

# The Classical Measurement Error (MEC) model

## Parametrization

This is an implementation of the classical ME model for a fixed effect. It is best described by an example, let the model be

$$y = \beta x + \epsilon$$

where  $y$  is the response,  $\beta$  the effect of the true covariate  $x$  with zero mean Gaussian noise  $\epsilon$ . The issue is that  $x$  is not observed directly, but only through  $w$ , where

$$w = x + u$$

where  $u$  is zero mean Gaussian noise. Even though this setup is possible to implement using basic features ("copy" and multiple likelihoods), we provide the following model which replaces the above,

$$y = \nu + \epsilon$$

where  $\nu = \beta x$  has the correct distribution depending on various parameters:  $\beta$  has prior  $\pi(\beta)$ ,  $x$  is apriori  $\mathcal{N}(\mu_x I, \tau_x I)$ , and  $s \times \tau_u$  is the observation precision for  $x$  (i.e.,  $\text{Prec}(u|x)$ ).<sup>1</sup> Here,  $s$  is a vector of fixed scalings.

## Hyperparameters

This model has 4 hyperparameters,  $\theta = (\theta_1, \theta_2, \theta_3, \theta_4)$  where  $\theta_2$ ,  $\theta_3$  and  $\theta_4$  are default set to be fixed (ie defined to be known). The values of  $\theta_2$ ,  $\theta_3$  and  $\theta_4$  are set to mimic a classical fixed effect, so they will always make sense. To achieve the ME model, please use the appropriate choices for (some of) these parameters!

The hyperparameter specification is as follows:

$$\theta_1 = \beta$$

and the prior is defined on  $\theta_1$ ,

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

and the prior is defined on  $\theta_2$ ,

$$\theta_3 = \mu_x$$

and the prior is defined on  $\theta_3$ ,

$$\theta_4 = \log(\tau_x)$$

and the prior is defined on  $\theta_4$ .

## Specification

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

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

The `w` are the observed values of the true but unknown covariates  $x$ , with the *assumption*, that if two or more elements of `w` are *identical*, then they refer to the *same* element in the true covariate  $x$ . 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.

---

<sup>1</sup>Note: The second argument in  $\mathcal{N}(,)$  is the precision not the variance.

## 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 9.21034037197618  
fixed TRUE  
to.theta function(x) log(x)  
from.theta function(x) exp(x)

theta3

name mean.x  
short.name mu.x  
prior gaussian  
param 0 1e-04  
initial 0  
fixed TRUE  
to.theta function(x) x  
from.theta function(x) x

theta4

name prec.x  
short.name prec.x  
prior loggamma  
param 1 10000  
initial -9.21034037197618  
fixed TRUE  
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** mec

## Example

```
n = 100
beta = 4
prec.y = 1
prec.u = 1
prec.x = 1
## true unobserved covariate
x = rnorm(n, sd = 1/sqrt(prec.x))
## the observed covariate
s = runif(n,min=0.5,max=2)
w = x + rnorm(n, sd = 1/sqrt(prec.u*s))
## regression model using the unobserved 'x'
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)
prior.prec.x = c(10, 9)
prior.prec.y = c(10, 9)

formula = y ~ 1 +
  f(w, model="mec", scale=s,
    hyper = list(
      beta = list(
        param = prior.beta,
        fixed = FALSE
      ),
      prec.u = list(
        param = prior.prec.u,
        initial = log(prec.u),
        fixed = FALSE
      ),
      prec.x = list(
        param = prior.prec.x,
        initial = log(prec.x),
        fixed = FALSE
      ),
      mean.x = list(
        initial = 0,
```

```

                                fixed=TRUE
                                )
                            )
                    )

r = inla(formula,
        data = data.frame(y, w, s),
        family = "gaussian",
        control.family = list(
            hyper = list(
                prec = list(param = prior.prec.y,
                           initial = log(prec.y),
                           fixed=FALSE
                          )
            )
        )
    )

summary(r)

```

## Notes

- INLA provide the posterior of  $\nu$  and NOT  $x$ .
- The posterior of  $\nu$  comes (default) in the order given by the sorted (from low to high) values of  $w$ . The entry `$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.