Generic 0 model

Parametrization

The Type 0 generic model implements the following precision matrix

$$\mathbf{Q} = \tau \mathbf{C}$$

where $\mathbf{C}$ is the structure matrix.

Hyperparameters

The precision parameters of the generic0 model is represented as

$$\theta = \log(\tau)$$

and prior is assigned to θ

Specification

The generic0 models is specified inside the `f()` function as

```
f(<whatever>,model="generic0",Cmatrix = <Cmat>,
              prior=c(<prior.model.theta>),
              param=c(<param.prior.theta1>))
```

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

- a list of type `Cmatrix = list(i = c(), j = c(), values = c())`, where `i`, `j` and `values` are vectors of the non-zero elements of $\mathbf{C}$. Note that `i` and `j` start from 1, and only the upper or lower part of $\mathbf{C}$ has to be given.
- 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 $\mathbf{C}_{ij}$ is the corresponding element of the precision matrix. Only the non-zero elements of the precision matrix need to be stored in the file.

See the following example for an application

Example

In the example below we define a RW1 model first using the `generic0` model and this using the `rw1` model.

```
#simulate data
n=50
z=seq(1,n,length.out=n)
y=sin(z/n*2*pi)+rnorm(n,mean=0,sd=0.5)
data=data.frame(y=y,z=z)

#specify Cmatrix
i=c(1:n,1:(n-1))
```

```
j=c(1:n,2:n)
values=c(1,rep(2,n-2),1,rep(-1,n-1))

#pass the C matrix as a list
#note that for the generic0 model constraints and diagonal have to be defined by the user
formula= y~f(z,model="generic0",Cmatrix=list(i = i, j = j, values = values),
             rankdef=1,constr=TRUE,diagonal=1e-05)
result=inla(formula,data=data,family="gaussian")

#pass the C matrix as a file
filename= inla.sparse2file(list(i=i,j=j,values=values))
formula1= y~f(z,model="generic0",Cmatrix=filename,
              rankdef=1,constr=TRUE,diagonal=1e-05)
result1=inla(formula1,data=data,family="gaussian")
unlink(filename)

#this is the same model defined using the rw1 model
formula2=y~f(z,model="rw1")
result2=inla(formula2,data=data,family="gaussian")
```

Notes

None