

# Beta-binomial Normal-approximation

## Parametrisation

The Beta-binomial distribution arise from a hierarchical model where the probability  $p$  is Beta-distributed

$$\pi(p) = \frac{1}{B(\alpha, \beta)} p^{\alpha-1} (1-p)^{\beta-1}, \quad \alpha > 0, \beta > 0$$

and the response  $y$  is Binomial

$$\pi(y | p) = \binom{n}{y} p^y (1-p)^{(n-y)}, \quad y = 0, 1, \dots, n.$$

The marginal distribution for  $y$  is the Beta-binomial,

$$\pi(y) = \binom{n}{y} \frac{B(y + \alpha, n - y + \beta)}{B(\alpha, \beta)}.$$

The mean and variance of  $y$  are given as

$$\mu = n \frac{\alpha}{\alpha + \beta} = n\mu_p, \quad \text{and} \quad \sigma^2 = n\mu_p(1 - \mu_p)(1 + (n-1)\rho),$$

where  $\mu_p = \frac{\alpha}{\alpha + \beta}$  is the mean of the probability  $p$  from the Beta-distribution, and  $\rho = \frac{1}{\alpha + \beta + 1}$  ( $0 < \rho < 1$ ) is the pairwise correlation between the  $n$  Bernoulli draws and an over-dispersion parameter.

The “Beta-Binomial Normal approximation”-model implements a Normal-approximation to this distribution, treating  $y$  as a continous Normal distributed variable, with mean  $n\mu_p$  and variance  $n\mu_p(1 - \mu_p)(1 + s(n-1)\rho)$ . We have added a constant scaling  $s \geq 0$  for more flexibility, where  $s = 1$  gives the Beta-Binomial above.

## Link-function

The mean probability  $\mu_p$  is linked to the linear predictor by

$$\mu_p = \frac{\exp(\eta)}{1 + \exp(\eta)}$$

using the default logit-link.

## Hyperparameter

The hyperparameter is the over-dispersion parameter  $\rho$ , which is represented as

$$\rho = \frac{\exp(\theta)}{1 + \exp(\theta)}$$

and the prior is defined on  $\theta$ .

## Specification

- family = `betabinomialna`
- Required arguments:  $y$  and `Ntrials = n` (default `Ntrials = 1`), and `scale = s` (default `s = 1`).

## Hyperparameter specification and default values

**doc** The Beta-Binomial Normal approximation likelihood

**hyper**

**theta**

**hyperid** 62101

**name** overdispersion

**short.name** rho

**initial** 0

**fixed** FALSE

**prior** gaussian

**param** 0 0.4

**to.theta** function(x) log(x / (1 - x))

**from.theta** function(x) exp(x) / (1 + exp(x))

**survival** FALSE

**discrete** TRUE

**link** default logit loga cauchit probit cloglog loglog robit sn

**pdf** betabinomialna

## Example

In the following example we estimate the parameters in a simulated example.

```
n = 300
rho = 0.2
z = rnorm(n, sd=0.2)
Ntrials = sample(10:20, n, replace=TRUE)
eta = 1 + z
p.eta = exp(eta)/(1+exp(eta))
a = p.eta * (1-rho)/rho
b = (p.eta * rho - p.eta - rho + 1)/rho
p = rbeta(n, a, b)
y = rbinom(n, Ntrials, p)

formula = y ~ 1 + z
data = data.frame(y, z)
## exact
r = inla(formula, data = data,
         family = "betabinomial", Ntrials=Ntrials)
summary(r)
## approximate
ra = inla(formula, data = data,
          family = "betabinomialna", Ntrials=Ntrials)
summary(ra)
```

```

## exact simulation from the approximate model
n = 1000
rho = 0.1
z = rnorm(n, sd=0.4)
Ntrials = sample(1:20, n, replace=TRUE)
eta = 1 + z
p = exp(eta)/(1+exp(eta))
s = runif(n)
m = Ntrials * p
v = Ntrials * p * (1.0 - p) * (1.0 + s * (Ntrials - 1) * rho)
y = rnorm(n, mean = m, sd = sqrt(v))

formula = y ~ 1 + z
data = data.frame(y, z, s)
r = inla(formula, data = data, scale = s,
        family = "betabinomialna", Ntrials=Ntrials, verbose = TRUE,
        control.inla = list(strategy = "adaptive"))
summary(r)

```

## Notes

None.