MWSUG 2017 - Paper AA05

Unconventional Statistical Models with the NLMIXED Procedure

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

SAS®/STAT and SAS/ETS software have several procedures which estimate parameters from generalized linear models for a variety of continuous and discrete distributions. The GENMOD, COUNTREG, GLIMMIX, LIFEREG, and FMM procedures, among others, offer a limited range of unconventional types of analysis options, including those for zero-inflated, truncated, and censored data. The COUNTREG procedure includes the Conway-Maxwell Poisson distribution and the negative binomial with two variance functions. The FMM procedure includes the generalized Poisson distribution as well as the ability to work with several truncated and zero-inflated distributions for both discrete and continuous data. The NLMIXED procedure can be utilized to duplicate their results in order to gain insight into the complex computational details. The capability to enter complex programming statements into NLMIXED and to a limited extent, the GLIMMIX procedure, can be expanded to work with data from even more unconventional situations.

INTRODUCTION

Parameter estimation from several "unconventional" statistical models for count data will be illustrated in this paper. Most of them are derived from the "conventional" models commonly applied to data analysis situations. The models illustrated here are called unconventional since they are not options available on the MODEL statements of most linear models procedures. A few of these models are available yet are not applied in data analyses as frequently as they could be, and their existence is likely unknown. The objective of this paper is to introduce parameter estimation methods for these models as a starting point for consideration how and when they may be applied in common data analysis situations. If not currently available in SAS procedures they can be estimated by accessing statistical packages in R through the IML procedure; however, when computationally possible most of them can be programmed with statements written in the NLMIXED procedure. Also, as illustrated with a few examples in the Appendix, the GLIMMIX procedure can be programmed for many distributions with one linear predictor for the mean.

THE NLMIXED PROCEDURE

This paper is an extension of the contents of the SGF paper, "Fitting Statistical Models with the NLMIXED and MCMC Procedures" (High, ElRayes, 2017) which illustrates how to write the four basic types of statements the NLMIXED procedure needs to estimate parameters for statistical models: the initial parameter estimates (PARMS), the linear predictor eta, the mean (with the inverse link), and log-likelihood equations for several parametric distributions.

The NLMIXED procedure requires these types of programming statements for most statistical models. The motivating data set for the statistical models for many of the examples contains two explanatory variables: one categorical variable (group coded as A/B) and a continuous variable x. The linear predictor eta is written such that level B of the group variable is the reference category (corresponding to reference category coding). The required statements in the NLMIXED procedure for all examples is:

```
PROC NLMIXED DATA = indata(rename=(response = y)) ;
PARMS b0 2  b1 1 b2 .1 phi .5;
eta = b0 + b1*(group='A') + b2*x;
mu = exp(eta); * inverse link;
lglk =  < enter log-likelihood equation with parameters mu and phi > ;
MODEL y ~ general( lglk ) ;
TITLE "NLMIXED: model type";
RUN;
```

In the examples for PROC NLMIXED demonstrated here the response variable is assumed to have the name y. Other names for the response should be changed to y with the rename=() option attached to the input data set name. The NLMIXED MODEL statement and especially the log-likelihood equation all contain the name y, which may appear multiple times in the latter, and thus does not need to be edited.

UNCONVENTIONAL MODELS BASED ON THE POISSON DISTRIBUTION

With response data collected as non-negative integer counts, the Poisson distribution is often considered as an analysis option with the GENMOD and GLIMMIX procedures. However, the restriction that the mean equals the variance is frequently violated due to under- or over-dispersion in the data. The negative binomial distribution is the common choice when over-dispersion is present; however, it cannot handle under-dispersion, which is not encountered nearly as often, yet alternatives are available when it does. For these situations, several "unconventional" extensions to the Poisson distribution may be applied.

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 Poisson is a special case of both types. The COM distribution models data that exhibit either under- or over-dispersion. Both types of parameterizations of the COM distribution can be programmed in the NLMIXED procedure. The "lambda" parameterization is described by Morris (2017) as it was derived by Sellers (2010). The "mu" parameterization is the default in COUNTREG with formulas described in Guikema (2008). The parameter estimates from the mu parameterization are easily compared with the estimated coefficients from a Poisson model with the GENMOD and GLIMMIX procedures. A simple illustration for under-dispersion is encountered with the air-freight breakage data 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;
```

Both criteria for assessing goodness of fit (Deviance=0.227 and Pearson Chi-Square=0.222) indicate under-dispersion is present, since for the Poisson distribution 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 model can be run with the COUNTREG procedure:

```
PROC COUNTREG DATA=freight;
MODEL broken = transfers / dist=compoisson parameter=mu ;
OUTPUT OUT=predicted pred=pred nu=nu dispersion=dispr variance=vrc;
RUN;
```

The estimated parameter values 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 both models, however, the standard errors for the COM model are considerably smaller due to underdispersion (implied by the negative value for _lnNu= -1.757). The COM model can also be programmed with NLMIXED statements:

```
PROC NLMIXED DATA=freight;
PARMS b0 -2 b1 .1 nu 3 ;
BOUNDS nu > 0;
 eta = b0 + b1*transfers;
 mu = EXP(eta);
```

```
S_ = 1;
DO i_ =1 to 45; *an upper limit of 45 since the data are clearly under-dispersed;
  f_ = 1;
  DO n_ = 1 to i_;
    f_ = f_  * ((mu/n_)**nu) ;
  END;
  S_ = S_ + f_;
END;
lglk = nu *( y*LOG(mu) - lgamma(y+1)) - log(S_);
MODEL y ~ general(lglk);
ESTIMATE '-log(nu)' -log(nu);
TITLE 'NLMIXED: COM Poisson with parameter=mu';
RUN;
```

In the NLMIXED code, mu is the mean and nu the dispersion parameter, the value of nu=5.78 (transformed to `-log(nu)=-1.757` in COUNTREG) with the freight data where values of nu greater than 1 indicate under-dispersion. The DO loop with 45 iterations for each row of data indicate the intensive computational requirements for this type of analysis. As the index `i_` increases, the computations for `S_` are based on ratios of two extremely large numbers. Although the equation indicates the value `f_` will eventually converge, certain 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), the convergence is faster, often the upper limit for `i_` can be less than 50. With counts having relatively large means (e.g., greater than 15) or with count data having a large amount of over-dispersion, the upper limit for `i_` must be large, perhaps much larger than 100 as presented in Morris (2017). Note that the convergence of this log-likelihood can be observed by running the computations in a DATA step for a few observations (place an OUTPUT statement after `S_ = S_ + f_ ;` and change PARMS to RETAIN).

Approximate formulas are also presented for the mean and variance (Sellers, 2010, Morris 2017). They are relevant when certain, conditions exist which may often be unrealistic. However, means and variances with the more precise formulas (provided in the PROC COUNTREG documentation) can be computed in a DATA step for combinations of the explanatory data along with the coefficients estimated from PROC NLMIXED which will match the results from the OUTPUT statement of PROC COUNTREG. Here is an example of this process:

```
* Transpose the parameter estimates from NLMIXED in the file called prmCOM produced
  with ODS having the coefficient names of b0 (intercept) and b1 (slope);

PROC TRANSPOSE DATA=prmCOM out=tprmCOM(DROP=_name_ _Label_) ;
VAR estimate;
ID parameter;
RUN;

DATA xdat;
DO transfers=0 to 3; OUTPUT; END; * Enter unique values of the explanatory data ;
RUN;                              * in an external data set ;

DATA mn_vr2;
SET xdat;
IF _n_ = 1 then SET tprmCOM; * read estimated coefficients;
KEEP transfers mn vr disprsn;
eta = b0 + b1*transfers;      * enter the linear predictor from NLMIXED;
mu = exp(eta);                * inverse link from NLMIXED;
S_ = 1;
do i_ = 1 to 45; * choose upper bound large enough so that the ratio f_ converges;
f_ = 1;
 do n_ = 1 to i_;
  f_ = f_ * ((mu/n_)**nu) ;
end;
S_ = S_ + f_;
end;
```

```
* an upper bound of 35 is chosen so the CDF of the distribution is very close to 1;
sm  = 0; DO j = 0 to 35; sm  = sm  + (( j    *(mu**(j*nu))) / (FACT(j)**nu) ); END;
sm2 = 0; DO j = 0 to 35; sm2 = sm2 + (((j**2)*(mu**(j*nu))) / (FACT(j)**nu) ); EMD;
mn = (1/S_)*sm ;
vr = (1/S_)*sm2 - (mn**2) ;
disprsn = vr/mn;
OUTPUT;
RUN;
```

The programming statements to estimate the COM model parameters with PROC GLIMMIX is given in the Appendix.

COM: MODEL THE MEAN AND THE DISPERSION

The COUNTREG procedure offers both estimate of the mean and the dispersion parameter (heterogenous model) with the DISPMODEL statement:

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

The main difference for the NLMIXED code is to enter the linear predictor for nu and its inverse link (the log link here) with a negative sign for the linear predictor of nu to match the results of COUNTREG.

```
PROC NLMIXED DATA=indat;
PARMS b0 3 b1 .1 bn0 .1 bn1 .1;
  eta = b0 + b1*(group=0);     mu = EXP(eta);
etaNu = bn0 + bn1*(group=0);   nu = EXP(-etaNu);
  < enter code for lglk from COM Poisson, mu parameterization >
lglk = nu *( y*LOG(mu) - lgamma(y+1)) - log(S_);
MODEL y ~ general(lglk);
ESTIMATE 'group 0 dispersion' exp(-(bn0+bn1));
ESTIMATE 'group 1 dispersion' exp(-bn0);
RUN;
```

GPR: GENERALIZED POISSON REGRESSION

The generalized Poisson distribution was developed by Consul and Jain (1973). Consul and Famoye (1992) also provide an overview of this model and its derivation. The conventional Poisson model is a special case of this distribution. Whereas the negative binomial heterogeneity parameter ($k > 0$) is based on the gamma distribution, the generalized Poisson's heterogeneity parameter is based on the lognormal distribution allowing it to be applied with data having either under- or over dispersion. However, there are limitations to the amount of under-dispersion present for which the model works. Since the range of the response variable depends on the dispersion parameter k, it violates one of the standard conditions for consistency and asymptotic normality of maximum likelihood estimation (Cameron and Trivedi, p. 171).

The GPR model with the parameterization of Famoye and Singh (2006) can be programmed with PROC NLMIXED:

```
PROC NLMIXED DATA= indat ;
PARMS b0 .75  b1 .1  b2 -.1 k .5 ;
eta = b0 + b1*(group='A') + b2*x;
mu = exp(eta);
lglk = y*log(mu /(1+ (k*mu))) + (y-1)*log(1+(k*y))
      + ((-mu*(1+(k*y))) / (1 + (k*mu))) - lgamma(y+1);
MODEL y ~ general( lglk ) ;
RUN;
```

See Hilbe (2011, p. 340) for the density function for this particular probability distribution (there are errors in the log-likelihood equations (10.65 and 10.66) which has been corrected in the NLMIXED code).

Examples of how to implement the parameterization of the generalized Poisson model (Joe and Zhu, 2005) with R, PROC FMM, and PROC GLIMMIX programming statements are given in the Appendix.

P-IG: POISSON – INVERTED GAUSSIAN

The negative binomial distribution (to be discussed later) is derived as a mixture of a Poisson and gamma distributions; thus, it could be called the Poisson-inverted gamma distribution. Applying instead the inverted Gaussian distribution to the mean results in the Poisson-inverted Gaussian (P-IG) model. This model is especially suited for extremely over-dispersed count data, even beyond the situations for which the negative binomial distribution is appropriate. Unlike the negative binomial distribution, the P-IG probability density function does not have a concise mathematical formula. The formula involves a modified Bessel function of the third kind and for that reason has not be widely implemented in statistical software. Though it could conceptually be programmed in PROC NLMIXED, this distribution is currently available in R with the gamlss package which can be run with PROC IML within a SAS program (see Appendix for instructions of the initial steps needed in order to run R code with SAS). Data stored in a SAS dataset (e.g., with the name cdata) can access the P-IG routine via PROC IML:

```
PROC IML;
run ExportDataSetToR("WORK.cdata", "cdata"); * SAS data set converted to a R file;
submit / R;
library(gamlss)
pivg <- gamlss(y ~ x1 + x2, data=cdata, family=PIG)
summary(pivg)
# Compute fitted values
  pivg.fitted = predict(pivg, newdata=cdata)
  fitted <- pivg.fitted
endsubmit;
run ImportDataSetFromR ("fitted", "fitted"); * copy data to a SAS data set;
QUIT;

PROC PRINT DATA=fitted;
TITLE "Fitted values";
RUN;
```

THE NEGATIVE BINOMIAL DISTRIBUTION

One of several discrete and continuous distributions that can be evaluated with PROC NLMIXED is the negative binomial, selected as an option with the MODEL statement:

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

The two parameters n and p define 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 typically illustrated with the parameter n as an integer, it can have any positive real number. The parameterization with this 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 another parameterization:

```
PROC NLMIXED DATA = indat;
PARMS b0 1 b1 .5 b2 .5 k .1;
eta = b0 + b1*(group='A') + b2*x;
mu = exp(eta);
p = 1 /(1 + (mu*k));
MODEL y ~ NEGBIN(1/k, p);
TITLE 'NLMIXED: Negative Binomial with MODEL statement';
RUN;
```

Computations of the negative binomial regression model with NLMIXED are also equivalent with PROCs GENMOD, FMM, and GLIMMIX by directly entering the log-likelihood equation for the negative binomial:

```
PROC NLMIXED DATA = indat ;
PARMS b0 1 b1 .5 b2 .5 k .1;
eta = b0 + b1*(group='A') + b2*x;
mu = exp(eta);
lglk = y*log(k*mu) - (y+(1/k))*log(1+(k*mu))
      + lgamma(y+(1/k)) - lgamma(1/k) - lgamma(y+1) ;
MODEL y ~ general(lglk);
TITLE 'NLMIXED: Negative Binomial with log-likelihood';
RUN;
```

where mu is the mean (a function of the parameter estimates in the linear predictor) and k the dispersion parameter as it appears in the pdf for the negative binomial distribution in GLIMMIX documentation (except for the normal distribution, the variance of an observation in a generalized linear model is a function of k and mu). With the log link [the inverse link is EXP(eta)], the Poisson distribution is a limiting case of the negative binomial log-likelihood as k approaches 0 from the right (Hilbe, p. 221); with k close to 0, results from the Poisson and negative binomial distribution with the log link are nearly the same.

UNCONVENTIONAL MODELS FOR THE NEGATIVE BINOMIAL DISTRIBUTION

The following models are presented as extensions or alternatives to the conventional negative binomial distribution. They are described in greater detail in chapter 10 of "Negative Binomial Regression," by Joseph Hilbe (2011). In this book they are illustrated with R, STATA, and LIMDEP programs. SAS/STAT software does not currently include most of them as modeling options, especially in the GENMOD and GLIMMIX procedures. However, many of them can be computed with SAS through programming statements entered into PROC NLMIXED (a few can also be programmed in GLIMMIX).

GEOM: GEOMETRIC REGRESSION MODEL

The geometric distribution is a special case of the negative binomial distribution with k = 1/phi = 1:

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

Let k=1 in the equation for the negative binomial log-likelihood:

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

The lgamma functions are no longer needed sin lgamma(1/1) = 0 and lgamma(y+(1/1)) = lgamma(y+1). Thus, when a negative binomial model has k close to 1, the geometric distribution may be an alternative with the following log likelihood equation:

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

Substitute this log-likelihood for the geometric distribution in the code for the negative binomial distribution in PROC NLMIXED. The choice between the negative binomial or geometric distributions can be evaluated by comparing the fit statistics (e.g., AICC or BIC) since the geometric is a special case of the negative binomial. Also, an ESTIMATE statement in PROC NLMIXED with the negative binomial model will test whether k has a significant difference from 1:

```
ESTIMATE "k = 1?" k - 1;
```

CGM: CANONICAL GEOMETRIC REGRESSION MODEL

As with the negative binomial, the log link is usually applied with the geometric distribution. The canonical link for the geometric distribution is:

```
Link: eta = -LOG( (1/mu ) + 1)
```

The canonical inverse link and log-likelihood equation for the geometric distribution are then entered into NLMIXED with these statements:

```
    mu = 1/(EXP(-eta) - 1);
lglk = y*LOG(mu) - (y+1)*LOG(1+mu);
```

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

The variance function for the negative binomial (p=2) is $mu(1+k*mu) = mu + k*mu^2$. The linear negative binomial (p=1) has variance function $mu*(1+k) = mu + k*mu^1$. The exponent in the second term of the variance function defines the value of p. Note the similarity in the variance functions between the quasi-likelihood Poisson and NB1:

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

The difference in the two models is the QL Poisson is multiplied by a constant, phi, determined from the output, whereas the NB1 variance function contains the parameter k to be estimated with maximum likelihood. The result is two different models. PROC COUNTREG from SAS/ETS 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 NB1 Model is derived from the likelihood for the NB2 model with k replaced by k/mu (Cameron and Trevedi, 2013).

```
PROC NLMIXED DATA = indat;
PARMS b0 .1  b1 .1 b2 .1 k .1;
eta = b0 + b1*(group='A') + b2*x;
mu = exp(eta);
lglk = y*log(k) - (y+(1/(k/mu)))*log(1+k)
     + lgamma(y+(1/(k/mu))) - lgamma(1/(k/mu)) - lgamma(y+1) ;
MODEL y ~ GENERAL(lglk);
TITLE "Negative Binomial: p=1";
RUN;
```

NB-C: CANONICAL NEGATIVE BINOMIAL

Like the geometric regression model, the log link is usually the default choice for the negative binomial which allows one to model Poisson overdispersion. The canonical link for the negative binomial is described in McCullagh and Nelder's foundational text on generalized linear models (1989, pp. 373-374). It is an example of a model where the variance function contains a second parameter to be estimated. They perceived the difficulty in applying this type of model to data would be estimating this parameter, rather than entering it as a constant. For the negative binomial, the canonical link, where k is the dispersion parameter, is:

```
Link: eta = -LOG((1/(k*mu)) + 1);
```

The statements for mu with the inverse canonical link and the log-likelihood in terms of mu and k for PROC NLMIXED are:

```
   mu = 1 / (k*(EXP(-eta)-1)); * inverse canonical link contains k;
lglk = y*log(k*mu) - (y+(1/k))*log(1+(k*mu))
     + lgamma(y+(1/k)) - lgamma(1/k) - lgamma(y+1); * lglk is neg bin with p=2;
```

NB-H: HETEROGENEOUS NEGATIVE BINOMIAL REGRESSION

The heterogeneous model extends the three types of negative binomial models discussed thus far (NB1, NB2, and NB-C) by allowing the ancillary parameter k to be estimated from the data.

```
PARMS  b0 .1  b1 .1  b2 .1
       bk0 .1 bk1 .1 ;  * initial coef estimates for linear predictor and dispersion;

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

 * linear predictor for the dispersion parameter;
 etaK = bk0 + bk1*(group='A');  * dispersion depends on group;
    k = EXP(etaK);

  lglk = y*log(k*mu) - (y+(1/k))*log(1+(k*mu))
     + lgamma(y+(1/k)) - lgamma(1/k) - lgamma(y+1); * lglk is neg bin with p=2;
```

The heterogeneous negative binomial model is a valuable tool to assess sources of overdispersion not available in PROCs GLIMMIX or GENMOD. In particular, one can determine which factors contribute to the dispersion parameter.

NB-P: GENERALIZED NEGATIVE BINOMIAL

The generalized negative binomial model was developed to allow more flexibility in estimating the variance than is available with NB1 and NB2 models. The generalized formula allows the estimation of the exponent in the variance:

$$\begin{aligned} Var &= mu + k*mu^{Q} \\ &= mu*(1 + k*mu^{(Q-1)}) \end{aligned}$$

The generalized negative binomial model contains three parameters to be estimated: mu, k, and Q (from which P is derived). The log-likelihood equation entered in PROC NLMIXED is the revised equation 10.47 from Hilbe (2011, the eq. on the bottom of p. 324 is incorrect) found in the errata list from Jan. 12, 2012 (available on the book's website).

```
PROC NLMIXED DATA= indat;
PARMS b0 .8  b1 .1  b2 -.4 k 3 Q .2;
eta = b0 + b1*(group='A') + b2*x;
mu = exp(eta);
lglk =  ( (1/k)*(mu**Q) * log( ((1/k)*(mu**Q))/(((1/k)*(mu**Q)) + mu) ))
     + (y*log(1 - (((1/k)*(mu**Q)) / (((1/k)*(mu**Q)) + mu))) )
     + lgamma(y + ((1/k)*(mu**Q))) - lgamma((1/k)*(mu**Q)) - lgamma(y+1);
MODEL y ~ general(lglk) ;
ESTIMATE 'P'  2 + Q ;
RUN;
```

GWR: GENERALIZED WARING REGRESSION

The generalized Waring regression model is not well-known for count data even though its origins go back to Edward Waring (1736-1798) and other documents in the past century. Derived from the negative binomial distribution, it is yet another way to account for over-dispersion in a regression model that takes into account three sources of variation: exposure to risk (liability), differences due to individual characteristics (proneness), and pure chance (randomness). The difference from a negative binomial model is how the first two types of variation are treated as separate sources of over-dispersion observed in the model.

The model is briefly described and compared with the negative binomial distribution in Hilbe (2011, pp. 328-333). PROC NLMIXED estimates the parameters with the following statements:

```
PROC NLMIXED DATA= indat;
PARMS b0 2 b1 .1 b2 .1 k .5 rho 4;
 eta = b0 + b1*(group='A') + b2*x;
  mu = exp(eta);
  aa = (mu*k)/(rho-1);
lglk = lgamma(aa+y) - lgamma(aa) + lgamma(k+y) - lgamma(k)
      - lgamma(aa+k+rho+y) + lgamma(aa+rho) + lgamma(k+rho)
      - lgamma(rho) - lgamma(y+1);
MODEL y ~ general( lglk ) ;
RUN;
```

The package GWRM in R has been recently released and described by Vilchez-Lopez (2016) and contains features to fit the generalized Waring regression model parameters (with versions of R greater than 3.0.0). The R commands given here are different than illustrated in Hilbe, pp. 330-333. After installing it (along with the required supplementary items), PROC IML in SAS can run this count data model, in particular to produce estimates of proneness and liability for a categorical predictor as outputs:

```
PROC IML;
run ExportDataSetToR("WORK.indat", "indat");
submit / R;
library(GWRM)
bvls <- gw(y ~ group, data=indat)
summary(bvls)
prt <- partvar(bvls)
endsubmit;
run ImportDataSetFromR ("prt", "prt");
QUIT;

DATA prt2; SET prt; SET indat(keep=id group);

PROC MEANS DATA =prt2 n mean maxdec=5;
CLASS group;
VAR Prop_Variance_Components_Randomn Prop_Variance_Components_Liabili
    Prop_Variance_Components_Pronene;
run;
```

Examples of the use of the generalized Waring regression model are given in Vilchez-Lopez (2016). How to interpret this model and incorporate it into data analysis situations deserves additional attention. Of all the count data models illustrated here, it could be considered among the most important candidate model for future development in one or more of the SAS count data procedures. Hilbe even states that “it may well become a well-used model in coming years” (2011, p. 333).

TRUNCATION AND CENSORING

Data analysis may involve the truncation or censoring of observations, perhaps in unconventional ways. They are described as left and right referring to the lower or upper tails of the distribution. [Interval censoring is also a possibility but will not be discussed here.] With either type, the adjustment to the log-likelihood is based on the cumulative probability of the outcomes which define the values involved in the truncation or censoring; however, the adjustment applied to the pdf is different for each type, though the methods are basically the same for both discrete (counts) and continuous data. Chapter 12 of Hilbe (2011) describes fitting censored and truncated models for the Poisson and negative binomial regression models.

Truncation of a distribution occurs when values less than or equal to a minimum C1 (left) or values greater or equal than a maximum C2 (right) do not exist. For count data, a basic example of truncation is when 0 is not a viable outcome, such as number of days for a hospital stay. With left or right truncation, the adjustment is to divide the pdf by the sum of probabilities for the available data so that the cumulative probability of the truncated distribution is 1.

```
 Left truncation at C1: lglk = LOG(pdf / (Pr(y GT C1)) )
                              = LOG(pdf) – LOG(Pr(y GT C1))
Right truncation at C2: lglk = LOG(pdf / (Pr(y LT C2)) )
                               LOG(pdf) - LOG(Pr(y LT C2))
```

Censoring occurs when data less than or equal to a minimum C1 (left) or greater than or equal to a maximum C2 (right) exist but for whatever reason are not yet known, the subject is lost to the study with the last reported value, or cannot be measured, so the value recorded is the boundary point C1 or C2. A binary variable called cens is also included in the computations which equals 1 when the value is censored and equals 0 otherwise:

```
 Left censoring at C1: lglk = (cens=0)*LOG(pdf) + (cens=1)*LOG(Pr(y LE C1))
Right censoring at C2: lglk = (cens=0)*LOG(pdf) + (cens=1)*LOG(Pr(y GE C2))
```

Two types of distribution functions are required for these situations:

```
CDF = Cumulative Distribution Function = Pr(y LE C)
SDF = Survival Distribution Function   = Pr(y GT C)
```

For a given value of y from any discrete or continuous distribution, the result of adding these two functions is CDF(y) + SDF(y) = 1.

SAS also has functions to compute both the cumulative and the survival probabilities of many of the commonly known discrete and continuous distributions. When forming the log-likelihood equations to run with PROC NLMIXED there are also the log CDF and SDF functions:

```
LOGCDF = LOG Cumulative Distribution Function = LOG(Pr(y LE C))
LOGSDF = LOG Survival Distribution Function   = LOG(Pr(y GT C))
```

In general, writing log-likelihood equations for PROC NLMIXED with the older probability functions is not recommended; use the log of the components of the probability density or cumulative density functions whenever possible. However, the newer LOGCDF and LOGSDF functions are written with greater accuracy and in my experience usually give equivalent results to writing out the formulas, especially for count data. A few examples will illustrate how to apply these functions.

POISSON: LEFT TRUNCATION AT 0

For count data truncated at 0, PROC FMM will compute the truncated Poisson:

```
PROC FMM DATA =tpois;
CLASS group;
MODEL y = group / DIST=tpoisson link=log;
TITLE 'FMM: Zero Truncated Poisson';
RUN;
```

PROC FMM also has a 0 truncated negative binomial distribution that is called with `dist=tnegbin`. For a Poisson distribution with truncation of 0, the log-likelihood for PROC NLMIXED can be coded in two ways:

```
lglk = y*LOG(mu) - mu  - lgamma(y+1)
       - LOG(1 - EXP(-mu));        * based on the pdf subtract LOG(1-P(y EQ 0));
lglk = y*LOG(mu) - mu  - lgamma(y+1)
      - LOGSDF('Poisson', 0, mu);  * subtract LOG(PR(y GT 0)) ;
```

The 0 truncated negative binomial model can be evaluated with analogous equations.

POISSON: LEFT TRUNCATION AT CUTPOINT =3

For unconventional truncation values greater than 0 (either on the left or right), programming statements in PROC NLMIXED can be applied. The LOGSDF functions for either the Poisson or negative binomial distributions provide a convenient way to sum the probabilities (rather than write out the formula or

accumulate sums of probabilities in a DO loop). For example, when values of y less than or equal to a cutpoint of 3 do not exist (i.e., 4 is the minimum value collected), the Poisson log-likelihood becomes:

```
lglk = y*LOG(mu) - mu - lgamma(y+1)
       - LOGSDF('Poisson', 3, mu);  * subtract LOG(PROB(y GE 4));
```

NEGATIVE BINOMIAL: LEFT TRUNCATION AT CUTPOINT =3

The negative binomial model with truncation at C1=3 uses these formulas:

```
      pp = 1/(1 + (mu*k));
    lglk = (y*log(k*mu) - (y + (1/k))*LOG(1+(k*mu))
           + lgamma(y+(1/k)) - lgamma(y+1) - lgamma(1/k) )
           - LOGSDF('NEGBINOMIAL', 3, pp, 1/k); * subtract LOG(PROB(y GE 4));
```

NEGATIVE BINOMIAL: RIGHT TRUNCATION AT CUTPOINT=15

When data are truncated on the right, all values of the response greater than or equal to the cut-point C2 are omitted (as defined here). In this case divide the pdf from 0 to the C2-1 by the area of the distribution for the relevant range of possible values. For example with a right-truncated distribution at C2=15, only values from 0 to 14 are relevant; values greater than or equal to 15 do not exist. Since the cumulative distribution is applied, the log-likelihood equation for a right-truncated negative binomial distribution at C2=15 efficiently enters the LOGCDF function:

```
     pp = 1/(1 + (mu*k));
   lglk = (y*log(k*mu) - (y + (1/k))*log(1+ (k*mu) )
          + lgamma(y+(1/k)) - lgamma(y+1) - lgamma(1/k) )
          - LOGCDF('NEGBINOMIAL', 15-1, pp, 1/k);  * subtract LOG(PR(y LE 14));
```

NEGATIVE BINOMIAL: RIGHT CENSORING AT CUTPOINT=14

With right-censored data, values of the response greater than the cut-point are either the last known value or a value which cannot be measured beyond the cutpoint. In this case the cumulative distribution of the response greater than or equal to C2 (survival) becomes the component to adjust the log-likelihood equation. Since the log of the cumulative distribution is to be entered for a censored value, the log-likelihood equation for a right-censored negative binomial distribution at C2=14 uses the LOGSDF function:

```
     pp = 1/(1 + (mu*k));
   lglk = (cens=0)*((y*log(k*mu) - (y+(1/k))*log(1+(k*mu))
                  + lgamma(y+(1/k)) - lgamma(y+1) - lgamma(1/k)))
        + (cens=1)*(LOGSDF('NEGBINOMIAL', 14-1, pp, 1/k));
```

The censored value at C2 for the response entered into the LOGSDF function has 1 subtracted, C2=14-1, since the SDF functions compute Pr(Y > C2).

Modeling continuous data with truncated or censored observations proceeds in an analogous manner with LOGCDF and LOGSDF functions available for them as well. An example of the right-censored Weibull distribution coded with PROC NLMIXED is given in High and ElRayes (2017) One notable feature of the Weibull and Log-Logistic distributions are the closed form equations for both the CDF and SDF functions, which make writing a log-likelihood equation for them more straight-forward than using SAS probability functions.

CONCLUSION

The objective of this paper is to briefly present parameter estimation methods for 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 PROC NLMIXED or with R commands through PROC IML for a few of the examples. Computations of expected values and variances are not considered. The specific circumstances where each model could be applied, how to evaluate the output such as model fit and interpretation of coefficients of these models also need further consideration.

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 Model Using SAS® PROC NLMIXED SAS Global Forum, Paper 250-2009

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.

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.

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

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

Hilbe, J. (2011) *Negative Binomial Regression*, 2nd ed. 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*, Second Edition, 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.

Vilchez-Lopez, S, Saez-Castillo, AJ, and Olmo-Jimenz, MJ, (2016) GWRM: An R Package for Identifying Sources of Variation in Oversdispersed Count Data, PLoS ONE 11(12): e0167570, doi:10.1371/journal.pone.01675570.

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

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

APPENDIX

FITTING STATISTICAL MODELS WITH GLIMMIX PROGRAMMING STATEMENTS

A feature of GLIMMIX is that unconventional statistical models with one linear predictor can also be run by omitting the dist=<> option on the MODEL statement and entering the log likelihood equation and variance functions directly into GLIMMIX with programming statements. Unlike PROC NLMIXED, code entered into GLIMMIX does not need to specify initial parameter values nor is it necessary to write out the linear predictor eta. Also, random effects can be incorporated in the usual direct manner with a RANDOM statement. Terms for the variance, mean, linear predictor, and dispersion all have special internally coded names that begin and end with the underscore, e.g., _variance_, _mu_, _linp_, and _phi_ respectively.

```
PROC GLIMMIX DATA= indata(rename=(response = y)) ;
CLASS group;
MODEL y = group x / link=log solution;
LSMEANS group / cl;
_variance_ = < distribution variance > ;
IF _mu_= .  or _linp_ = . then _logl_ = . ;
   else DO;
      IF ( _mu_ < 1E-12 ) then _logl_ = -1E20;
      else DO;
           _logl_ =  < enter log likelihood in terms of _mu_ and _phi_ > ; END;
   END;
RUN;
```

CONWAY-MAXWELL POISSON

The freight data set with under-dispersion can be evaluated with the COM Poisson regression model with the parameterization mu with GLIMMIX:

```
PROC GLIMMIX DATA=freight(rename=(broken=y)) ;
MODEL y = transfers / solution link=log ;
if _mu_= .  or _linp_ = . then _logl_ = . ;
   else do;
      if ( _mu_ < 1E-12 ) then _logl_ = -1E20;
      else do;
  S_ = 1;
  do i_ = 1 to 70;
   f_ = 1;
     do n_ = 1 to i_; f_ = f_  * ((_mu_/n_)**_phi_) ; end;
    S_ = S_ + f_;
   end;
  _logl_ = _phi_ *( y*LOG(_mu_) - lgamma(y+1)) - log(S_);
   end;
  end;
_variance_ = _mu_/_phi_  ;
run;
```

GENERALIZED POISSON

The FMM procedure can run the generalized Poisson regression model with the parameterization of Joe and Zhu (2005) as follows:

```
PROC FMM DATA=indat;
CLASS group;
MODEL y = group x / dist=gpoisson;
RUN;
```

This parameterization of the generalized Poisson regression model can also be run with PROC GLIMMIX by entering programming statements. See Example 45.14 Generalized Poisson Mixed Model for Overdispersed Count Data in the SAS/STAT, V. 14.1 documentation.

The generalized Poisson regression model can be run in R with the VGAM package:

```
PROC IML;
RUN ExportDataSetToR("indat", "indat");
SUBMIT / R;
library(VGAM)
vgl <- vglm(y ~ group + x, genpoisson(zero=1), data=indat )
summary(vgl)
ENDSUBMIT;
QUIT;
```

NEGATIVE BINOMIAL (P=1)

GLIMMIX runs the NB2 model with the dist=negbin option on the MODEL statement; the NB1 model can also be evaluated through entering programming statements for the log-likelihood.

```
PROC GLIMMIX data=ngbn  ;
CLASS group
MODEL model y = group x / link=log solution;
_variance_ = _mu_*( 1 + _phi_);
if _mu_= .  or _linp_ = . then _logl_ = . ;
  else do;
   if ( _mu_ < 1E-12 ) then _logl_ = -1E20;
   else do;
   _logl_ =  y*log(_phi_) - (y+(1/(_phi_/_mu_)))*log(1+ _phi_)
        + lgamma(y+(1/(_phi_/_mu_ ))) - lgamma(1/(_phi_/_mu_)) - lgamma(y+1); end;
 end;
RUN;
```

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 the R code with PROC IML