Count Models

Week 10
POLS 8830: Advanced Quantitative Methods

Ryan Carlin
Georgia State University
rcarlin@gsu.edu

Count Data

- Observations literally count how many times a particular event happened in a period of time
  - Example: conflicts in a year, bills reported out of a committee during a congressional term, etc.
  - Note: the temporal effect is not an important predictor
  - Count variables are never negative
- Historically, count data analyzed using OLS
  - Results often were inefficient, inconsistent and biased
  - Problem occurs because OLS fits a line based on the observations only
  - Has difficulty accounting for the expected number of events

Count Data

- Example: suppose we observe an event occurring 5 times in the span of 1 year
- How should we interpret this observation?
  - Can we claim that over 2 years we should expect 10 events to occur?
- Perhaps we need to model a ratio of the number of observed events to the number of expected events:
  - $\dfrac{\#\text{ of observed events}}{\#\text{ of expected events}}$
- Such that the occurrence of an observed event does not change the expected number

Modeling Count Data

- Appropriately modeling this relationship requires selecting the correct statistical distribution — in this case the Poisson
- Recall the Poisson distribution

$$f_{Poisson}(y_i|\lambda) = \frac{e^{-\lambda}\lambda^{y_i}}{y_i!}$$

- where $\lambda > 0$ and $y_i = 0, 1, 2, \ldots$

The Poisson Distribution

- Is the most basic model for count data
- Has the defining characteristic that the conditional mean of the outcome (μ) equals the conditional variance (λ)
  - In reality, the conditional variance is often greater than the conditional mean
  - This leads to a problem called overdispersion
  - A second problem involves the number of predicted 0s, which is often much less than the observed number of 0s

Properties of the Poisson Distribution

- If the conditional variance equals the conditional mean (as assumed), this is called equidispersion
- As λ increases (i.e. higher rate), the probability of 0 decreases
- As λ increases the mass of the distribution shifts its skew to the right becoming more bell shaped
- As λ and y_i increase, the Poisson distribution converges to the Normal distribution

Key Assumption of the Poisson

- Events are independent
- When an event occurs it must not affect the probability of another event occurring in the future
- The denominator (expected number of events) must remain constant
- Problem with this assumption
  - Changing the rate (λ) across individual observations leads to heterogeneity in the likelihood (probability) of future events
  - This leads to overdispersion of the model's fit
  - The result is that our estimates are inefficient

Key Assumption of the Poisson

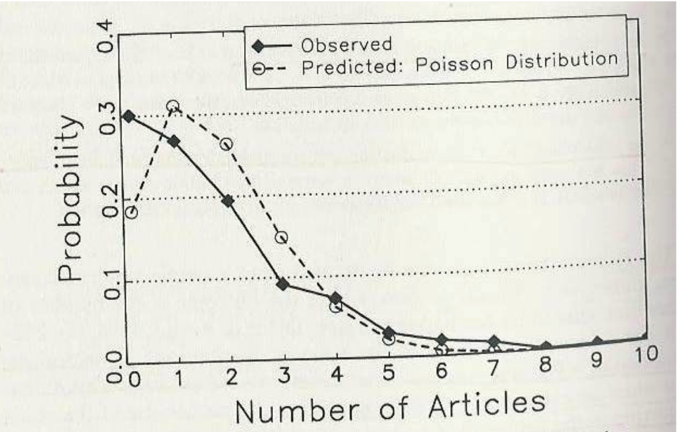

Figure 8.2. Distribution of Observed and Predicted Counts of Articles

Application of the Poisson Model

- Because we cannot observe the population parameter (λ), we rewrite the equation using our combination of independent variables and their estimated coefficients
- $\lambda = E(y_i|\mathbf{X}) = exp(\mathbf{X}\beta)$
  - Note: taking the exponential of $\mathbf{X}\beta$ forces λ to be positive

Application of the Poisson Model

- Likelihood function:

$$L(\beta|y, \mathbf{X}) = \prod_{i=1}^{N} \frac{e^{-e^{\mathbf{X}\beta}} e^{\mathbf{X}\beta^{y_i}}}{y_i!}$$

- Taking the natural log gives us:

$$\ln L = \sum_{i=1}^{N} \frac{e^{-e^{\mathbf{X}\beta}} e^{\mathbf{X}\beta^{y_i}}}{y_i!}$$

Poisson Model in R

- Syntax: `glm(DV ~ IV ..., data=df, family="poisson", ...)`

```
Call:
glm(formula = number_of_victims ~ number_of_perpetrators + GDP +
    GDP_Growth + Trade_Perc_GDP + Mineral_Rents_Perc, family = "poisson",
    data = combo.df[combo.df$type_of_attack == 3, ])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-51.015   -1.887   -1.483   -0.333   53.099

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)            1.310e+00  1.004e-02  130.53   <2e-16 ***
number_of_perpetrators 5.940e-03  2.362e-05  251.47   <2e-16 ***
GDP                   -5.950e-13  9.417e-15  -63.18   <2e-16 ***
GDP_Growth             1.348e-02  4.188e-04   32.19   <2e-16 ***
Trade_Perc_GDP        -1.748e-03  1.394e-04  -12.54   <2e-16 ***
Mineral_Rents_Perc    -1.704e-01  9.366e-03  -18.19   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 269229  on 43819  degrees of freedom
Residual deviance: 239734  on 43814  degrees of freedom
  (14978 observations deleted due to missingness)
AIC: 289031

Number of Fisher Scoring iterations: 11
```

- Note: $N \cong 40{,}000$

Negative Binomial Model

- Theory behind the Negative Binomial Model
  - Rarely does the Poisson regression yield ‘good’ estimates in practice
  - For most data the conditional variance is larger than the conditional mean (i.e. overdispersion)
- The Negative Binomial model accounts for this occurrence
  - It estimates an additional parameter (α) that reflects any unobserved heterogeneity in the probability of an event occurring

Comparing Negative Binomial and Poisson

- Negative Binomial adds an error term (δ) that is assumed to be uncorrelated with the IVs
- Both Poisson and Negative Binomial have same structure for estimating coefficients of the independent variables
- Expected rates between models will be similar
- Standard errors in the Poisson model will be biased downwards, leading to overly inflated z-scores which leads to spurious results

Application of the Negative Binomial

- The logic behind the Poisson Regression is still valid for the Negative Binomial

$$\Pr(y_i|\mathbf{X},\delta) = \frac{e^{-\lambda}\lambda^{y_i}}{y_i!}$$

- Because δ is unknown, we cannot compute the $\Pr(y_i|\mathbf{X})$ directly
  - Instead, we must assume δ is drawn from a separate probability distribution (Gamma)
  - We can then compute $\Pr(y_i|\mathbf{X})$ as a weighted combination of $\Pr(y_i|\mathbf{X},\delta)$ for all values of δ

Application of the Negative Binomial

- This assumption leads to

$$\Pr(y_i|\mathbf{X}) = \frac{\Gamma(y+\alpha^{-1})}{y!\Gamma(\alpha^{-1})}\left(\frac{\alpha^{-1}}{\alpha^{-1}+\lambda}\right)^{\alpha^{-1}}\left(\frac{\lambda}{\alpha^{-1}+\lambda}\right)^{y}$$

- Where Γ represents the Gamma function and α determines the amount of dispersion
- When $\alpha = 0$ the Negative Binomial distribution converges to a Poisson distribution

Negative Binomial Model in R

- Syntax: `glm(DV ~ IV ..., data=df, family=negative.binomial(theta = 1), ...)`

```
Call:
glm(formula = number_of_victims ~ number_of_perpetrators + GDP +
    GDP_Growth + Trade_Perc_GDP + Mineral_Rents_Perc, data = combo.df[combo.df$type_of_attack ==
    3, ])

Deviance Residuals:
   Min       1Q   Median       3Q      Max
-28.49    -1.44    -1.15    -0.18   377.56

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             5.130e+00  1.083e-01  47.379  < 2e-16 ***
number_of_perpetrators  3.723e-02  7.698e-04  48.368  < 2e-16 ***
GDP                    -4.501e-13  3.004e-14 -14.984  < 2e-16 ***
GDP_Growth              3.171e-02  4.903e-03   6.467 1.01e-10 ***
Trade_Perc_GDP         -1.202e-03  1.155e-03  -1.040    0.298
Mineral_Rents_Perc     -3.340e-01  5.279e-02  -6.326 2.55e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 38.65833)

    Null deviance: 1793281  on 43819  degrees of freedom
Residual deviance: 1693776  on 43814  degrees of freedom
  (14978 observations deleted due to missingness)
AIC: 284515

Number of Fisher Scoring iterations: 2
```

Negative Binomial Model in R

- Syntax: `glm(DV ~ IV ..., data=df, family=negative.binomial(theta = 1), ...)`
- This relies on the `negative.binomial()` call which comes from the **MASS** package
    - Can also use `family = "negbin"` which is just theta=1
- theta is the parameter of the negative binomial distribution.
- theta = 1 forms a special case, which converges to the geometric distribution.
- theta = 1 is usually fine, but theta can be parameterized to be estimated in the model with the `glm.nb()` command from the **MASS** package
- `glm.nb()` requires you to initialize theta, with the `init.theta()` call

Zero-Inflated Count Models

- Developed in response to the failure of the Poisson model to account for dispersion and excess zeros
  - Does so by changing how the mean structure is estimated
  - Allows the zeros to be generated by another process than the one modeled for actual counts
- Assumes two latent (unobserved) groups
  - Always 0 group — has outcome zero with probability 1
  - Not always 0 group — might have a zero, but also has non-zero probability of having an actual count

Zero-Inflated Count Models

- General process for Zero-Inflated Counts
  - One model accounts for the membership of Group A (Always 0 group)
  - A second model estimates the membership of Group B (Not-Always 0 group)
  - Computes the overall probabilities as a mixture of the probabilities for each group

Estimating Zero-Inflated Count Models

- Calculate the likelihood of an individual observation belonging to Group A using a traditional logit or probit model
    - Note: this overestimates the number of 0 counts
- For the observations not predicted to ALWAYS be 0, model the probability of the count (where 0 is still a possibility)
    - Model as a Poisson or Negative Binomial
- Mix the probabilities according to the proportion of individual observations predicted to be in each group

Estimating Zero-Inflated Count Models

- **pscl** package
- Both follow the standard `glm()` format but use | to specify the logit element as in MLMs
- R syntax — Zero-Inflated Poisson:
  - `zeroinfl(DV ~ IV ...| IV ..., data=df, dist="poisson")`
    - Note: "dist" not "family"
- R syntax — Zero-Inflated Negative Binomial:
  - `zeroinfl(DV ~ IV ...| IV ..., data=df, dist="negbin", link = "logit")`
    - Note: "dist" not "family"

Zero-inflated Poisson

```
Call:
zeroinfl(formula = number_of_victims ~ number_of_perpetrators + scale(GDP) + GDP_Growth + Trade_Perc_GDP + Mineral_Rents_Perc |
    number_of_perpetrators, data = combo.df[combo.df$type_of_attack == 3, ])

Pearson residuals:
     Min       1Q   Median       3Q      Max
-27.6669  -0.6575  -0.6310  -0.1791 160.6168

Count model coefficients (poisson with log link):
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)            1.9674099  0.0120907 162.721  < 2e-16 ***
number_of_perpetrators 0.0048189  0.0000283 170.255  < 2e-16 ***
scale(GDP)            -0.1695054  0.0091923 -18.440  < 2e-16 ***
GDP_Growth             0.0046803  0.0005837   8.019 1.07e-15 ***
Trade_Perc_GDP        -0.0034515  0.0001865 -18.507  < 2e-16 ***
Mineral_Rents_Perc    -0.0301634  0.0073189  -4.121 3.77e-05 ***

Zero-inflation model coefficients (binomial with logit link):
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)           -0.4985593  0.0255265  -19.53   <2e-16 ***
number_of_perpetrators -0.0100670  0.0002775  -36.28   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Number of iterations in BFGS optimization: 12
Log-likelihood: -1.104e+05 on 8 Df
```

- The upper half is the poisson portion, while the bottom is the logit component

Zero-inflated Negative Binomial

```
Call:
zeroinfl(formula = number_of_victims ~ number_of_perpetrators + scale(GDP) + GDP_Growth + Trade_Perc_GDP + Mineral_Rents_Perc |
    number_of_perpetrators, data = combo.df[combo.df$type_of_attack == 3, ], dist = "negbin", link = "logit")

Pearson residuals:
      Min        1Q    Median        3Q       Max
 -0.52670  -0.47756  -0.46213  -0.07279 147.20059

Count model coefficients (negbin with log link):
                        Estimate Std. Error  z value Pr(>|z|)
(Intercept)            1.5886798  0.0385608   41.199  < 2e-16 ***
number_of_perpetrators 0.0144586  0.0002652   54.523  < 2e-16 ***
scale(GDP)            -0.3433960  0.0130103  -26.394  < 2e-16 ***
GDP_Growth             0.0168979  0.0017442    9.688  < 2e-16 ***
Trade_Perc_GDP        -0.0011521  0.0004607   -2.501   0.0124 *
Mineral_Rents_Perc    -0.0899891  0.0163729   -5.496 3.88e-08 ***
Log(theta)            -1.2822611  0.0109755 -116.829  < 2e-16 ***

Zero-inflation model coefficients (binomial with logit link):
                       Estimate Std. Error z value Pr(>|z|)
(Intercept)           -21.12308  997.37645  -0.021    0.983
number_of_perpetrators -0.06154   10.60126  -0.006    0.995
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Theta = 0.2774
Number of iterations in BFGS optimization: 73
Log-likelihood: -6.608e+04 on 9 Df
```

- The upper half is the negative binomial portion, while the bottom is the logit component

Zero-inflated Models

- You can test the appropriateness over the ZI models versus the Poisson or Negative-Binomial models through the Vuong non-nested hypothesis test included in the pscl package
  - `vuong(model1, model2, ...)`
  - Model 1: non-zero inflated; Model 2: zero-inflated
  - Models from `glm`, `negbin`, or `zeroinfl()`
  - Significant results indicate appropriateness of ZI model

```
NA or numerical zeros or ones encountered in fitted probabilities
dropping these 5 cases, but proceed with caution
Vuong Non-Nested Hypothesis Test-Statistic:
(test-statistic is asymptotically distributed N(0,1) under the
 null that the models are indistinguishible)
-------------------------------------------------------------
              Vuong z-statistic             H_A    p-value
Raw                   -37.39060 model2 > model1 < 2.22e-16
AIC-corrected         -37.38831 model2 > model1 < 2.22e-16
BIC-corrected         -37.37838 model2 > model1 < 2.22e-16
```