

dGompertz

Parametrisation

The Gompertz distribution with cure fraction has log “survival function”

$$\log S(y) = -\frac{\mu}{\alpha} (\exp(\alpha y) - 1)$$

for response $y \geq 0$, $\mu > 0$. The “cumulative distribution function” and the “density” then follows as

$$F(y) = 1 - \exp \left[-\frac{\mu}{\alpha} (\exp(\alpha y) - 1) \right]$$

and

$$f(y) = \mu \exp \left[\alpha y - \frac{\mu}{\alpha} (\exp(\alpha y) - 1) \right].$$

Note that α is allowed to be negative making to allow for a cure fraction, hence $\lim_{y \rightarrow \infty} F(y) < 1$ for $\alpha < 0$.

Link-function

The parameter μ is linked to the linear predictor η as:

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

Hyperparameters

The shape parameter α is represented as

$$\alpha = \theta$$

and the prior is defined on θ .

Specification

- `family="dgomperzsurv"` for survival models
- Required arguments: y (to be given in a format by using `inla.surv()` for survival models)

Hyperparameter specification and default values

`doc destructive gomperz (survival) distribution`

`hyper`

`theta`

`hyperid 108101`

`name shape`

`short.name alpha`

`output.name.intern alpha_intern for dGompertz`

`output.name alpha parameter for dGompertz`

`initial -1`

`fixed FALSE`

`prior normal`

`param 0 10`

`to.theta function(x) x`

```
from.theta function(x) x
```

```
experimental TRUE
```

```
survival TRUE
```

```
discrete FALSE
```

```
link default log neglog
```

```
pdf dgomperz
```

Example

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

```
gen.cure.gompertz = function(n,a,b,p) {
  if (length(b) == 1) b <- rep(b, n)
  if (length(p) == 1) p <- rep(p, n)
  rm = rbinom(n=n,size=1,prob=1-p)
  t=rep(NA,n)
  for(i in 1:n){
    t[i]=ifelse(rm[i]==0,
               Inf,
               log((-a/b[i])*log(1-runif(n=1,min=0,max=1-p[i]))) + 1)*(1/a))
  }
  t_finite = ifelse(t==Inf,0,t)
  u2 = runif(n=n,0,max(t_finite))
  t2 = pmin(t,u2)
  delta = ifelse(t<u2,1,0)
  return(cbind(t2,delta))
}

n = 1000
a_par=-0.5 ##defective
x <- rnorm(n)
b_par <- exp(0.2 + x)
p_par = exp(b_par/a_par) ##cure fraction

data = gen.cure.gompertz(n=n,a=a_par,b=b_par,p=p_par)
colnames(data) = c("tempo", "delta")
data <- as.data.frame(data)

data$x <- x
r <- inla(inla.surv(tempo, delta) ~ 1 + x,
          data = data,
          family = "dgomperzsurv",
          control.inla = list(cmin = 0),
          verbose = TRUE)
summary(r)
```

Notes