

## log1exp effect of a covariate

### Parametrization

This model implements a non-linear effect of a positive covariate  $x$  as a part of the linear predictor,

$$\beta \log(1 + \exp(\alpha - \gamma x))$$

where  $\beta, \alpha, \gamma \in \Re$  and  $x \geq 0$ .

### Hyperparameters

This model has three hyperparameters, the scaling  $\beta$ , halflife  $a$  and shape  $k$ ,

$$\theta_1 = \beta \quad \theta_2 = \alpha \quad \theta_3 = \gamma$$

and the priors are given for  $\theta_1, \theta_2$  and  $\theta_3$ .

### Specification

```
f(x, model="log1exp", hyper = ..., precision = <precision>)
```

where `precision` is the precision for the tiny noise used to implement this as a latent model.

### Hyperparameter specification and default values

`hyper`

`theta1`

```
hyperid 39011
name beta
short.name b
initial 1
fixed FALSE
prior normal
param 0 1
to.theta function(x) x
from.theta function(x) x
```

`theta2`

```
hyperid 39012
name alpha
short.name a
initial 0
fixed FALSE
prior normal
param 0 1
to.theta function(x) x
from.theta function(x) x
```

`theta3`

```
hyperid 39013
```

```

    name gamma
    short.name g
    initial 0
    fixed FALSE
    prior normal
    param 0 1
    to.theta function(x) x
    from.theta function(x) x

constr FALSE

nrow.ncol FALSE

augmented FALSE

aug.factor 1

aug.constr

n.div.by

n.required FALSE

set.default.values FALSE

status experimental

pdf log1exp

```

## Example

```

log1exp = function(x, beta, alpha, gamma)
{
  return (beta * log(1.0 + exp(alpha - gamma * x)))
}

n = 100
lambda = 2
s=0.1
x = rpois(n, lambda = lambda)
beta = 1
alpha = 0
gamma = .5

y = log1exp(x, beta, alpha, gamma) + rnorm(n, sd = s)
r = inla(y ~ -1 + f(x, model="log1exp"),
  data = data.frame(y, x),
  family = "gaussian",
  control.inla = list(h=0.001),
  control.family = list(
    hyper = list(
      prec = list(
        initial = log(1/s^2),
        fixed = TRUE))))
summary(r)

```

## Notes

None