Generic2 model

Parametrization

The generic model implements the following precision matrix

$$\mathbf{Q} = \begin{bmatrix} \tau_u \mathbf{I} & -\tau_u \mathbf{I} \\ -\tau_u \mathbf{I} & \tau_u \mathbf{I} + \tau_v \mathbf{C} \end{bmatrix}$$
 (1)

where C is (a given) symmetric matrix. This model arrives from the hierarchical model,

$$oldsymbol{v} \sim \mathcal{N}(oldsymbol{0}, au_v oldsymbol{C})$$

and

$$\boldsymbol{u} \mid \boldsymbol{v} \sim \mathcal{N}(\boldsymbol{v}, \tau_u \boldsymbol{I})$$

and the precision matrix in Eq. (1) implements the joint precision matrix of

$$\begin{bmatrix} u \\ v \end{bmatrix}$$

using the following hyperparameters

$$\tau_v$$
 and $h^2 = \frac{1/\tau_v}{1/\tau_v + 1/\tau_u}$.

Hyperparameters

The two parameters in the generic2 model are represented as

$$\theta_1 = \log(\tau_v)$$

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

and priors are assigned to (θ_1, θ_2) .

YES THIS IS CORRECT! This is because the prior spesification would then be similar to another spesification, but allows one to compute the posterior marginal for h^2 more easily.

Specification

The generic2model is specified inside the f() function as

where <Cmat> can be given in two different ways:

- a dense matrix or a sparse-matrix defined be Matrix::sparseMatrix().
- the name of a file giving the structure matrix. The file should have the following format

$$i \quad j \quad \mathbf{C}_{ij}$$

where i and j are the row and column index and C_{ij} is the corresponding element of the precision matrix. Only the non-zero elements of the precision matrix need to be stored in the file.

```
Hyperparameter spesification and default values
doc A generic model (type 2)
hyper
     theta1
         hyperid 20001
         name log precision cmatrix
         short.name prec
         initial 4
         fixed FALSE
         prior loggamma
         param 1 5e-05
         to.theta function(x) log(x)
         from.theta function(x) exp(x)
     theta2
         hyperid 20002
         name log precision random
         short.name prec.random
         initial 4
         fixed FALSE
         prior loggamma
         param 1 0.001
         to.theta function(x) log(x)
         from.theta function(x) exp(x)
constr FALSE
nrow.ncol FALSE
augmented FALSE
aug.factor 2
aug.constr 2
n.div.by
n.required TRUE
set.default.values TRUE
pdf generic2
Example
require(mvtnorm)
n = 200
Cm = matrix(runif(n^2,min=-1,max=1),n,n)
Cm = Cm \% * \% t(Cm)
```

Sigma = solve(Cm)

```
sd = 0.001
z = rnorm(n, sd=sd)
eta = rmvnorm(n=1,sigma = Sigma)
s = 0.1
y = c(eta) + rnorm(n,sd=s) + z
idx = 1:n
##
## Alternative 1
##
file = "Cmatrix.dat"
cat("",file=file, append = FALSE)
for(i in 1:n)
{
   j = i
   cat(i,j,Cm[i,j], "\n", sep = " ", file=file, append=TRUE)
   if (i < n)
       for(j in (i+1):n)
           cat(i, j, Cm[i,j], "\n", sep = " ", file=file, append=TRUE)
formula = y ~ f(idx, model = "generic2", Cmatrix = file,
               initial=c(0,0), fixed=c(F,F))
## Alternative 2
## formula = y ~ f(idx, model = "generic2", Cmatrix = Cm,
##
                 initial=c(0,0), fixed=c(F,F))
## Alternative 3
## Cm.sparse = as(Cm, "dgTMatrix")
## formula = y ~ f(idx, model = "generic2", Cmatrix = Cm.sparse,
                  initial=c(0,0), fixed=c(F,F))
result = inla(formula, data=data.frame(y,idx),
             control.family = list(initial = log(1/sd^2), fixed=TRUE),
             verbose = TRUE)
## tau.u should be about 1/s^2 = 100. increase 'n' above to get
tau.u = result$summary.hyperpar["Precision-cmatrix for idx", "mean"]
h2 = result$summary.hyperpar["h2 for idx", "mean"]
tau.v = h2/(1-h2)*tau.u
print(paste("tau.v", tau.v, "should be (for large n)", 1/s^2))
```

Notes

The option constr=TRUE will impose a sum-to-zero constraint on v only.