The Gamma-distribution

Parametrisation

The Gamma-distribution has the following density

$$\pi(y) = \frac{b^a}{\Gamma(a)} y^{a-1} \exp(-by), \qquad a > 0, \quad b > 0, \quad y > 0,$$

where $\mathrm{E}(y) = \mu = a/b$ and $\mathrm{Var}(y) = 1/\tau = a/b^2$, where τ is the precision and μ is the mean. We will use the following parameterisation for the precision

$$\tau = (s\phi)/\mu^2$$

where ϕ is the precision parameter (or $1/\phi$ is the dispersion parameter) and $s > 0$ is a fixed scaling, which gives this density

$$\pi(y) = \frac{1}{\Gamma(s\phi)} \left(\frac{(s\phi)}{\mu}\right)^{(s\phi)} y^{(s\phi)-1} \exp\left(-(s\phi)\frac{y}{\mu}\right)$$

Link-function

The linear predictor η is linked to the mean μ using a default log-link

$$\mu = \exp(\eta)$$

Hyperparameter

The hyperparameter is the precision parameter ϕ, which is represented as

$$\phi = \exp(\theta)$$

and the prior is defined on θ.

Specification

- family = `gamma` for regression models and family = `gamma.surv` for survival models.
- Required arguments: for `gamma.surv`, y (to be given in a format by using `inla.surv()`), and for `gamma`, y and s.

The scalings have default value 1.

Hyperparameter spesification and default values

doc The Gamma likelihood

hyper

theta

hyperid 58001
name precision parameter
short.name prec
initial 4.60517018598809
fixed FALSE

prior loggamma
param 1 0.01
to.theta `function(x) log(x)`
from.theta `function(x) exp(x)`

survival FALSE

discrete FALSE

link default log quantile

pdf gamma

doc The Gamma likelihood (survival)

hyper

theta
hyperid 58101
name precision parameter
short.name prec
initial 0
fixed FALSE
prior loggamma
param 1 0.01
to.theta `function(x) log(x)`
from.theta `function(x) exp(x)`

survival TRUE

discrete FALSE

status experimental

link default log quantile

pdf gammasurv

Example

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

```
n = 1000
x = rnorm(n)
eta = 1 + x
mu = exp(eta)
prec.scale = runif(n, min = 0.5, max = 2)
prec.par = 1.2
a = prec.par * prec.scale
b = mu / (prec.par * prec.scale)
y = rgamma(n, shape = a, scale = b)
r = inla(y ~ 1 + x,  data = data.frame(y, x),
        scale = prec.scale, family = "gamma")
```

Notes

None.