The Matérn-model

Parametrisation

This model is the Gaussian field with Matérn correlation function on a regular `nrow x ncol` -lattice. On an infinite lattice, the correlation function is

$$\mathrm{Corr}(d) = \frac{1}{2^{\nu-1}\Gamma(\nu)} (\kappa\, d)^{\nu}\ K_{\nu}(\kappa d), \qquad \alpha = \nu + \mathrm{d}/2,$$

where K_ν is the modified Bessel function and $\Gamma(\cdot)$ is the Gamma-function. The range is *defined* to be

$$r = \sqrt{8\nu}/\kappa$$

which about the distance where the covariance function becomes about 0.1.

The boundary conditions are so that the values are taken to be 0 outside the lattice. No further boundary options are available at this time.

Hyperparameters

The hyperparameters are the precision parameter τ and the range r,

$$\theta = (\tau, r)$$

The latent field has marginal variance $1/\tau$ and range (as defined above) r. Note that ν is fixed parameter and the model is available only for $\nu = 1, 2, 3$ ($\nu = 0$ is not yet ready). The hyperparameters are represented internally as

$$(\log\tau, \log r)$$

the prior are assigned to these quantities.

Specification

The `matern2d` model is specified inside the `f()` function as:

```
f(<whatever>, model="matern2d",nrow=<n.of rows>,ncol=<n.of columns>,
              nu = <value for nu, one of 1,2 or 3>, hyper = <hyper>)
```

Hyperparameter spesification and default values

hyper

theta1

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)`

theta2

name log range
short.name range

initial 2
fixed FALSE
prior loggamma
param 1 0.01
to.theta `function(x) log(x)`
from.theta `function(x) exp(x)`

constr FALSE

nrow.ncol TRUE

augmented FALSE

aug.factor 1

aug.constr

n.div.by

n.required FALSE

set.default.values TRUE

pdf matern2d

Example

```
nrow=20
ncol=30
n = nrow*ncol
s.noise = 1

zi.mat = matrix(NA,nrow=nrow,ncol=ncol)
i=1:nrow
for(j in 1:ncol)
    zi.mat[i,j] = 3*exp(-(i-j)^2/4)

## iid noise
noise.mat=matrix(rnorm(nrow*ncol, sd=s.noise),nrow,ncol)

## make simulated data with no spatial component
y.mat = zi.mat + noise.mat

## convert matrices to the internal representation in INLA
y = inla.matrix2vector(y.mat)
node = 1:n
formula= y ~ 1+ f(node, model="matern2d", nu=1, nrow=nrow, ncol=ncol,
        hyper = list(range = list(param =c(1, 1),
                                  prior = "loggamma",
                                  initial=1),
                     prec = list(param=c(1, 1))))
data=data.frame(y=y,node=node)

## fit the model
```

```
result=inla(formula, family="gaussian", data=data, verbose=TRUE,
        control.predictor = list(compute = TRUE),
        control.family = list(hyper = list(theta = list(initial = log(1/s.noise^2),
                                                fixed = FALSE))),
        keep=T)

## plot the posterior mean for 'predictor' and compare with the truth
dev.new()
INLA:::inla.display.matrix(zi.mat)
dev.new()
INLA:::inla.display.matrix(INLA:::inla.vector2matrix(result$summary.linear.predictor$mean,nrow
```

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`.

This model has much similarity with `rw2d`; please read the documentation for `rw2d`.