Proper/Non-intrinsic Besag model for spatial effects

Parametrization

The proper version of the Besag model for random vector $\mathbf{x} = (x_1, \ldots, x_n)$ is defined as

$$x_i | x_{-i}, \tau, d \quad \sim \quad \mathcal{N}\left(\frac{1}{d + n_i} \sum_{i \sim j} x_j, \frac{1}{\tau(d + n_i)}\right) \tag{1}$$

where n_i is the number of neighbours of node i, $i \sim j$ indicates that the two nodes i and j are neighbours, $d > 0$ is an extra term added on the diagonal controlling the "properness" and $\tau > 0$ is a "precision-like" (or scaling) parameter.

This parameterisation corresponds to this precision matrix $Q = (Q_{ij})$, where for $j \neq i$

$$Q_{ii} = \tau(n_i + d) \qquad \text{and} \qquad Q_{ij} = -\tau.$$

Hyperparameters

The precision parameter τ is represented as

$$\theta_1 = \log \tau$$

and the prior is defined on θ_1. The diagonal parameter d is represented as

$$\theta_2 = \log d$$

and the prior is defined on θ_2.

Specification

The besag model is specified inside the `f()` function as

```
f(<whatever>, model="besagproper", graph=<graph>,
  hyper=<hyper>)
```

The neighbourhood structure of `x` is passed to the program through the `graph` argument. The structure of this file is described below.

Hyperparameter spesification and default values

doc `A proper version of the Besag model`

hyper

theta1

hyperid `12001`

name `log precision`

short.name `prec`

prior `loggamma`

param `1 5e-04`

initial `2`

fixed `FALSE`

to.theta `function(x) log(x)`

from.theta `function(x) exp(x)`

theta2

hyperid 12002

name log diagonal

short.name diag

prior loggamma

param 1 1

initial 1

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 TRUE

set.default.values TRUE

pdf besagproper

Example

```
## pick a graph
graph = system.file("demodata/germany.graph", package="INLA")
g = inla.read.graph(graph.file)

## we will use replicated samples in our testing
nrep = 5

## make life easy; use dense matrix algebra
d = 1.0
tau = 1.0
Q = matrix(0, g$n, g$n)
diag(Q) = tau * (d + g$nnbs)
for(i in 1:g$n) {
    if (g$nnbs[i] > 0) {
        Q[i, g$nbs[[i]]] = -tau
        Q[g$nbs[[i]], i] = -tau
    }
}
R = chol(Q) ## 'chol' returns the upper triangular

## simulate data with replications
y = c()
```

```
for(i in 1:nrep) {
    y = c(y, backsolve(R, rnorm(g$n)))
}

i = rep(1:g$n, nrep)
replicate = rep(1:nrep, each = g$n)
formula = y ~ f(i, model="besagproper",  graph = graph,
        replicate=replicate,
        hyper = list(diag = list(param = c(1, 1)))) -1

## use 'exact' observations, so we fix the noise precisin to a high
## value
r = inla(formula,
        data = data.frame(y, i, replicate),
        family = "gaussian",
        control.family = list(
                hyper = list(
                        prec = list(
                                initial = 10,
                                fixed=TRUE))))
```

Notes

If $d = 0$ and the parameter `rankdef=1` is set, then this model corresponds to the `besag` model. `constr=FALSE` is default for this model.