Zero-inflated models: Poisson, Binomial, negative Binomial and BetaBinomial

Parametrisation

There is support two types of zero-inflated models, which we name type 0 and type 1. These are defined for both the Binomial, the Poisson, the censored Poisson, the negative Binomial and BetaBinomial likelihood. For simplicity we will describe only the Poisson as the other cases are similar.

Type 0

The (type 0) likelihood is defined as

$$\text{Prob}(y \mid \ldots) = p \times 1_{[y=0]} + (1-p) \times \text{Poisson}(y \mid y > 0)$$

where p is a hyperparameter where

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

and θ is the internal representation of p; meaning that the initial value and prior is given for θ. This is model is called `zeroinflatedpoisson0` (and `zeroinflatedbinomial0`).

Type 1

The (type 1) likelihood is defined as

$$\text{Prob}(y \mid \ldots) = p \times 1_{[y=0]} + (1-p) \times \text{Poisson}(y)$$

where p is a hyperparameter where

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

and θ is the internal representation of p; meaning that the initial value and prior is given for θ. This is model is called `zeroinflatedpoisson1` (and `zeroinflatedbinomial1`).

Link-function

As for the Poisson, the Binomial the negative Binomial and the BetaBinomial.

Hyperparameters

For Poisson and the Binomial, there is one hyperparameter; where

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

and the prior and initial value is is given for θ.

For the negative Binomial and BetaBinomial, there are two hyperparameters. The overdispersion parameter n for the negative Binomial is represented as

$$\theta_1 = \log(n)$$

and the prior is defined on θ_1. The zero-inflation parameter p, is represented as

$$p = \frac{\exp(\theta_2)}{1 + \exp(\theta_2)}$$

and the prior and initial value is is given for θ_2. For the BetaBinomial it is similar.

Specification

- family = `zeroinflatedbinomial0`

- family = `zeroinflatedbinomial1`

- family = `zeroinflatednbinomial0`

- family = `zeroinflatednbinomial1`

- family = `zeroinflatedpoisson0`

- family = `zeroinflatedpoisson1`

- family = `zeroinflatedcenpoisson0`

- family = `zeroinflatedcenpoisson1`

- family = `zeroinflatedbetabinomial0`

- family = `zeroinflatedbetabinomial1`

- Required arguments: As for the Binomial, the negative Binomial, BetaBinomial and Poisson likelihood.

Hyperparameter spesification and default values

Zeroinflated Binomial Type 0

doc Zero-inflated Binomial, type 0

hyper

> **theta**
>
>> **hyperid** 90001
>> **name** logit probability
>> **short.name** prob
>> **initial** -1
>> **fixed** FALSE
>> **prior** gaussian
>> **param** -1 0.2
>> **to.theta** function(x) log(x / (1 - x))
>> **from.theta** function(x) exp(x) / (1 + exp(x))

survival FALSE

discrete FALSE

link default logit loga cauchit probit cloglog loglog robit sn

pdf zeroinflated

Zeroinflated Binomial Type 1

doc Zero-inflated Binomial, type 1

hyper

    theta

        hyperid 91001
        name logit probability
        short.name prob
        initial -1
        fixed FALSE
        prior gaussian
        param -1 0.2
        to.theta `function(x) log(x / (1 - x))`
        from.theta `function(x) exp(x) / (1 + exp(x))`

survival FALSE

discrete FALSE

link default logit loga cauchit probit cloglog loglog robit sn

pdf zeroinflated

Zeroinflated NegBinomial Type 0

doc Zero inflated negBinomial, type 0

hyper

    theta1

        hyperid 95001
        name log size
        short.name size
        initial 2.30258509299405
        fixed FALSE
        prior pc.mgamma
        param 7
        to.theta `function(x) log(x)`
        from.theta `function(x) exp(x)`

    theta2

        hyperid 95002
        name logit probability
        short.name prob
        initial -1
        fixed FALSE
        prior gaussian
        param -1 0.2
        to.theta `function(x) log(x / (1 - x))`

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

survival FALSE

discrete FALSE

link default log

pdf zeroinflated

Zeroinflated NegBinomial Type 1

doc Zero inflated negBinomial, type 1

hyper

> **theta1**
>
>> **hyperid** 96001
>> **name** log size
>> **short.name** size
>> **initial** 2.30258509299405
>> **fixed** FALSE
>> **prior** pc.mgamma
>> **param** 7
>> **to.theta** `function(x) log(x)`
>> **from.theta** `function(x) exp(x)`
>
> **theta2**
>
>> **hyperid** 96002
>> **name** logit probability
>> **short.name** prob
>> **initial** -1
>> **fixed** FALSE
>> **prior** gaussian
>> **param** -1 0.2
>> **to.theta** `function(x) log(x / (1 - x))`
>> **from.theta** `function(x) exp(x) / (1 + exp(x))`

survival FALSE

discrete FALSE

link default log

pdf zeroinflated

Zeroinflated BetaBinomial Type 0

doc Zero-inflated Beta-Binomial, type 0

hyper

> **theta1**

hyperid 88001

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))

theta2

hyperid 88002

name logit probability

short.name prob

initial -1

fixed FALSE

prior gaussian

param -1 0.2

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 zeroinflated

Zeroinflated BetaBinomial Type 1

doc Zero-inflated Beta-Binomial, type 1

hyper

theta1

hyperid 89001

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))

theta2

hyperid 89002

name logit probability

short.name prob

initial -1

fixed FALSE

prior gaussian

param -1 0.2

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 zeroinflated

Zeroinflated Poisson Type 0

doc Zero-inflated Poisson, type 0

hyper

> **theta**
>
> > **hyperid** 85001
> >
> > **name** logit probability
> >
> > **short.name** prob
> >
> > **initial** -1
> >
> > **fixed** FALSE
> >
> > **prior** gaussian
> >
> > **param** -1 0.2
> >
> > **to.theta** function(x) log(x / (1 - x))
> >
> > **from.theta** function(x) exp(x) / (1 + exp(x))

survival FALSE

discrete FALSE

link default log

pdf zeroinflated

Zeroinflated Poisson Type 1

doc Zero-inflated Poisson, type 1

hyper

> **theta**
>
> > **hyperid** 86001
> >
> > **name** logit probability
> >
> > **short.name** prob
> >
> > **initial** -1
> >
> > **fixed** FALSE
> >
> > **prior** gaussian

      param -1 0.2

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

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

survival FALSE

discrete FALSE

link default log

pdf zeroinflated

Zeroinflated Censored Poisson Type 0

doc Zero-inflated censored Poisson, type 0

hyper

    theta

      hyperid 87101

      name logit probability

      short.name prob

      initial -1

      fixed FALSE

      prior gaussian

      param -1 0.2

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

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

status experimental

survival FALSE

discrete FALSE

link default log

pdf zeroinflated

Zeroinflated Censored Poisson Type 1

doc Zero-inflated censored Poisson, type 1

hyper

    theta

      hyperid 87201

      name logit probability

      short.name prob

      initial -1

      fixed FALSE

      prior gaussian

      param -1 0.2

```
        to.theta function(x) log(x / (1 - x))
        from.theta function(x) exp(x) / (1 + exp(x))
```

status experimental

survival FALSE

discrete FALSE

link default log

pdf zeroinflated

Example

In the following example we estimate the parameters in a simulated example for both type 0 and type 1.

Poisson

```
## type 0
n=100
a = 1
b = 1
z = rnorm(n)
eta = a + b*z
p = 0.2
E = sample(c(1,5,10,15), size=n, replace=TRUE)
lambda = E*exp(eta)

## first sample y|y>0
y = rpois(n, lambda = lambda)
is.zero = (y == 0)
while(sum(is.zero) > 0)
{
    y[is.zero] = rpois(sum(is.zero), lambda[is.zero])
    is.zero = (y == 0)
}
## then set some of these to zero
y[ rbinom(n, size=1, prob=p) == 1 ] = 0

data = list(y=y,z=z)
formula = y ~ 1+z
result0 = inla(formula, family = "zeroinflatedpoisson0", data = data, E=E)
summary(result0)

## type 1
y = rpois(n, lambda = lambda)
y[ rbinom(n, size=1, prob=p) == 1 ] = 0
data = list(y=y,z=z)
formula = y ~ 1+z
result1 = inla(formula, family = "zeroinflatedpoisson1", data = data, E=E)
summary(result1)
```

Binomial

```
## type 0
n=100
a = 1
b = 1
z = rnorm(n)
eta = a + b*z
p = 0.2
Ntrials = sample(c(1,5,10,15), size=n, replace=TRUE)
prob = exp(eta)/(1 + exp(eta))

y = rbinom(n, size = Ntrials, prob = prob)
is.zero = (y == 0)
while(sum(is.zero) > 0)
{
    y[is.zero] = rbinom(sum(is.zero), size = Ntrials[is.zero], prob = prob[is.zero])
    is.zero = (y == 0)
}
y[ rbinom(n, size=1, prob=p) == 1 ] = 0
data = list(y=y,z=z)
formula = y ~ 1+z
result0 = inla(formula, family = "zeroinflatedbinomial0", data = data, Ntrials = Ntrials)
summary(result0)


## type 1
y = rbinom(n, size = Ntrials, prob = prob)
y[ rbinom(n, size=1, prob=p) == 1 ] = 0
data = list(y=y,z=z)
formula = y ~ 1+z
result1 = inla(formula, family = "zeroinflatedbinomial1", data = data, Ntrials=Ntrials)
summary(result1)
```

Advanced example

In the following example we estimate the parameters in a simulated example for a type0 likelihood, where one linear predictor enters the zero-probability and one other linear predictor enters the non-zero Poisson for example. The same trick can be used for other models of type0. The trick is that the likelihood

$$p^* 1_{[y=0]} + (1 - p^*)P(y|y > 0)$$

can be reformulated as a Bernoulli likelihood for the "class"-variable

$$z = \begin{cases} 1, & \text{if } y = 0 \\ 0, & \text{if } y > 0. \end{cases}$$

where p^* is the probability for success, and zero-inflated type0 likelihood (with fixed $p = 0$) for those $y > 0$. Since p^* and the linear predictor in P is separated into two likelihoods, we can apply one linear predictor to each one, hence extend the basic model to cases where p^* also depends on a linear predictor. Here is a small simulated example doing this.

```
require(INLA)


n = 100
a = 0.5
b = 1.5
x1 = rnorm(n, sd = 0.5)


eta.z = -a - b*x1
z = rbinom(n, 1, inla.link.logit(eta.z, inverse=TRUE))
n.y = sum(z)


x2 = rnorm(n.y, sd = 0.5)
eta.y = a + b*x2
lambda = exp(eta.y)
y = rpois(n.y, lambda)


is.zero = (y == 0)
while(sum(is.zero) > 0)
{
   y[is.zero] = rpois(sum(is.zero), lambda[is.zero])
   is.zero = (y == 0)
}


Y = matrix(NA, n + n.y, 2)
Y[1:n, 1] = z
Y[n + 1:n.y, 2] = y


form = Y ~ 0 + mu.z + mu.y + cov.z + cov.y
ldat = list(
        Y=Y,
        mu.z=rep(1:0, c(n, n.y)),
        mu.y=rep(0:1, c(n, n.y)),
        cov.z=c(x1, rep(NA,n.y)),
        cov.y=c(rep(NA, n), x2))
```

```
res <- inla(form, data=ldat,
            family=c('binomial', 'zeroinflatedpoisson0'),
            control.family=list(
                    list(),
                    list(hyper = list(
                                prob = list(
                                        initial = -20,
                                        fixed = TRUE)))))
round(res$summary.fix, 4)
```

Notes

None.

Extentions

There are some extentions available which currently is only implemented for the cases where its needed/requested.

Type 2 Is like Type 1 but where (for the Poisson)

$$p = 1 - \left(\frac{E \exp(x)}{1 + E \exp(x)} \right)^{\alpha}$$

where $\alpha > 0$ is the hyperparameter instead of p (and $E \exp(x)$ is the mean). Available for Poisson as `zeroinflatedpoisson2`, for binomial as `zeroinflatedbinomial2` and for the negative binomial as `zeroinflatednbinomial2`.

The internal representation is $\theta = \log(\alpha)$ and prior is defined on $\log(\alpha)$.

Zeroinflated Poisson Type 2

doc Zero-inflated Poisson, type 2

hyper

    theta

        hyperid 87001
        name log alpha
        short.name a
        initial 0.693147180559945
        fixed FALSE
        prior gaussian
        param 0.693147180559945 1
        to.theta function(x) log(x)
        from.theta function(x) exp(x)

survival FALSE

discrete FALSE

link default log

pdf zeroinflated

Zeroinflated Binomial Type 2

doc Zero-inflated Binomial, type 2

hyper

    theta

        hyperid 92001
        name alpha
        short.name alpha
        initial -1
        fixed FALSE
        prior gaussian
        param -1 0.2
        to.theta function(x) log(x)
        from.theta function(x) exp(x)

survival FALSE

discrete FALSE

link default logit loga cauchit probit cloglog loglog robit sn

pdf zeroinflated

Zeroinflated Negative Binomial Type 2

doc Zero inflated negBinomial, type 2

hyper

    theta1

        hyperid 99001
        name log size
        short.name size
        initial 2.30258509299405
        fixed FALSE
        prior pc.mgamma
        param 7
        to.theta function(x) log(x)
        from.theta function(x) exp(x)

    theta2

        hyperid 99002
        name log alpha
        short.name a
        initial 0.693147180559945
        fixed FALSE
        prior gaussian
        param 2 1
        to.theta function(x) log(x)

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

survival FALSE

discrete FALSE

link default log

pdf zeroinflated

Zeroinflated Negative Binomial Type 1 Strata 2

doc Zero inflated negBinomial, type 1, strata 2

hyper

> **theta1**
>
> > **hyperid** 97001
> > **name** log size
> > **short.name** size
> > **initial** 2.30258509299405
> > **fixed** FALSE
> > **prior** pc.mgamma
> > **param** 7
> > **to.theta** `function(x) log(x)`
> > **from.theta** `function(x) exp(x)`
>
> **theta2**
>
> > **hyperid** 97002
> > **name** logit probability 1
> > **short.name** prob1
> > **initial** -1
> > **fixed** FALSE
> > **prior** gaussian
> > **param** -1 0.2
> > **to.theta** `function(x) log(x / (1 - x))`
> > **from.theta** `function(x) exp(x) / (1 + exp(x))`
>
> **theta3**
>
> > **hyperid** 97003
> > **name** logit probability 2
> > **short.name** prob2
> > **initial** -1
> > **fixed** FALSE
> > **prior** gaussian
> > **param** -1 0.2
> > **to.theta** `function(x) log(x / (1 - x))`
> > **from.theta** `function(x) exp(x) / (1 + exp(x))`
>
> **theta4**
>
> > **hyperid** 97004

13

name logit probability 3

short.name prob3

initial -1

fixed TRUE

prior gaussian

param -1 0.2

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

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

theta5

    hyperid 97005

    name logit probability 4

    short.name prob4

    initial -1

    fixed TRUE

    prior gaussian

    param -1 0.2

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

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

theta6

    hyperid 97006

    name logit probability 5

    short.name prob5

    initial -1

    fixed TRUE

    prior gaussian

    param -1 0.2

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

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

theta7

    hyperid 97007

    name logit probability 6

    short.name prob6

    initial -1

    fixed TRUE

    prior gaussian

    param -1 0.2

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

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

theta8

    hyperid 97008

    name logit probability 7

    short.name prob7

    initial -1

    fixed TRUE

prior gaussian

param -1 0.2

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

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

theta9

hyperid 97009

name logit probability 8

short.name prob8

initial -1

fixed TRUE

prior gaussian

param -1 0.2

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

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

theta10

hyperid 97010

name logit probability 9

short.name prob9

initial -1

fixed TRUE

prior gaussian

param -1 0.2

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

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

theta11

hyperid 97011

name logit probability 10

short.name prob10

initial -1

fixed TRUE

prior gaussian

param -1 0.2

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

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

status experimental

survival FALSE

discrete FALSE

link default log

pdf zeroinflated

Zeroinflated Negative Binomial Type 1 Strata 3

doc Zero inflated negBinomial, type 1, strata 3

hyper

    theta1

        hyperid 98001
        name logit probability
        short.name prob
        initial -1
        fixed FALSE
        prior gaussian
        param -1 0.2
        to.theta `function(x) log(x / (1 - x))`
        from.theta `function(x) exp(x) / (1 + exp(x))`

    theta2

        hyperid 98002
        name log size 1
        short.name size1
        initial 2.30258509299405
        fixed FALSE
        prior pc.mgamma
        param 7
        to.theta `function(x) log(x)`
        from.theta `function(x) exp(x)`

    theta3

        hyperid 98003
        name log size 2
        short.name size2
        initial 2.30258509299405
        fixed FALSE
        prior pc.mgamma
        param 7
        to.theta `function(x) log(x)`
        from.theta `function(x) exp(x)`

    theta4

        hyperid 98004
        name log size 3
        short.name size3
        initial 2.30258509299405
        fixed TRUE
        prior pc.mgamma
        param 7
        to.theta `function(x) log(x)`
        from.theta `function(x) exp(x)`

theta5

    hyperid 98005

    name log size 4

    short.name size4

    initial 2.30258509299405

    fixed TRUE

    prior pc.mgamma

    param 7

    to.theta function(x) log(x)

    from.theta function(x) exp(x)

theta6

    hyperid 98006

    name log size 5

    short.name size5

    initial 2.30258509299405

    fixed TRUE

    prior pc.mgamma

    param 7

    to.theta function(x) log(x)

    from.theta function(x) exp(x)

theta7

    hyperid 98007

    name log size 6

    short.name size6

    initial 2.30258509299405

    fixed TRUE

    prior pc.mgamma

    param 7

    to.theta function(x) log(x)

    from.theta function(x) exp(x)

theta8

    hyperid 98008

    name log size 7

    short.name size7

    initial 2.30258509299405

    fixed TRUE

    prior pc.mgamma

    param 7

    to.theta function(x) log(x)

    from.theta function(x) exp(x)

theta9

    hyperid 98009

    name log size 8

    short.name size8

initial 2.30258509299405

fixed TRUE

prior pc.mgamma

param 7

to.theta function(x) log(x)

from.theta function(x) exp(x)

theta10

hyperid 98010

name log size 9

short.name size9

initial 2.30258509299405

fixed TRUE

prior pc.mgamma

param 7

to.theta function(x) log(x)

from.theta function(x) exp(x)

theta11

hyperid 98011

name log size 10

short.name size10

initial 2.30258509299405

fixed TRUE

prior pc.mgamma

param 7

to.theta function(x) log(x)

from.theta function(x) exp(x)

status experimental

survival FALSE

discrete FALSE

link default log

pdf zeroinflated

0.0.1 Zero and N-inflated Binomial likelihood: type 3

This is the case where

$$
\begin{aligned}
\mathrm{Prob}(y \,|\, \ldots) \;=\; & p_0 \times 1_{[y=0]} + \\
& p_N \times 1_{[y=N]} + \\
& (1 - p_0 - p_N) \times \mathrm{binomial}(y, N, p)
\end{aligned}
$$

where:

$$
p = \frac{\exp(\eta)}{1 + \exp(\eta)} \qquad p_0 = \frac{p^{\alpha_0}}{1 + p^{\alpha_0} + (1-p)^{\alpha_N}} \qquad p_N = \frac{(1-p)^{\alpha_N}}{1 + p^{\alpha_0} + (1-p)^{\alpha_N}}
$$

There are 2 hyperparameters, α_0 and α_N, governing zero-inflation where: The zero-inflation parameters α_0 and α_N are represented as $\theta_0 = \log(\alpha_0)$; $\theta_N = \log(\alpha_N)$ and the prior and initial value is given for θ_0 and θ_N respectively.

Here is an example

18

```
nsim<-10000
x<-rnorm(nsim)
alpha0<-1.5
alphaN<-2.0
p = exp(x)/(1+exp(x))
p0 = p^alpha0 / (1 + p^alpha0 + (1-p)^alphaN)
pN = (1-p)^alphaN / (1 + p^alpha0 + (1-p)^alphaN)
P<-cbind(p0, pN, (1-p0 -pN))
N<-rpois(nsim,20)
y<-rep(0,nsim)
for(i in 1:nsim)
    y[i]<-sum(rmultinom(1,size = 1,P[i,])*c(0,N[i],rbinom(1,N[i],p[i])))
formula = y ~1 + x
r = inla(formula, family = "zeroninflatedbinomial3",  Ntrials = N, verbose = TRUE,
         data = data.frame(y, x))
```

and the default settings

doc Zero and N inflated binomial, type 3

hyper

    theta1

        hyperid 93101

        name alpha0

        short.name alpha0

        initial 1

        fixed FALSE

        prior loggamma

        param 1 1

        to.theta function(x) log(x)

        from.theta function(x) exp(x)

    theta2

        hyperid 93102

        name alphaN

        short.name alphaN

        initial 1

        fixed FALSE

        prior loggamma

        param 1 1

        to.theta function(x) log(x)

        from.theta function(x) exp(x)

status experimental

survival FALSE

discrete FALSE

link default logit loga cauchit probit cloglog loglog robit sn

pdf zeroinflated