

The Berkson Measurement Error (MEB) model

Parametrization

This is an implementation of the Berkson measurement error model for a fixed effect. The observed covariate is w but it is x that goes into the linear predictor

$$\eta = \dots + \beta x + \dots ,$$

where $x = w + u$, u is Gaussian with precision $\tau_u \times s$, and s is a vector of fixed scalings.

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 ,

$$\theta_2 = \log(\tau_u)$$

and the prior is defined on θ_2 .

Specification

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

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

Here, `w` 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 `w` are used, so if two or more elements of `w` are *identical*, then they refer to the *same* element in the covariate 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.u
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 = 100
beta = 2
w = rnorm(n)
prec.u = 1
prec.y = 1
s = runif(n,min=0,max=1)
x = w + rnorm(n, sd = 1/sqrt(s*prec.u))
y = 1 + beta*x + rnorm(n, sd = 1/sqrt(prec.y))

## prior parameters
prior.beta = c(0, 0.0001)
prior.prec.u = c(10, 9/prec.u)
prior.prec.y = c(10, 9/prec.y)

formula = y ~ f(w,model="meb", scale=s,
               hyper = list(
                 beta = list(
                   param = prior.beta,
                   fixed = FALSE
                 ),
                 prec.u = list(
                   param = prior.prec.u,
                   fixed = FALSE
                 )
               )
)

r = inla(formula, data = data.frame(y, w, s),
         family = "gaussian",

```

```

control.family = list(
  hyper = list(
    prec = list(param = prior.prec.y,
                 fixed = FALSE
               )
          )
)

```

Notes

- INLA provide the posterior of $\nu = \beta x$ and NOT x . The results comes (default) in the order given by the sorted (from low to high) values of \mathbf{x} and the field ID gives the mapping.
- The option **scale** defines the scaling in the same order as argument **values**. It is therefore adviced to also give argument **values** when **scale** is used to be sure that they are consistent.