The Berkson Measurement Error (MEB) model

Parametrization

This is an implementation of the Berksom measurement error model for a fixed effect. The observed covariate is x but its $\widetilde{x}$ that goes into the linear predictor

$$\eta = \ldots + \beta\widetilde{x} + \ldots$$

where $\widetilde{x} = x + \epsilon$ where ϵ is Gaussian with precision τs where s is a vector of fixed scalings (default all are 1).

Hyperparameters

This model has 2 hyperparameters, $\theta = (\theta_1, \theta_2)$. The hyperparameter specification is as follows:

$$\theta_1 = \beta$$

and the prior is defined on θ_1,

$$\theta_2 = \log(\tau)$$

and the prior is defined on θ_2.

Specification

The MEB is specified inside the `f()` function as

```
f(x, [<weights>,] model="meb", hyper = <hyper>, scale = <s>)
```

Here, `x` are the observed covariates, and the fixed scaling of the observational precision is given in argument `scale`. If the argument `scale` is not given, then s is set to 1.

Note that only the unique values of `x` are used, so if two or more elements of `x` are *identical*, then they refer to the *same* element in the covariate $\widetilde{x}$.

Hyperparameter specification and default values

hyper

    theta1

        name beta

        short.name b

        prior gaussian

        param 1 0.001

        initial 1

        fixed FALSE

        to.theta `function(x) x`

        from.theta `function(x) x`

    theta2

        name prec.obs

        short.name prec

        prior loggamma

        param 1 1e-04

        initial 6.90775527898214

> **fixed** FALSE
>
> **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

status experimental

pdf meb

Example

```
n = 300
beta = 2
x = rnorm(n)
prec.x = 10
prec.y = 1000
s = runif(n)
x.tilde = x + rnorm(n, sd = 1/sqrt(s*prec.x))
y = 1 + beta * x.tilde + rnorm(n, sd = 1/sqrt(prec.y))

r = inla(y ~ f(x, model="meb", scale = s),
        family = "gaussian",
        data = data.frame(y, x, s))
```

Notes

- `INLA` provide the posterior of $\beta\tilde{x}$ and NOT $\tilde{x}$. The results comes in the order given by the sorted (from low to high) values of `x` and the field `ID` gives the mapping.