

# Binomial-mixture likelihood

## Parametrisation

### Details

This likelihood is a mixture of three binomials, where two of them are simple predictors and one is general. The probability for success is

$$p = w_1 p_1 + w_2 p_2 + w_3 p_3$$

where  $w_1 + w_2 + w_3 = 1$  are the positive weights,

$$\text{logit}(p_1) = \beta_1 z_1 + \beta_2 z_2 + \beta_3 z_3 + \beta_4 z_4 + \beta_9 z_9,$$

$$\text{logit}(p_2) = \beta_5 z_5 + \beta_6 z_6 + \beta_7 z_7 + \beta_8 z_8 + \beta_9 z_{10}$$

and

$$\text{logit}(p_3) = \eta + \beta_9 z_{11}$$

for fixed covariates  $z_i$ . The linear predictor  $\eta$  is defined in the formula.

**Note:**  $\beta_9$  is the *same* variable in the three expressions.

## Link-function

The link-function is given as usual, and they are all equal.

## Hyperparameters

The nine regression coefficients  $\beta_i$  are treated as hyperparameters.

## Specification

- `family="binomialmix"`
- Required arguments: A  $n \times 2$  matrix  $Y$  with the observations and the number of trials  $s$ ,  $Y = (y, s)$ , a  $n \times 11$  matrix  $Z$  with the covariates  $Z = c(z_1, \dots, z_{11})$ , and a  $n \times 2$  matrix  $W$  with weights  $W = (w_1, w_2)$ . The `inla.mdata` is used as

```
inla.mdata(Y, Z, W) ~ ...
```

## Hyperparameter specification and default values

doc Binomial mixture

hyper

theta1

hyperid 56551  
name beta1  
short.name beta1  
output.name beta1 for binomialmix observations  
output.name.intern beta1 for binomialmix observations  
initial 0  
fixed FALSE  
prior normal  
param 0 100  
to.theta function(x) x  
from.theta function(x) x

theta2

hyperid 56552  
name beta2  
short.name beta2  
output.name beta2 for binomialmix observations  
output.name.intern beta2 for binomialmix observations  
initial 0  
fixed FALSE  
prior normal  
param 0 100  
to.theta function(x) x  
from.theta function(x) x

theta3

hyperid 56553  
name beta3  
short.name beta3  
output.name beta3 for binomialmix observations  
output.name.intern beta3 for binomialmix observations  
initial 0  
fixed FALSE  
prior normal  
param 0 100  
to.theta function(x) x  
from.theta function(x) x

theta4

hyperid 56554  
name beta4  
short.name beta4  
output.name beta4 for binomialmix observations  
output.name.intern beta4 for binomialmix observations  
initial 0  
fixed FALSE  
prior normal

```

    param 0 100
    to.theta function(x) x
    from.theta function(x) x
theta5
    hyperid 56555
    name beta5
    short.name beta5
    output.name beta5 for binomialmix observations
    output.name.intern beta5 for binomialmix observations
    initial 0
    fixed FALSE
    prior normal
    param 0 100
    to.theta function(x) x
    from.theta function(x) x
theta6
    hyperid 56556
    name beta6
    short.name beta6
    output.name beta6 for binomialmix observations
    output.name.intern beta6 for binomialmix observations
    initial 0
    fixed FALSE
    prior normal
    param 0 100
    to.theta function(x) x
    from.theta function(x) x
theta7
    hyperid 56557
    name beta7
    short.name beta7
    output.name beta7 for binomialmix observations
    output.name.intern beta7 for binomialmix observations
    initial 0
    fixed FALSE
    prior normal
    param 0 100
    to.theta function(x) x
    from.theta function(x) x
theta8
    hyperid 56558
    name beta8
    short.name beta8
    output.name beta8 for binomialmix observations
    output.name.intern beta8 for binomialmix observations
    initial 0
    fixed FALSE
    prior normal
    param 0 100

```

```
to.theta function(x) x
from.theta function(x) x
theta9
  hyperid 56559
  name beta9
  short.name beta9
  output.name beta9 for binomialmix observations
  output.name.intern beta9 for binomialmix observations
  initial 0
  fixed FALSE
  prior normal
  param 0 100
  to.theta function(x) x
  from.theta function(x) x
status experimental
survival FALSE
discrete TRUE
link default logit probit
pdf binomialmix
```

## Example

Here is a simple example.

```
n <- 10^5
## high value of 'size' makes it easy to check the likelihood implementation
size <- sample(100:200, n, replace = TRUE)

beta.p1 <- rnorm(4, sd = 0.5)
beta.p2 <- rnorm(4, sd = 0.5)
beta9 <- rnorm(1, sd = 0.5)
beta <- c(beta.p1, beta.p2, beta9)

Z <- matrix(NA, n, 11)
W <- matrix(NA, n, 2)
Y <- matrix(NA, n, 2)

x <- rnorm(n, sd = 0.5)
xx <- rnorm(n, sd = 0.5)
eta <- numeric(n)

for (i in 1:n) {
  Z[i, ] <- rnorm(11)
  w <- c(rbeta(2, 1, 10), rbeta(1, 10, 1))
  w <- w/sum(w)
  W[i, ] <- w[1:2]

  p1 <- inla.link.invlogit(sum(beta.p1 * Z[i, 1:4]) + beta9 * Z[i, 9])
  p2 <- inla.link.invlogit(sum(beta.p2 * Z[i, 4 + 1:4]) + beta9 * Z[i, 10])

  eta[i] <- 1 + x[i] + xx[i] + beta9 * Z[i, 11]
  p3 <- inla.link.invlogit(eta[i])

  p <- w[1] * p1 + w[2] * p2 + w[3] * p3
  Y[i, ] <- c(rbinom(1, size = size[i], prob = p), size[i])
}

r <- inla(inla.mdata(Y, Z, W) ~ 1 + x + xx,
  family = "binomialmix",
  data = list(Y = Y, Z = Z, W = W, x = x, xx = xx),
  verbose = TRUE,
  control.inla = list(int.strategy = "eb"))

print(round(dig = 4, cbind(estimate = r$summary.fixed[, "mean"], true = 1)))
print(round(dig = 4, cbind(estimate = r$summary.hyperpar[, "mean"], true = beta)))
```