The RW2d-model

Parametrization

The 2-dimensional random walk model is defined on a regular grid. The full conditional distributions for the nodes in the interior of the grid are given by:

$$\operatorname{Prec}(x_i \mid \mathbf{x}_{-i}, \tau) = 20\tau. \tag{2}$$

Necessary corrections to to equations (1) and (2) near the boundary can be found using the stencils in [Terzopoulos, 1988]. For a detailed description of this model see [Rue and Held, 2005, Sec. 3.4.2].

Hyperparameters

The precision parameter τ is the only hyperparameter, $\theta = \tau$. It is represented internally as $\log \tau$ and the prior is assigned to $\log \tau$.

Specification

The rw2d model is specified insiede the f() function as:

```
f(<whatever>,model="rw2d",
  nrow=<n.of rows>,ncol=<n.of columns>, bvalue=<bvalue>,
  hyper= <hyper>, scale.model = FALSE)
```

The logical option scale.model determine if the model should be scaled to have an average variance (the diagonal of the generalized inverse) equal to 1. This makes prior spesification much easier. Default is FALSE so that the model is not scaled.

The argument bvalue defines how the boundary is handled. If bvalue=0 (default) then precision matrix is created like we are condition on 0's outside the region. This model is proper. If bvalue=1 then only points inside the region is considered, and non-standard stencils are used near the boundary. This model is improper.

Hyperparameter spesification and default values

```
{f doc} Thin-plate spline model
```

hyper

theta

```
hyperid 32001
name log precision
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)
```

constr TRUE

```
nrow.ncol TRUE
augmented FALSE
aug.factor 1
aug.constr
n.div.by
n.required FALSE
set.default.values TRUE
pdf rw2d
Example
nrow=50
ncol=25
n = nrow*ncol
s.mat=matrix(NA,nrow=nrow,ncol=ncol)
j=1:ncol
for(i in 1:nrow)
    s.mat[i,j] = 0.1*(i+2*j)
## a covariate
z.mat=matrix(runif(nrow*ncol),nrow,ncol)
## noise
noise.mat=matrix(rnorm(nrow*ncol, sd=0.3),nrow,ncol)
## make simulated data
y.mat = s.mat + 0.5*z.mat + noise.mat
## convert matrices to the internal representation in INLA
y = inla.matrix2vector(y.mat)
z = inla.matrix2vector(z.mat)
node = 1:n
formula= y ~ z + f(node, model="rw2d", nrow=nrow, ncol=ncol)
data=data.frame(y=y,z=z,node=node)
## fit the model
result=inla(formula, family="gaussian", data=data)
#plot the posterior mean for 'node' with the truth
dev.new()
INLA:::inla.display.matrix(s.mat)
dev.new()
INLA:::inla.display.matrix(INLA:::inla.vector2matrix(result$summary.random$node$mean,nrow,ncol
```

Notes

All indexes in the R-INLA library are one-dimensional so an appropriate mapping is required to get it into the ordering defined internally in inla; see ?inla.matrix2vector, ?inla.vector2matrix, ?inla.node2lattice and ?inla.lattice2node.

The $\frac{n-r}{2}\log(|R|^*)$ -part (with r=3) of the normalisation constant is not computed, hence you need to add this part to the log marginal likelihood estimate, if you need it. Here, $Q=\tau R$.

References

[Rue and Held, 2005] Rue, H. and Held, L. (2005). Gaussian Markov Random Fields: Theory and Applications, volume 104 of Monographs on Statistics and Applied Probability. Chapman & Hall, London.

[Terzopoulos, 1988] Terzopoulos, D. (1988). The computation of visible-surface representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):417–438.