A short introduction on how to fit a **SPDE** model with **INLA**

Elias T. Krainski and Håvard Rue

November 20, 2013

This document ilustrates how to do a geostatistical fully Bayesian analysis through the **S**tochastic **P**artial **D**iferential **E**quation approach, [Lindgren et al., 2011], with **I**ntegrated **N**ested **L**aplace **A**proximation, [Rue et al., 2009], using the **INLA** package, `http://www.r-inla.org`.

1 Data simulation

Locations and the Random Field (RF) **covariance** matrix, exponential correlation function

```
n = 200; coo = matrix(runif(2*n), n)
k <- 10;    s2rf <- 0.7 ## RF params.
R <- s2rf*exp(-k*as.matrix(dist(coo)))
```

RF sample, a multivariate Normal realization

```
s <- drop(rnorm(n)%*%chol(R))
```

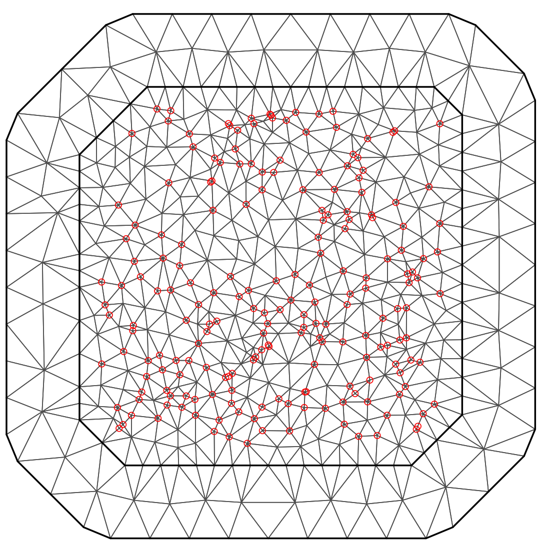
A **covariate** effect and a noise can be added

```
x <- runif(n);   beta <- 1:2;   s2e <- 0.3
lin.pred <- beta[1] + beta[2]*x + s
y <- lin.pred + rnorm(n, 0, sqrt(s2e))
```

2 Model fitting: steps

1. **Mesh:** a triangulation to discretize the random field (RF) at m nodes. The additional triangles outer domain is to avoid boundary effects.

```
mesh <- inla.mesh.2d(
    coo, ## provide locations or domain
    max.edge=c(1/k, 2/k), ## mandatory
    cutoff=0.1/k) ## good to have >0
plot(mesh, asp=1);  points(coo, col='red')
```



A little warning about the triangle on the mesh. Is good to have they aproximately isosceles. Also, we have to look at the range of the process and compare it with the edges lengths of the inner mesh triangles. We need the lengths of the edges to be less than the range of the process. If it is too small, there might not be any spatial effect.

2. **Define the $n \times m$ projector matrix** to project the process at the mesh nodes to locations

```
dim(A <- inla.spde.make.A(
    mesh=mesh, loc=coo)) ## 'n' by 'm'

## [1] 200 505
```

3. **Build the SPDE model** on the mesh. Exponential correlation function, $\alpha = 3/2$

```
spde <- inla.spde2.matern(
    mesh=mesh, alpha=1.5)
```

4. **Create a stack data** for the estimation. This is a way to allow models with complex linear predictors. In our case, we have a SPDE model defined on m nodes. It must be combined with the covariate (and the intercept) effect at n locations. We do it using different projector matrices.

```
stk.e <- inla.stack(tag='est', ## tag
    data=list(y=y),  ## response
    A=list(A, 1), ## two projector matrix
    effects=list(## two elements:
        s=1:spde$n.spde, ## RF index
        data.frame(b0=1, x=x)))
```

5. **Fit** the posterior marginal distributions for all model parameters

```
formula <- y ~ 0 + b0 + x + ## fixed part
    f(s, model=spde) ## RF term
res <- inla(formula,
          data=inla.stack.data(stk.e),
    control.predictor=list(A =
        inla.stack.A(stk.e))# projector
```

3 Posterior marginal distributions - PMDs

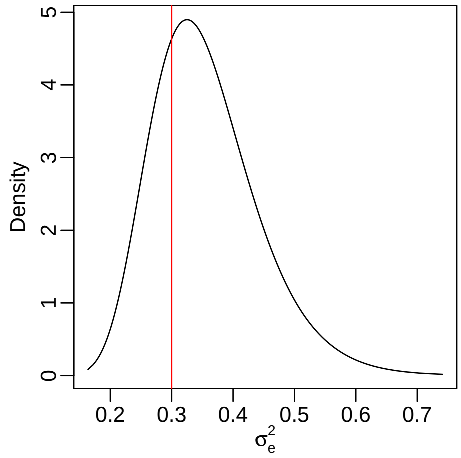
Summary of the regression coefficients PMDs

```
round(res$summary.fixed, 4)

##      mean     sd 0.025quant 0.5quant 0.975quant  mode kld
## b0 1.041 0.2928     0.4223    1.050      1.607 1.064   0
## x  1.903 0.1895     1.5286    1.903      2.273 1.905   0
```

INLA works with precisions. We have to transform the precision PMD to have the variance PMD. It can be done and visialized by

```
post.s2e <-
  inla.tmarginal(# tranformation function
  function(x) 1/x, ## inverse transf.
  res$marginals.hyperpar$
'Precision for the Gaussian observations')
plot(post.s2e, type='l', ylab='Density',
     xlab=expression(sigma[e]^2))
abline(v=s2e, col=2) ## add true value
```

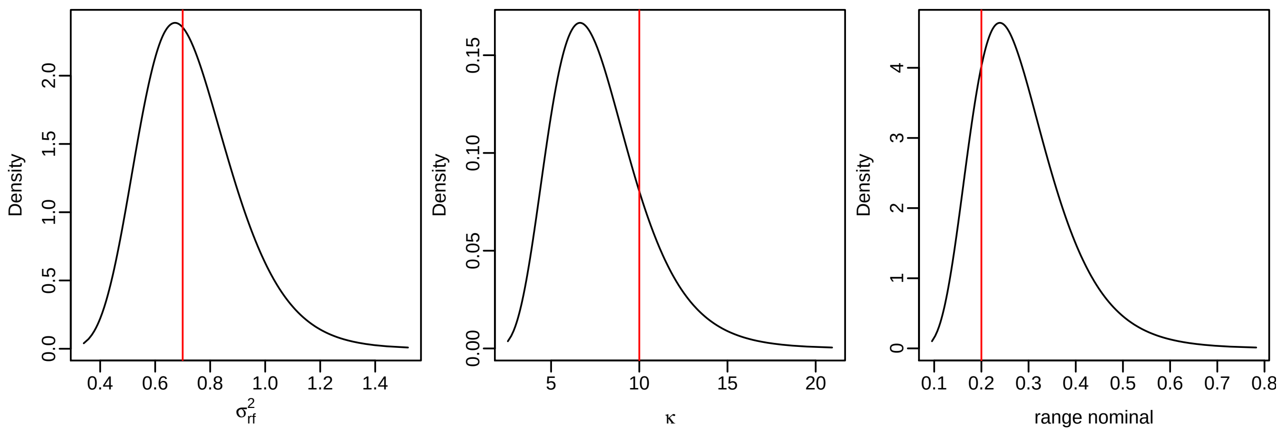


The SPDE approach uses a local variance, τ^2, such that $\sigma^2_{rf} = 1/(2\pi\kappa^2\tau^2)$. On **INLA** we work $\log(\tau^2)$ and $\log(\kappa)$. So, especially for σ^2_{rf}, we have to do an additional computation. The PMDs for all RF parameters on user scale are computed by

```
rf <- inla.spde.result(
    inla=res, ## the inla() output
    name='s', ## name of RF index set
    spde=spde, ## SPDE model object
    do.transf=TRUE) ## to user scale
```

It can be visualized by

```
plot(rf$marginals.var[[1]], ty = "l", xlab = expression(sigma[rf]^2), yla = "Density")
abline(v = s2rf, col = 2)  ## add the true value
plot(rf$marginals.kap[[1]], type = "l", xlab = expression(kappa), ylab = "Density")
abline(v = k, col = 2)  ## add the true value
plot(rf$marginals.range[[1]], type = "l", xlab = "range nominal", ylab = "Density")
abline(v = sqrt(4)/k, col = 2)  ## add the true value
```



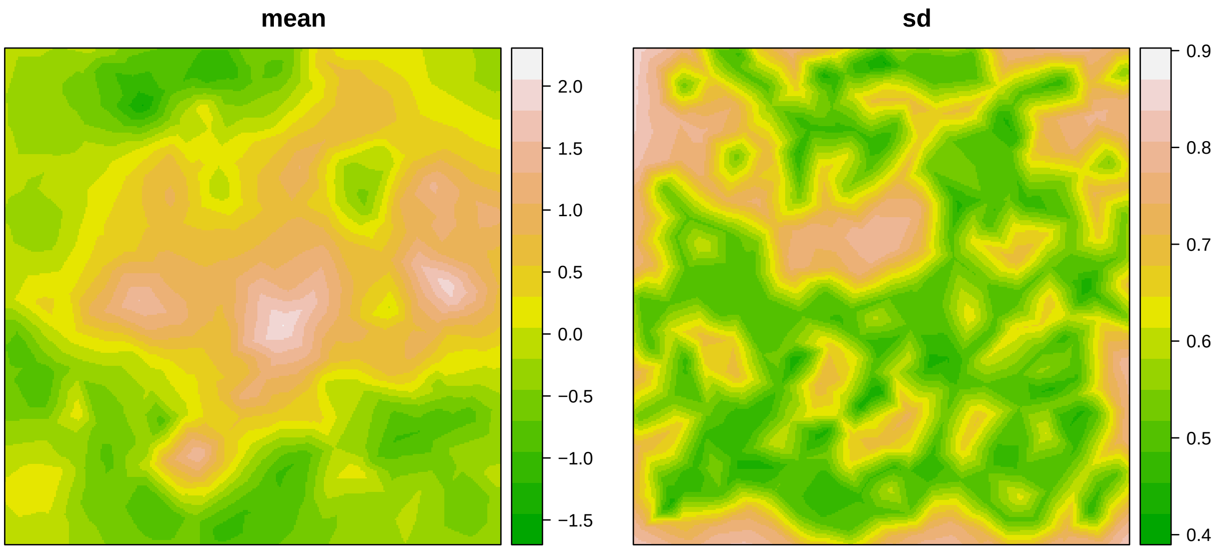
4 Projection on a grid

An interesting result is the map of the RF on a grid. The simplest way to have it is by projection. We just have to define the projector matrix and project, for example, the posterior mean and posterior standard deviation on the grid.

```
gproj <- inla.mesh.projector(mesh, xlim = 0:1, ylim = 0:1, dims = c(300, 300))
g.mean <- inla.mesh.project(gproj, res$summary.random$s$mean)
g.sd <- inla.mesh.project(gproj, res$summary.random$s$sd)
```

We can visualize it by

```
library(lattice);     library(gridExtra)
trellis.par.set(regions=list(col=terrain.colors(16)))
grid.arrange(levelplot(g.mean, scales=list(draw=F), xlab='', ylab='', main='mean'),
             levelplot(g.sd, scal=list(draw=F), xla='', yla='', main='sd'), nrow=1)
```



5 Prediction

Define target locations, the corresponding projector matrix and covariate values at target locations

```
tcoo <- rbind(c(0.3, 0.3), c(0.5, 0.5), c(0.7, 0.7))
dim(Ap <- inla.spde.make.A(mesh = mesh, loc = tcoo))

## [1]   3 505

x0 <- c(0.5, 0.5, 0.5)
```

To do a fully Bayesian analysis, we include the target locations on the estimation process by assigning NA for the response at these locations. Defining the prediction stack

```
stk.pred <- inla.stack(tag='pred', A=list(Ap, 1), data=list(y=NA), ## response as NA
                       effects=list(s=1:spde$n.spde, data.frame(x=x0, b0=1)))
```

Fit the model again with the full stack

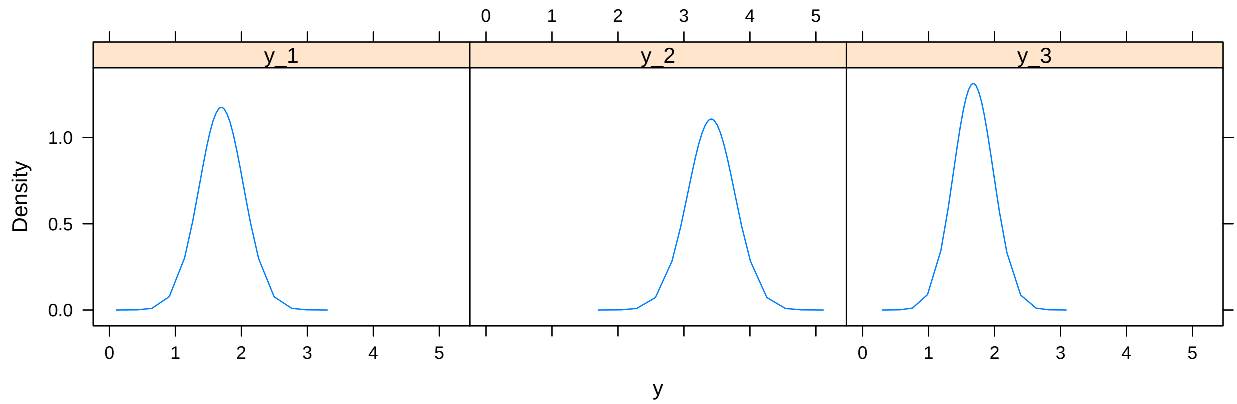
```r
stk.full <- inla.stack(stk.e, stk.pred)
p.res <- inla(formula, data=inla.stack.data(stk.full), ## full stack
              control.predictor=list(compute=TRUE, ## compute the predictor
                 A=inla.stack.A(stk.full))) ## using full stack data
```

Get the prediction data index and collect the PMD to work with

```r
pred.ind <- inla.stack.index(stk.full, tag = "pred")$data
ypost <- p.res$marginals.fitted.values[pred.ind]
```

Visualize PMD for the linear predictor at target locations with commands bellow

```r
names(ypost) <- paste('y', seq_along(ypost), sep='_');     library(plyr)
xyplot(y~x | .id, ldply(ypost), panel='llines', xlab='y', ylab='Density')
```



In **INLA** we have some functions to work with marginals distributions

```r
apropos("marginal")

## [1] "inla.dmarginal"   "inla.emarginal"   "inla.hpdmarginal"
## [4] "inla.mmarginal"   "inla.pmarginal"   "inla.qmarginal"
## [7] "inla.rmarginal"   "inla.smarginal"   "inla.tmarginal"
```

```r
inla.mmarginal(ypost[[1]]) ## mode

## [1] 1.697

inla.qmarginal(c(0.15, 0.7), ## quantiles
               ypost[[1]])

## [1] 1.345 1.875

inla.pmarginal(inla.qmarginal(
    0.3, ypost[[1]]), ypost[[1]])

## [1] 0.3
```

References

[Lindgren et al., 2011] Lindgren, F., Rue, H., and Lindström, J. (2011). An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach (with discussion). *J. R. Statist. Soc. B*, 73(4):423–498.

[Rue et al., 2009] Rue, H., Martino, S., and Chopin, N. (2009). Approximate bayesian inference for latent gaussian models using integrated nested laplace approximations (with discussion). *Journal of the Royal Statistical Society, Series B*, 71(2):319–392.