Cox Proportional Hazards Model

Parametrisation

In the Cox proportional hazards model, defines the hazard rate as:

$$h(t) = h_0(t)\exp(\eta)$$

where

$h_0(\cdot)$**:** baseline hazard

η**:** linear predictor

We start from a finite partition of the time axis $0 = s_0 < s_1 < \ldots, s_K$ and assume the baseline hazard to be constant in each time interval

$$h_0(t) = \exp(b_k) \text{ for } t \in (s_{k-1}, s_k], \qquad k = 1, \ldots, K$$

and assign $\mathbf{b} = (b_1, \ldots, b_K)$ a Gaussian prior (RW1 or RW2) with unknown precision τ_b

Link-function

The parameter η is the linear predictor

Hyperparameters

The log precision $\log \tau_b$ for the piecewise constant hazard

Specification

- family = `coxph`
- Required arguments:
  - y (to be given in a format by using `inla.surv()` function)
  - `control.hazard = list()` to control the prior for the piecewise constant hazar, see `?control.hazard` for more information.

Example

In the following example we estimate the baseline hazard in a simulated case

```
n = 1000
alpha = 2
beta = 2
x = runif(n)
eta = 1+beta*x
lambda = exp(eta)
y = rweibull(n, shape= alpha, scale= lambda^(1/-alpha))
event = rep(1,n)
data = list(y=y, event=event, x=x)

formula=inla.surv(y,event)~ x

model=inla(formula, family ="coxph", data=data, verbose=T,
        control.hazard=list(model="rw1", n.intervals=20))
```

Notes

- The Cox model can be used only for uncensored or right censored data.
- The model for the piecewise constant baseline hazard is specified through `control.hazard`
- A general frame work to represent time is given by `inla.surv`
- If the observed times y are large/huge, then this can cause numerical overflow in the likelihood routines giving error messages like

```
file: smtp-taucs.c  hgid: 891deb69ae0c  date: Tue Nov 09 22:34:28 2010 +0100
Function: GMRFLib_build_sparse_matrix_TAUCS(), Line: 611, Thread: 0
Variable evaluates to NAN/INF. This does not make sense. Abort...
```

  If you encounter this problem, try to scale the observatios, `time = time / max(time)` or similar, before running `inla()`.