Multilevel Models

10. Models for Overdispersed Count Data

Germán Rodríguez

Princeton University

April 25, 2018

Negative Binomial

Count data often exhibit *overdispersion* relative to a Poisson model, in the sense that the variance exceeds the mean.

A solution is to add a multiplicative gamma random effect at level one, with mean one and variance σ^2. This results in a negative binomial model, for which the mean and variance are

$$E(Y) = \mu \quad \text{and} \quad \text{var}(Y) = \mu(1 + \sigma^2 \mu)$$

The variance here is a quadratic function of the mean.

The model can be extended to multiple levels by adding additional normal random effects in the log scale.

I often find, however, that this is overkill, as multilevel Poisson models already allow overdispersion.

Software Notes: Negative Binomial

Stata can fit random-intercept negative binomial models using `xtnbreg` and more general random-coefficient negative binomial models using `menbreg`.

In R there is a `glmer.nb()` function that extends `glmer()` to negative binomial models, using adaptive quadrature for random-intercept models and PQL for models with more than one random effect.

In addition, `rstanarm` has a `stan_glmer.nb()` function to fit these models using Hamiltonian Monte Carlo (HMC).

In the health-reform data a random-intercept NB model gives results similar to the Poisson model, and a random-slope model where the reform coefficient varies randomly turns out not to be identified, resulting in a reform variance of zero (even if you restrict the fit to women observed both times).

Excess Zeroes

Another frequent occurrence with count data is to observe an excess of zeroes compared to the Poisson standard. For example in the health reform data 30% of the observations have no doctor visits, whereas a simple Poisson model predicts only 11%.

A negative binomial model often helps improve matters. In the health reform data, using a negative binomial model predicts 31% with no visits, a much better fit. The random-slope model considered in the previous unit also predicts about 30% zero visits.

There are, however, two specialized models that introduce an additional equation to take care of the excess zeroes: zero-inflated and hurdle models.

Zero-Inflated Poisson

The zero-inflated Poisson model introduced by Lambert (1992) postulates the existence of a latent class where the outcome is always zero, and another class where the outcome is drawn from a Poisson distribution.

The model uses a logit equation to predict membership in the "always zero" class, and a log-linear equation for the mean of the Poisson distribution. Both can include covariates, and the model produces structural and random zeroes.

There is also a zero-inflated negative binomial model, but again I find that this is overkill, as either zero-inflation or the level-one random effect can often model the excess zeroes.

The model can be extended in principle to a multilevel setting, adding random intercepts and slopes.

Software Notes: Zero-Inflated

Single-level zero-inflated models can be fit in Stata using `zip` for Poisson and `zinb` for negative binomial.

In R I recommend the `pscl` package, which has a `zeroinf()` function, with a `dist` argument to specify the distribution as the default "poisson" or "negbin".

There are no packaged procedures in Stata or R for zero-inflated multilevel models, but these may be programmed in Stan.

Hurdle Models

An alternative approach uses two separate models:

- a logit model to distinguish zero and positive counts, and
- a zero-truncated Poisson model to represent the counts conditional on them exceeding zero.

One can also use a negative binomial distribution for the second step, but again I find that this is often overkill.

In this model there is only one kind of zero, which makes the distinction between zero and one or more clearer. Unfortunately the coefficients no longer have a simple interpretation in terms of relative effects on the mean, because the mean of the truncated part is $\mu/(1 - e^{-\mu})$ rather than μ. But one can always compute marginal effects.

Hurdle models can be extended to a multilevel setting by adding Gaussian random intercepts or slopes.

Software Notes: Hurdle Models

Fitting single-level hurdle models is easy because you fit separate logit and zero-truncated Poisson or negative binomial models.

In Stata the commands are `logit` and `tpoisson` (which supersedes `ztp`) for Poisson or `tnbreg` (which supersedes `ztnb`) for negative binomial.

In R you may use `glm()` for the Bernoulli part and the `VGAM` package, which has a function `vglm()` with a `family` argument that can be "pospoisson" or "posnegbinomial" for the truncated count portion.

Once again there are no packaged procedures in Stata or R for multilevel versions of hurdle models (or even the truncated count equation), but they can be programmed in Stan.

A Random-Intercept Hurdle Model

Here is the model we developed in class to fit a random-intercept hurdle model to the health reform data. We start with the data and parameters blocks:

```
dr_code = '
data {
  int N ;               // nobs
  int y[N];             // outcome
  int K;                // number of predictors
  row_vector[K] x[N];   // predictors
  int M;                // number of groups
  int g[N];             // mapping
  vector[2] Zero;       // mean of ri
}
parameters {
  real alpha1;           // logit equation
  vector[K] beta1;
  real alpha2;           // truncated-poisson equation
  vector[K] beta2;
  vector[2] u[M];             // random intercepts
  vector<lower=0>[2] sigma;   // st deviations of ri
  corr_matrix[2] Omega;       // correlation of ri
}
```

The model continues in the next slide.

The Model

Next we write a block to compute the covariance of the random effects and define the model, including the priors and likelihood

```
transformed parameters {
  cov_matrix[2] V;
  V = quad_form_diag(Omega, sigma);
}
model {
  alpha1 ~ normal(0,10);
  beta1 ~ normal(0,10);
  alpha2 ~ normal(0,10);
  beta2 ~ normal(0,10);
  u ~ multi_normal(Zero, V);
  for(n in 1:N) {
    (y[n] == 0) ~ bernoulli_logit(alpha1 + u[g[n]][1] + x[n] * beta1);
    if(y[n] > 0)
      y[n] ~ poisson(exp(alpha2 + u[g[n]][2] + x[n] * beta2))T[1,];
  }
}
'
```

The Bernoulli term contributes to the likelihood p for zeros and $1 - p$ for positive counts, and the Poisson term contributes a zero-truncated Poisson density for positive counts.

Fitting The Model

We read the data from the website, create a list and run the model

```
library(foreign)
dr <- read.dta("http://data.princeton.edu/pop510/drvisits.dta")
map <- function(id) { f <- table(id); rep(1:nrow(f), f) }
xvars = c("reform","age","educ","married","badh","loginc","summer")
dr_data <- list(N=nrow(dr), K=length(xvars), y = dr$numvisit,
  x = dr[,xvars], M = length(unique(dr$id)), g = map(dr$id),
  Zero = c(0,0))
library(rstan)
hri <- stan(model_code=dr_code, data=dr_data, chains=1, iter=1000)
```

The test run takes about one hour. The fixed effects look alright:

```
print(hri, pars=c("beta1[1]", "beta2[1]","sigma","Omega[1,2]"),probs=c(.025,.975),digits_summary=3)
...
             mean se_mean    sd   2.5%  97.5% n_eff  Rhat
beta1[1]     0.221   0.004 0.116  0.004  0.439  1000 1.010
beta2[1]    -0.015   0.001 0.036 -0.086  0.054  1000 0.999
sigma[1]     1.295   0.063 0.170  0.997  1.672     7 1.306
sigma[2]     0.787   0.006 0.032  0.728  0.856    28 1.076
Omega[1,2] -0.555   0.074 0.181 -0.856 -0.199     6 1.377
...
```

Unfortunately the results for the random effects are terrible, indicating lack of convergence and an effective sample size for the correlation of just 6!

An Alternative Model

I conclude that it is hard to estimate separate propensities for zero and positive counts. A simpler model postulates a single standard normal propensity z to visit a doctor. The logit equation has a term $\sigma_1 z$ to affect the probability of one or more visits, and the Poisson equation has a term $\sigma_2 z$ to affect the parameter μ.

This model runs in just about half an hour and yields sensible results:

```
> print(hgr, pars=c("beta1[1]", "beta2[1]", "sigma"),probs=c(.025,.975),digits_summary=3)
...
           mean se_mean    sd   2.5% 97.5% n_eff  Rhat
beta1[1] -0.189   0.003 0.102 -0.389 0.012  1000 0.999
beta2[1] -0.018   0.001 0.037 -0.087 0.056  1000 0.999
sigma[1]  0.917   0.010 0.142  0.647 1.198   192 1.013
sigma[2]  0.811   0.002 0.032  0.748 0.874   174 1.007
...
```

The reform has a large effect on whether women visit a doctor, and no effect on the number of visits of those who do. It would probably be worth running two longer chains to confirm the results.