

## logdist effect of a covariate

### Parametrization

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

$$\beta (1 + \exp(\alpha_1 \log(x) - \alpha_2 x))$$

where  $\beta \in \Re$ ,  $\alpha_1, \alpha_2 \in \Re^+$  and  $x \geq 0$ .

### Hyperparameters

This model has three hyperparameters, the scaling  $\beta$ ,  $\alpha_1$  and  $\alpha_2$ .

$$\theta_1 = \beta \quad \theta_2 = \log(\alpha_1) \quad \theta_3 = \log(\alpha_2)$$

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

### Specification

```
f(x, model="logdist", 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

doc A nonlinear model of a covariate

hyper

theta1

```
hyperid 39021
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 39022
name alpha1
short.name a1
initial 0
fixed FALSE
prior loggamma
param 0.1 1
to.theta function(x) log(x)
from.theta function(x) exp(x)
```

theta3

```

    hyperid 39023
    name alpha2
    short.name a2
    initial 0
    fixed FALSE
    prior loggamma
    param 0.1 1
    to.theta function(x) log(x)
    from.theta function(x) exp(x)

```

```
constr FALSE
```

```
nrow.ncol FALSE
```

```
augmented FALSE
```

```
aug.factor 1
```

```
aug.constr
```

```
n.div.by
```

```
n.required FALSE
```

```
set.default.values FALSE
```

```
pdf logdist
```

## Example

```

logdist = function(x, beta, alpha)
{
  return (beta * (1 + exp(alpha[1] * log(x) - alpha[2] * x)))
}

```

```
n = 1000
```

```
s=0.1
```

```
x = runif(n)
```

```
beta = 1
```

```
alpha = c(1, 0.5)
```

```
## start at the true values
```

```

hyper = list(
  beta = list(initial = beta),
  a1 = list(initial = log(alpha[1])),
  a2 = list(initial = log(alpha[2]))
)

```

```
## start somewhere else
```

```

hyper = list(
  beta = list(initial = 1),
  a1 = list(initial = 0),
  a2 = list(initial = 0)
)

```

```
y = logdist(x, beta, alpha) + rnorm(n, sd = s)
```

```
r = (inla(y ~ -1 + f(x, model="logdist", hyper = hyper),
```

```
data = data.frame(y, x),
family = "gaussian",
verbose=TRUE,
control.inla = list(h=0.0001),
control.family = list(
  hyper = list(
    prec = list(
      initial = log(1/s^2),
      fixed = TRUE))))))
summary(r)
```

## Notes

None