Generalized Linear Models

Lecture 8. Count data models II. Negative binomial model

Count data with (big) overdispersion

Big overdispersion, i.e $\mathbf{D}Y > 5\mathbf{E}Y$: Poisson distribution does not fit

One possible solution: **negative binomial distribution**
Keep in mind that negative binomial can also have overdispersion

Possible choices for negative binomial (*NB*) model:

- "regular" NB model (sometimes also referred to as NB(2))
- models with geometric distribution
- zero-modified (*ZINB, ZTNB, ZANB*) models

$+$ *NB(P)*, censored *NB*, *NB* with mixed effects, etc. Hilbe (2007) proposes 22 different types of NB models.

Remark: *NB* model with large parameter k can not be distinguished from Poisson model

J. M. Hilbe (2007). Negative Binomial Regression. Cambridge University Press

Negative binomial distribution $NB(k, \pi)$ (classic notation)

Anscombe (1949) – First NB model (*The Statistical Analysis of Insect Counts Based on the Negative Binomial Distribution*)

Plackett (1981), Lawless (1987) – log-likelihood for NB

Interpretation: NB distribution is known as the distribution of 'failures' until k-th 'success' in a Bernoulli process

$Y \sim NB(k, \pi)$, $0 < \pi < 1$; $k > 0$, usually integer; π – probability of 'success'

Pmf of NB distribution:

$$p(y; k, \pi) = \frac{\Gamma(k + y)}{y!\ \Gamma(k)}\ \pi^k (1 - \pi)^y$$

- mean: $\mathbf{E}Y = \mu = k(1 - \pi)/\pi$
- variance: $\mathbf{D}Y = k(1 - \pi)/\pi^2 = \mu + \frac{1}{k}\mu^2$

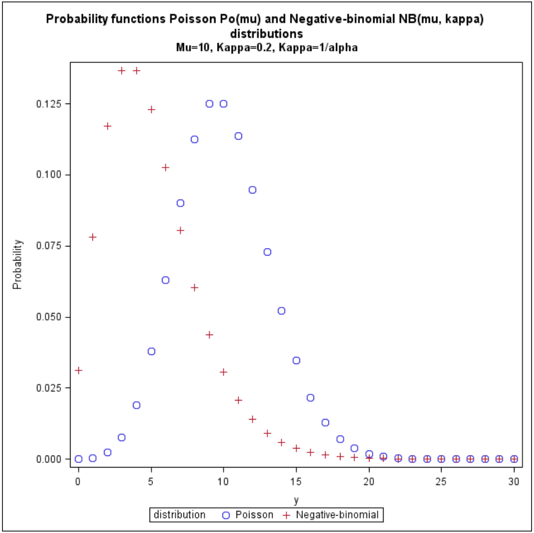
Distributions related to NB

The following distributions can be considered as sub cases of $NB(k, \pi)$:

1. Geometric distribution ($k = 1$)
2. Pascal distribution (integer k)
3. Polya distribution (real-valued k)

NB vs Poisson:
NB has more probability on zeros and heavier right tail (given the equal mean)

Poisson vs NB (mean $\mu = 10$)



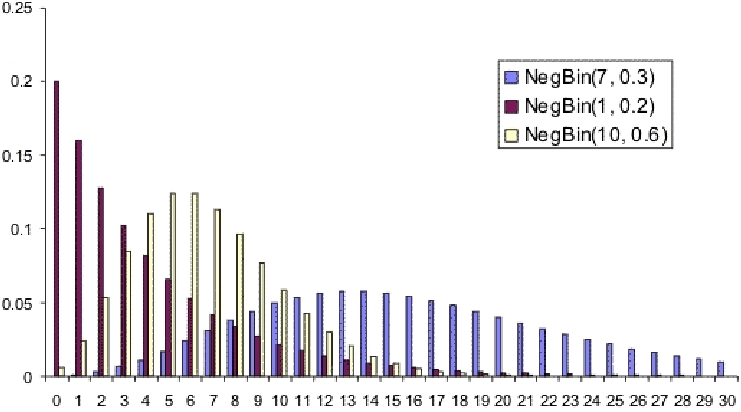
NB as a mixture of Poisson and Gamma

NB distribution can be interpreted as a Poisson-gamma mixture, i.e. conditional on a gamma-distributed variable Z, the variable Y has a Poisson distribution with mean Z

$$Y|Z = z \sim Po(z), \quad Z \sim \Gamma(\alpha, \lambda)$$

The resulting distribution is NB with $k = \alpha$ and $\pi = \frac{\lambda}{\lambda+1}$

Example. NB with different parameters



$NB(k, \pi_i)$ as a member of exponential family

Let us start with the pmf

$$p(y_i; k, \pi_i) = \frac{\Gamma(k + y_i)}{y_i! \ \Gamma(k)} \ \pi_i^k (1 - \pi_i)^{y_i}$$

and rewrite it in a form similar to exponential family

$$p(y_i; k, \pi_i) = \exp\{y_i \ln(1 - \pi_i) + k \ln \pi_i + \ln \Gamma(k + y_i) - \ln[y_i!\Gamma(k)]\}$$

Now

- $\theta_i = \ln(1 - \pi_i)$ and $\pi_i = 1 - \exp(\theta_i)$
- $b(\theta_i) = -k \ln \pi_i = -k \ln(1 - \exp \theta_i)$
- $\varphi_i = 1$
- mean $b^{'}(\theta_i) = ... = \frac{k(1-\pi_i)}{\pi_i} = \mu_i$
- variance $\varphi_i \cdot b^{''}(\theta_i) = ... = \mu_i + \frac{\mu_i^2}{k}$

Prove it!

$NB(\mu_i, k)$ as a member of exponential family

Since in GLM context we are interested in modelling the means, a reparametrized version of NB ($NB(\mu_i, k)$ with $\mu_i = \frac{k(1-\pi_i)}{\pi_i}$) can be more useful:

$$p(y_i; \mu_i, k) = \frac{\Gamma(k + y_i)}{y_i! \; \Gamma(k)} \; (\frac{k}{k+\mu_i})^k (1 - \frac{k}{k+\mu_i})^{y_i}$$

To show that this pmf belongs to exponential family, we rewrite it as

$$p(y_i; \mu_i, k) = \exp\{y_i \ln \frac{\mu_i}{k+\mu_i} + k \ln \frac{k}{k+\mu_i} + \ln \Gamma(k + y_i) - \ln[y_i!\Gamma(k)]\}$$

Thus

- $\theta_i = \ln \frac{\mu_i}{k+\mu_i}$
- $b(\theta_i) = -k \ln \frac{k}{k+\mu_i}$
- $b^{'}(\theta_i) = \mu_i$
- $b^{''}(\theta_i) = \mu_i + \frac{\mu_i^2}{k}$

Link functions used in *NB* models

(1) Canonical link:

$$\eta_i = g(\mu_i) = \ln \frac{\mu_i}{k + \mu_i} = -\ln(\frac{k}{\mu_i} + 1)$$

The corresponding response function:

$$\mu_i = h(\eta_i) = \frac{k}{\exp(-\eta_i) - 1}$$

(2) Log-link

$$\eta_i = g(\mu_i) = \ln(\mu_i), \quad \mu_i = h(\eta_i) = \exp(\eta_i)$$

(3) Identity link

Remark: Model with canonical link is difficult to interpret, *log*-link is used because of analogy with Poisson model and gives better results

Deviance of NB model

By definition, $D = 2((l(\boldsymbol{y}, \boldsymbol{y}) - l(\boldsymbol{y}, \hat{\boldsymbol{\mu}}))$

Using the $NB(\mu_i, k)$ parametrization, the deviance can be expressed as (prove it!):

$$D = 2\sum_i [y_i \ln \frac{y_i}{\hat{\mu}_i} - (y_i + k) \ln \frac{k + y_i}{k + \hat{\mu}_i}]$$

Notice that the first term is the same as for deviance of Poisson model

Poisson *vs* NB

Hypotheses:

$$\begin{cases} H_0: & \mathbf{D}Y_i = \mu_i & \text{Poisson dist.} \\ H_1: & \mathbf{D}Y_i = \mu_i + \alpha\mu_i^2 & \text{NB dist., } \alpha = \frac{1}{k} \end{cases}$$

or, in general,

$H_1 : \mathbf{D}Y = \mu_i + \alpha f(\mu_i)$, where $f(\cdot)$ is some function

The hypotheses can be written explicitly for α as:

$$\begin{cases} H_0: & \alpha = 0 \ \text{(Poisson dist.)} \\ H_1: & \alpha > 0 \ \text{(NB dist.)} \end{cases}$$

Main advantage of NB model compared to Poisson:
more flexible variance structure allows to estimate data with bigger variability, but *is not suitable to model underdispersion* (Tutz, 2012)

Testing Poisson *vs* NB (H_0 means Poisson)

- Cameron ja Trivedi (1996) test:
  α is estimated from $(y_i - \hat{\mu}_i)^2 - y_i = \alpha\hat{\mu}_i^2 + \varepsilon_i$, if α is significant $\Rightarrow H_1$
- Wooldridge (1996) test:
  α is estimated from $(y_i - \hat{\mu}_i)^2 - \hat{\mu}_i = \alpha\hat{\mu}_i^2 + \varepsilon_i$
- Lagrange multiplier test (Greene, 2002), score test (Rao, 1973), Wald test
- Likelihood ratio test (based on the fact that Poisson model is special case of NB model):

$$2(\ln L_{NB} - \ln L_{Pois}) \sim \chi_1^2$$

  NB! One-sided hypothesis: $H_1 : \alpha > 0$, i.e. one should use critical values $\chi^2_{\alpha_2,1}$ (e.g. $\chi^2_{0.05,1} = 3.8, \quad \chi^2_{0.1,1} = 2.7$)

  Decision rules?

Example. Cellular differentiation (1)

The effect of two agents of immuno-activating ability that may induce cell differentiation was investigated. As response variable the number of cells that exhibited markers after exposure was recorded. It is of interest if the agents TNF (tumor necrosis factor) and IFN (interferon) stimulate cell differentiation independently, or if there is a synergetic effect. 200 cells were examined at each dose combination.

The data is also analyzed in Fahrmeir & Tutz (1994), and available in R package `Fahrmeir`

The dataset contains 16 observations and 3 variables:

- `y` – number of cells differentiating
- `TNF` – dose of TNF, U/ml
- `IFN` – dose of IFN, U/ml

Poisson model for estimating the number of differentiating cells based on doses of TNF and IFN:

$$\mu = \exp(\beta_0 + \beta_1 \mathit{TNF} + \beta_2 \mathit{IFN} + \beta_3 \mathit{TNF} * \mathit{IFN})$$

Example. Cellular differentiation (2)

```
> library(Fahrmeir)
> data(cells)
> modelP=glm(y~TNF*IFN,family="poisson",data=cells)
> summary(modelP)
...
               Estimate  Std. Error z value  Pr(>|z|)
(Intercept)  3.43563627  0.06376778  53.877   < 2e-16 ***
TNF          0.01552810  0.00083085  18.689   < 2e-16 ***
IFN          0.00894613  0.00096685   9.253   < 2e-16 ***
TNF:IFN     -0.00005670  0.00001348  -4.205 0.0000261 ***
...
(Dispersion parameter for poisson family taken to be 1)
    Null deviance: 707.03  on 15  degrees of freedom
Residual deviance: 142.39  on 12  degrees of freedom
AIC: 243.69
```

Clearly we have overdispersion, since:

```
> modelP$deviance/modelP$df.residual
[1] 11.86544
```

Example. Cellular differentiation (3)

```
> modelQP=glm(y~TNF*IFN,family="quasipoisson",data=cells)
> summary(modelQP)
...
               Estimate  Std. Error t value Pr(>|t|)
(Intercept)  3.43563627  0.21844859  15.727 2.26e-09 ***
TNF          0.01552810  0.00284622   5.456 0.000146 ***
IFN          0.00894613  0.00331213   2.701 0.019273 *
TNF:IFN     -0.00005670  0.00004619  -1.227 0.243176
...
(Dispersion parameter for quasipoisson family taken to be 11.73534)
    Null deviance: 707.03  on 15  degrees of freedom
Residual deviance: 142.39  on 12  degrees of freedom
```

Notice that taking overdispersion into account makes the coefficient of the interaction term nonsignificant. We can also see that the overdispersion is estimated using Pearson residuals:

```
> sum(residuals(modelP,type="pearson")^2)/modelP$df.residual
[1] 11.73516
```

Example. Cellular differentiation (4)

One can argue further that

- overdispersion is too big to apply (quasi)Poisson model
- the variance structure in data does not correspond to Poisson model (as argued in Fahrmeir&Tutz):

```
> library(sqldf)
> sqldf("select TNF, avg(y) as mean_y,
         variance(y) as var_y from cells group by TNF")
  TNF mean_y     var_y
1   0  22.00  143.3333
2   1  45.25  400.9167
3  10  74.00 1608.6667
4 100 161.50 1655.0000
```

It is questionable whether the variance has linear relation to mean $\phi\mu_i$ as in quasipoisson or, e.g., $\mu_i + \frac{\mu_i^2}{k}$ (negative binomial)

Example. Cellular differentiation (5)

```
> library(MASS)
> modelNB=glm.nb(y~TNF*IFN,data=cells)
> summary(modelNB)
...
               Estimate  Std. Error z value    Pr(>|z|)
(Intercept)  3.40042871  0.16254096  20.920     < 2e-16 ***
TNF          0.01613032  0.00308130   5.235 0.000000165 ***
IFN          0.00933325  0.00307969   3.031     0.00244 **
TNF:IFN     -0.00005880  0.00005964  -0.986     0.32422
...
(Dispersion parameter for Negative Binomial(6.4237) family taken to
    Null deviance: 61.881  on 15  degrees of freedom
Residual deviance: 16.763  on 12  degrees of freedom
AIC: 156.88
              Theta:  6.42
          Std. Err.:  2.59
 2 x log-likelihood:  -146.882
```

Example. Cellular differentiation (6)

```
> modelQP=update(modelQP,.~.-TNF:IFN)
> modelNB=update(modelNB,.~.-TNF:IFN)
> cbind(coef(summary(modelQP))[,c(1,4)],
        coef(summary(modelNB))[,c(1,4)])
               Estimate     Pr(>|t|)    Estimate       Pr(>|z|)
(Intercept) 3.573116655 4.371668e-11 3.451510118 1.223505e-108
TNF         0.013142274 2.851438e-05 0.014421179  8.685567e-09
IFN         0.005854408 2.206719e-02 0.007751195  3.046597e-03
```

As we can see, the results of QP and NB model are quite similar, but the coefficients still differ. Taking into account the size of overdispersion, NB model is preferred.

Geometric distribution

Geometric distribution is a special case of NB distribution if $k = \alpha = 1$

Starting from the classical form of NB pmf we get $p(y; 1, \pi) = (1-\pi)^y \pi$, $y = 0, 1, 2, \ldots$
Mean is $\frac{1-\pi}{\pi}$, variance is $\frac{1-\pi}{\pi^2}$

In GLM context, let us use the $NB(\mu_i, k)$-parametrization:

$$p(y_i; \mu_i, k) = \exp\{y_i \ln \frac{\mu_i}{k+\mu_i} + k \ln \frac{k}{k+\mu_i} + \ln \Gamma(y_i + k) - \ln[y_i!\Gamma(k)]\}$$

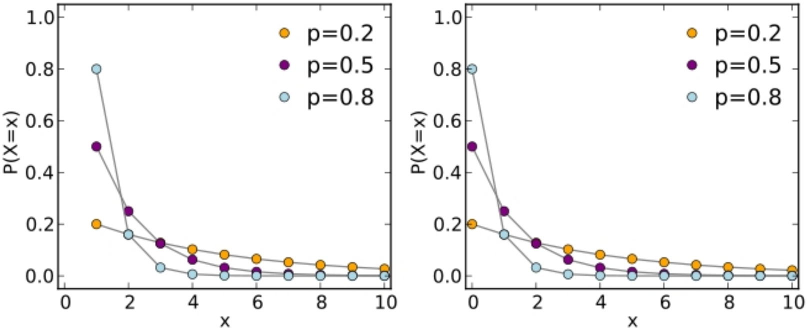
The formula simplifies since $k = 1$:

$$p(y_i; \mu_i, 1) = \exp[y_i \ln \frac{\mu_i}{1+\mu_i} + \ln \frac{1}{1+\mu_i}] = \exp[y_i \ln \frac{\mu_i}{1+\mu_i} - \ln(1+\mu_i)]$$

Canonical link:

$$g(\mu_i) = \ln \frac{\mu_i}{1+\mu_i} = -\ln(\frac{1}{\mu_i} + 1)$$

Geometric distribution. Example



Left: $p(x) = (1-p)^{x-1}p, \; x = 1, 2, \ldots$
Right: $p(x) = (1-p)^{x}p, \; x = 0, 1, 2, \ldots$

NB models with fixed k in R

Negative binomial models in R:

- if we don't know k ($k =$ `theta` in R): use `glm.nb()`
- if we know (have estimated) k: use `glm(...,family="negative.binomial"(theta=...))`
- for geometric distribution: `glm(...,family="negative.binomial"(theta=1))`