Negative Binomial

Parametrisation

The negative Binomial distribution is

$$\text{Prob}(y) = \frac{\Gamma(y+n)}{\Gamma(n)\Gamma(y+1)} p^n (1-p)^y$$

for responses $y = 0, 1, 2, \ldots$, where

n: number of successful trials (*size*), or dispersion parameter. Must be strictly positive, need not be integer.

p: probability of success in each trial.

Link-function

The mean and variance of y are given as

$$\mu = n\frac{1-p}{p} \qquad \text{and} \qquad \sigma^2 = \mu(1 + \frac{\mu}{n})$$

and the mean is linked to the linear predictor by

$$\mu = E\exp(\eta)$$

where the hyperparameter n (*size*) plays the role of an overdispersion parameter. E represents knows constant and $\log(E)$ is the offset of η.

Hyperparameters

The default parameterization (`variant=0`) is that the overdispersion parameter n (*size*) is represented as

$$\theta = \log(n)$$

and the prior is defined on θ.

An alternative parameterization (`variant=1`) is that the overdispersion parameter n (*size*) is represented as

$$\theta = \log(n) - \log(E)$$

and the prior is defined on θ.

Specification

- family = `nbinomial`
- Required arguments: y and E (default $E = 1$).
- Chose variant with either `control.family = list(variant=0)` (default) or `control.family = list(variant=1)`

Hyperparameter spesification and default values

hyper

theta

name size

short.name size

initial 2.30258509299405

fixed FALSE

prior loggamma

param 1 1

to.theta `function(x) log(x)`

from.theta `function(x) exp(x)`

survival FALSE

discrete TRUE

link default log logoffset

pdf nbinomial

Example

In the following example we estimate the parameters in a simulated example with negative binomial responses and assign the hyperparameter θ a Gaussian prior with mean 0 and precision 0.01

```
n = 1000
x = rnorm(n,  sd = 0.2)
eta = 1 + x
E = runif(n, min = 0, max=10)

mu = E * exp(eta)
size = 3
y = rnbinom(n, size=size, mu=mu)
r = inla(y ~ 1 + x, data = data.frame(y, x, E),
    family = "nbinomial", E=E)

mu = E * exp(eta)
size = E*3
y = rnbinom(n, size=size, mu=mu)
rr = inla(y ~ 1 + x, data = data.frame(y, x, E),
    family = "nbinomial",
    control.family = list(variant = 1),
    E=E)
```

Notes

As $n \to \infty$, the negative Binomial converges to the Poisson distribution. For numerical reasons, if n is too large:

$$\frac{\mu}{n} < 10^{-4},$$

then the Poisson limit is used.