Weibull

Parametrisation

The Weibull distribution is (`variant=0`)

$$f(y) = \alpha y^{\alpha-1} \lambda \exp(-\lambda y^{\alpha}), \qquad \alpha > 0, \qquad \lambda > 0$$

and (`variant=1`)

$$f(y) = \alpha y^{\alpha-1} \lambda^{\alpha} \exp(-(\lambda y)^{\alpha}), \qquad \alpha > 0, \qquad \lambda > 0$$

where

α**:** shape parameter.

Link-function

The parameter λ is linked to the linear predictor as:

$$\lambda = \exp(\eta)$$

Hyperparameters

The α parameter is represented as

$$\alpha = \exp(S\theta)$$

and the prior is defined on θ. The constant S currently set to 0.1 to avoid numerical instabilities in the optimization, since small changes of α can make a huge difference.

Specification

- family = `weibull` for regression and family = `weibullsurv` for survival
- Required arguments: y (to be given using `inla.surv()` for survival models), and `variant=0` (default) or 1 to define the parameterisation.

Hyperparameter spesification and default values

weibull

doc The Weibull likelihood
hyper
theta
hyperid 79001
name log alpha
short.name alpha
initial -2
fixed FALSE
prior pc.alphaw
param 5
to.theta `function(x, sc = 0.1) log(x) / sc`
from.theta `function(x, sc = 0.1) exp(sc * x)`
survival FALSE
discrete FALSE
link default log neglog quantile
pdf weibull

weibullsurv

doc The Weibull likelihood (survival)

hyper

theta

hyperid 79101
name log alpha
short.name alpha
initial -2
fixed FALSE
prior pc.alphaw
param 5
to.theta `function(x, sc = 0.1) log(x) / sc`
from.theta `function(x, sc = 0.1) exp(sc * x)`

theta2

hyperid 79102
name beta1
short.name beta1
initial -7
fixed FALSE
prior normal
param -4 100
to.theta `function(x) x`
from.theta `function(x) x`

theta3

hyperid 79103
name beta2
short.name beta2
initial 0
fixed FALSE
prior normal
param 0 100
to.theta `function(x) x`
from.theta `function(x) x`

theta4

hyperid 79104
name beta3
short.name beta3
initial 0
fixed FALSE
prior normal
param 0 100
to.theta `function(x) x`
from.theta `function(x) x`

theta5

hyperid 79105
name beta4
short.name beta4
initial 0
fixed FALSE
prior normal
param 0 100
to.theta `function(x) x`
from.theta `function(x) x`

theta6
hyperid 79106
name beta5
short.name beta5
initial 0
fixed FALSE
prior normal
param 0 100
to.theta `function(x) x`
from.theta `function(x) x`

theta7
hyperid 79107
name beta6
short.name beta6
initial 0
fixed FALSE
prior normal
param 0 100
to.theta `function(x) x`
from.theta `function(x) x`

theta8
hyperid 79108
name beta7
short.name beta7
initial 0
fixed FALSE
prior normal
param 0 100
to.theta `function(x) x`
from.theta `function(x) x`

theta9
hyperid 79109
name beta8
short.name beta8
initial 0
fixed FALSE

prior normal
param 0 100
to.theta `function(x) x`
from.theta `function(x) x`

theta10
hyperid 79110
name beta9
short.name beta9
initial 0
fixed FALSE
prior normal
param 0 100
to.theta `function(x) x`
from.theta `function(x) x`

theta11
hyperid 79111
name beta10
short.name beta10
initial 0
fixed FALSE
prior normal
param 0 100
to.theta `function(x) x`
from.theta `function(x) x`

survival TRUE

discrete FALSE

link default log neglog quantile

pdf weibull

Example

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

```
n = 1000
alpha = 1.1
beta = 2.2
x = c(scale(runif(n)))
eta = 1+beta*x
lambda = exp(eta)

for(variant in 0:1) {
    y = rweibull(n,
                 shape= alpha,
                 scale= if (variant == 0)
                           lambda^(-1/alpha)
                       else
                           1/lambda)
```

```
    print(paste("VARIANT=", variant))
    event = rep(1,n)
    data = list(y=y, event=event, x=x)

    formula=inla.surv(y,event)~ x
    r=inla(formula,
           family ="weibullsurv",
           data=data,
           control.family = list(list(variant = variant)))
    print("SURV")
    print(summary(r))

    formula= y ~ x
    r=inla(formula,
           family ="weibull",
           data=data,
           control.family = list(list(variant = variant)))
    print("REGRESSION")
    print(summary(r))
}
```

Notes

- Weibullsurv model can be used for right censored, left censored, interval censored data. If the observed times y are large/huge, then this can cause numerical overflow in the likelihood routine. If you encounter this problem, try to scale the observatios, `time = time / max(time)` or similar.