression Models" (Hilbe, 2011). Data sets from other resources are also available to test these models (e.g, data sets are available online for most of the examples illustrated in "Regression Analysis of Count Data" 2nd ed. by Cameron and Trevidi, 2013).

VARIANCE FUNCTIONS

An important aspect of analysis is to have aids to interpret which count data model works better with a given data. Among them are residual analysis to determine goodness of fit and comparisons of various information criteria with other of models.

The variance functions for the distributions described above are presented in the table below, which indicates a helpful guide to their utility with a given data set (for each model the mean is indicated with mu and the dispersion parameter is called phi):

Distribution	Expected Value	Variance Function (phi is distinct for each model)	Comments
Poisson	mu	mu	Equality of mean and variance
COM Poisson (param=mu)	mu	see SAS code	Equation for mean and variance involves a calculation with a DO loop similar to lglk
QL Poisson	mu	mu * cnst	cnst is estimated from model output
NB-1	mu	mu + phi*mu	The "linear" NB model
NB-2	mu	$mu + phi*mu^2$	The most commonly applied NB model
GEOM	mu	$mu + mu^2$	NB-2 model with phi=1
NB-P	mu	$mu + phi*mu^P$	Estimate 3 parameters: mu, phi, and P
NB-H	mu	$mu + (phi*v)mu^2$	Dispersion parameter v varies by obs
GPS-1	mu/(1-phi)	$mu / (1 - phi)^3$	Unrestricted GPR, under and over dispersion
GPS with PROC FMM	mu	$mu / (1 - phi)^2$	phi is restricted to be positive (i.e., works with over-dispersion only) – see SAS code file
GPS-2	mu	$mu * (1 + phi*mu)^2$	Restricted GPR, under and over dispersion
P-IG	mu	$mu + phi*mu^3$ * see comments	* The re-parameterization of P-IG by Dean (1989) used in the examples has the variance function $mu*(1+phi*mu)^2$

Table 1. Means and Variance functions for Various Count Data Distributions

Another form of the linear variance function for the NB-1 model looks much like the quasi-likelihood Poisson model:

```
QL Poisson: Var = mu* cnst
       NB1: Var = mu*(1+phi)
```

The difference in the two models is the QL Poisson variance has the mean multiplied by a constant, cnst, determined from the output, whereas the variance function for the NB-1 model contains a dispersion parameter phi which is estimated with maximum likelihood from the data. These two approaches are essentially two different models.

This geometric distribution has a variance function Var(Y)=mu+mu^2 and thus, like the Poisson, depends only on the mean, yet accounts for a relatively large amount of overdispersion in the data.

Means can be computed for all these models with PROC NLMIXED by utilizing the PREDICT statement:

```
PREDICT mu OUT=prd(KEEP= < variables in the model and "pred" RENAME=(pred=mu)  > ;
```

A file with the value of the dispersion parameter for each row (or saved from the parameter estimates table) can also be produced:

```
PREDICT phi OUT=dsp(KEEP=pred RENAME=(PRED=phi) );
```

The output file(s) can be processed to compute Pearson residuals, and with some distributions deviance residuals. Since the data produced with both PREDICT statements has the same row order in both files, residuals and the resulting chi-square statistic can be computed in a DATA step:

```
%LET df = %EVAL(nobs - nparams); * enter values for degrees of freedom;

DATA rsd(drop=chisq)
    gfit(keep=chisq);
SET mu END=eof;
SET dsp;  * dispersion parameter phi, saved in an ODS output file;
Reschi = (y - mu) / SQRT ( < variance > ); * enter formula for variance from table;
chi2 + (reschi**2);
OUTPUT rsd;
IF eof THEN
  DO; chisq = chi2/&df.; OUTPUT gfit; END;
RUN;
```

One exception to the computation of chi-square results occurs with the GPS-1 model with NLMIXED. The expected value of the response y is a function of both the mean and dispersion parameter:

```
expY = mu/(1-phi);
varY = mu /((1-phi)**3);
reschi = (y - expY) / SQRT ( varY );
chi2 + (reschi**2);
```

The mean and variance, the COM Poisson model is treated somewhat differently than the other types of models due to the looping nature of the computations. Approximation formulas for the mean and variance are available (Sellers, 2010, Morris 2017). They produce reasonable estimates when certain conditions exist. However, accurate means and variances can easily be computed with formulas involving DO loop calculations, similar to the computation of the log-likelihood (the variance formula is provided in the PROC COUNTREG documentation) and can easily be computed in a DATA step for specified values of the explanatory data along with a transposed data file containing the model coefficients estimated from PROC NLMIXED. Results match the mean and variance from the OUTPUT statement of PROC COUNTREG. An example of how to compute the mean, variance, and dispersion of the observations is given in the supplementary SAS code file which also includes equations for residuals and goodness of fit summaries.

PROC NLMIXED produces fit statistics (AIC, AICC, BIC) which can also be compared (smaller is better) across these models. To determine if the dispersion parameter results in a significantly better fit, a chisquare test can be devised that compares the drop in log-likelihood from a Poisson model with the distribution of interest. See SAS code for an example with the generalized Poisson model.

CONCLUSION

This paper provides an overview of methods for parameter estimation of several count data statistical models that are either not well known or not currently available with SAS/STAT or SAS/ETS procedures. The objective is to demonstrate how these models can be estimated with the NLMIXED procedure. The specific data situations where each model can be applied, how to evaluate the output such as model fit, and interpretation of coefficients of these models need further consideration. The dispersion parameter in a count data model is an essential aid to interpreting variability of the data.

REFERENCES

Cameron, A. Trivedi, P. (2013) *Regression Analysis of Count Data*, 2nd ed. Cambridge University Press, New York.

Chou, Nan-Ting Chou and Steenhard, David, Flexible Count Data Regression Models Using SAS®. PROC NLMIXED SAS Global Forum, Paper 250-2009

Consul, P. (1989) *Generalized Poisson Distribution*. Marcel Dekker, Inc. New York and Basel.

Consul, P. and G. Jain (1973). A generalization of the Poisson distribution. Technometrics 15:791-799.

Consul, P. and F. Famoye (1992). Generalized Poisson regression model. Communications in Statistics – Theory and Methods 21: 89-109.

Dean, C., J. F. lawless, and G. E. Willmot, (1989). A mixed Poisson-inverse Gaussian regression model. The Canadian Journal of Statistics, vol. 17. No 2, pp. 171-181.

Famoye, F. (1993) Restricted Generalized Poisson Regression Model. Commun. Statist. – Theory Meth., 22(5), 1335-1354.

Famoye, f. and K. Sing (2006). Zero-truncated generalized Poisson regression model with an application to domestic violence. Journal of Data Science 4: 117-130.

Greene, W. H. (2008). Functional forms for the negative binomial model for count data. Economics Letters 99, 585-590.

Guikema, S and Goffelt, J. (2008). A Flexible Count Data Regression Model for Risk Analysis. Risk Analysis, Vol. 28, No. 1.

Guo, Jie Q. and Pravin K. Trivedi (2002) Flexible Parmetric Models for Long-tailed Patent Count Distributions. Oxford Bulletin of Economics and Statistics, 64, 63-82.

Harris, T., Z. Yang, and J. W. Hardin. (2012). Modeling Underdispersed Count Data with Generalized Poisson Regression. Stata Journal 12 (no. 4) 736-747.

High, R. and ElRayes, W. (2017) Fitting Statistical Models with the NLMIXED and MCMC Procedures. Proceedings of the SAS Global 2017 Conference, Paper 902-2017.

Hilbe, J. (2011). *Negative Binomial Regression*, 2nd ed. Cambridge University Press, New York

Hilbe, J. (2014). *Modeling Count Data.* Cambridge University Press, New York

Joe, H., and Zhu, R. (2005). Generalized Poisson Distribution: The Property of Mixture of Poisson and Comparison with Negative Binomial Distribution. *Biometrical Journal* 47:219–229.

McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models*, 2nd ed., London: Chapman & Hall.

Morris, DS, Sellers, KF, Menger, A., (2017). Fitting a Flexible Model for Longitudinal Count Data Using the NLMIXED Procedure. SAS Global Forum, Paper 202-2017.

Neter, J., Kutner, M. H., Nachtsheim, C. J., and Wasserman, W. (1996). *Applied Linear Statistical Models*. Fourth Edition, Chicago: Irwin.

Sellers, K. and Shmuelli, G. (2010). A Flexible Regression Model for Count Data. Annals of Applied Statistics, 4:943-961.

Wicklin, Rick, “Direct me to the nuclear Bessels! “DO Loop” article accessed Jan. 8, 2018, https://blogs.sas.com/content/iml/2015/05/20/nuclear-bessels.html.

Zha, Liteng, Dominique Lord, and Yajie Zou. (2016) The Poisson inverse Gaussian (PIG) generalized linear regression model for analyzing motor vehicle crash data. Journal of Transportation Safety and Security, Vol. 8, No. 1, 18-35.

Your comments and questions are valued and encouraged. Contact the author at:

Robin High
Department of Biostatistics
College of Public Health
University of Nebraska Medical Center
984375 Nebraska Medical Center
Omaha, NE 68198-4375
email: rhigh@unmc.edu

APPENDIX

RUN R CODE WITH SAS

In order to run R code with PROC IML in SAS 9.4, first complete these steps.

1. Find the directory where sasv9.cfg is stored. On my Windows computer it is:

   C:\Program Files\SASHome\SASFoundation\9.4\nls\en

2. Open sasv9.cfg with notepad.

3. Find the line:

   /* DO NOT EDIT BELOW THIS LINE - INSTALL Application edits below this line */

4. At the beginning of a blank line somewhere below it enter:

   -RLANG

5. Save the sasv9.cfg file

6. Start the SAS program and run R code with PROC IML as illustrated below

When this setup process has been accomplished, running R code within PROC IML is convenient, including passing SAS data sets to R and converting R data into SAS datasets:

```
PROC IML;
run ExportDataSetToR("cdata", "cdata"); * SAS data set converted to a R file;
submit / R;
 < enter R code here just as in the R editor >
endsubmit;
run ImportDataSetFromR ("fitted", "fitted"); * place data from R into a SAS data set;
QUIT;
```