Advanced features in INLA and inlabru

University of Zurich, March, 2022

Instructor: Sara Martino

Department of Mathematical Science (NTNU)

Norwegian University of Science and Technology

What have we learned

INLA in a nurshell

- many data sets these days are complex, resulting in complex models, e.g. with complex dependence structures (spatial, temporal, etc..)
- usually Markov chain Monte Carlo (MCMC) methods have been used to fit these models
  - (realistically) complex models result in very long running times
  - often impossible (or unrealistic) to fit
- INLA (Integrated nested Laplace approximation) is an alternative to MCMC
  - much, much **faster**
  - suitable for a specific (but very large!) class of models

INLA in a nurshell

Three main ingredients in INLA

- Gaussian Markov random fields
- Latent Gaussian models
- Laplace approximations

which together (with a few other things) give a very nice tool for Bayesian inference

- quick
- accurate

Models amenable to INLA: **Latent Gaussian Models**

$$\mathbf{y}|\mathbf{x},\theta \sim \prod_i \pi(y_i|x_i,\theta) \qquad \text{Likelihood}$$

$$\mathbf{x}|\theta \sim \exp\left(-\frac{1}{2}\mathbf{x}^{\mathbf{T}}\mathbf{Q}(\theta)\mathbf{x}^{\mathbf{T}}\right) \qquad \text{Latent field (GMRF)}$$

$$\theta \sim \pi(\theta) \qquad \text{Hyperparameter}$$

- Each data point depends on only one of the elements in the latent Gaussian field $\mathbf{x}$.

Models amenable to INLA: **Latent Gaussian Models**

$$\mathbf{y}|\mathbf{x},\theta \sim \prod_i \pi(y_i|x_i,\theta) \qquad \text{Likelihood}$$

$$\mathbf{x}|\theta \sim \exp\left(-\frac{1}{2}\mathbf{x}^{\mathbf{T}}\mathbf{Q}(\theta)\mathbf{x}^{\mathbf{T}}\right) \qquad \text{Latent field (GMRF)}$$

$$\theta \sim \pi(\theta) \qquad \text{Hyperparameter}$$

- Each data point depends on only one of the elements in the latent Gaussian field $\mathbf{x}$.
- The size of the hyperparameter vector θ is small (say < 15)

Models amenable to INLA: **Latent Gaussian Models**

$$\mathbf{y}|\mathbf{x}, \theta \sim \prod_i \pi(y_i|x_i, \theta) \qquad \text{Likelihood}$$

$$\mathbf{x}|\theta \sim \exp\left(-\frac{1}{2}\mathbf{x}^{\mathbf{T}}\mathbf{Q}(\theta)\mathbf{x}^{\mathbf{T}}\right) \qquad \text{Latent field (GMRF)}$$

$$\theta \sim \pi(\theta) \qquad \text{Hyperparameter}$$

- Each data point depends on only one of the elements in the latent Gaussian field $\mathbf{x}$.
- The size of the hyperparameter vector θ is small (say < 15)
- The latent field $\mathbf{x}$, can be large but it is endowed with some conditional independence (Markov) properties so that the precision matrix $Q(\theta)$ is sparse.

Models amenable to INLA: **Latent Gaussian Models**

$$\mathbf{y}|\mathbf{x},\theta \sim \prod_i \pi(y_i|x_i,\theta) \qquad \text{Likelihood}$$

$$\mathbf{x}|\theta \sim \exp\left(-\frac{1}{2}\mathbf{x}^{\mathbf{T}}\mathbf{Q}(\theta)\mathbf{x}^{\mathbf{T}}\right) \qquad \text{Latent field (GMRF)}$$

$$\theta \sim \pi(\theta) \qquad \text{Hyperparameter}$$

- Each data point depends on only one of the elements in the latent Gaussian field $\mathbf{x}$.
- The size of the hyperparameter vector θ is small (say < 15)
- The latent field $\mathbf{x}$, can be large but it is endowed with some conditional independence (Markov) properties so that the precision matrix $Q(\theta)$ is sparse.
- The linear predictor depends **linearly** on the unknown smooth function of covariates.

Models amenable to INLA: **Latent Gaussian Models**

$$\mathbf{y}|\mathbf{x},\theta \sim \prod_i \pi(y_i|x_i,\theta) \qquad \text{Likelihood}$$

$$\mathbf{x}|\theta \sim \exp\left(-\frac{1}{2}\mathbf{x}^{\mathbf{T}}\mathbf{Q}(\theta)\mathbf{x}^{\mathbf{T}}\right) \qquad \text{Latent field (GMRF)}$$

$$\theta \sim \pi(\theta) \qquad \text{Hyperparameter}$$

- Each data point depends on only one of the elements in the latent Gaussian field $\mathbf{x}$.
- The size of the hyperparameter vector θ is small (say < 15)
- The latent field $\mathbf{x}$, can be large but it is endowed with some conditional independence (Markov) properties so that the precision matrix $Q(\theta)$ is sparse.
- The linear predictor depends **linearly** on the unknown smooth function of covariates.

All of this is implemented in the `INLA` library in `R`.

Yesterday we saw how to implement simple models with `INLA`

There are many more features that can be implemented with `INLA`

All of this is implemented in the `INLA` library in `R`.
Yesterday we saw how to implement simple models with `INLA`
There are many more features that can be implemented with `INLA`

- Spatial models
- Use several likelihoods
- Predict linear combinations of elements of the latent field
- Group/copy/replicate random fields

All of this is implemented in the `INLA` library in `R`.
Yesterday we saw how to implement simple models with `INLA`
There are many more features that can be implemented with `INLA`

- Spatial models
- Use several likelihoods
- Predict linear combinations of elements of the latent field
- Group/copy/replicate random fields

Many of these features are **a lot** easier to implement in `inlabru`

`inlabru` in a nutshell

`inlabru` is a friendlier version of `R-INLA`

- it makes `INLA` more accessible to the user
- makes complex features and predictions (especially for spatial data) a lot easier
- is a softer-wrapper around `INLA`
- allows to release the

inlabru

- Installation:
  - There is a CRAN version

```
install.packages("inlabru")
```

- You can also install the development version of inlabru from GitHub (recommended)

```
# install.packages("remotes")
remotes::install_github("inlabru-org/inlabru",
          ref="devel")
```

- Documentation
  - Web site: https://sites.google.com/inlabru.org/inlabru
  - Github: https://github.com/inlabru-org/inlabru

Spatial models

Types of Spatial Data

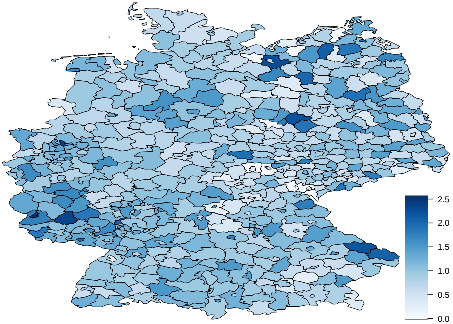
We can distinguish three types of spatial data

- Discrete space
  - data on a spatial grid
- Continuous space:
  - geostatistical data
  - spatial point data

Discrete Counts:

Data on a spatial grid

- examples: number of individuals in a region, average rainfall in a province
- (originally geostatistical or point data; gridded for practical reasons)

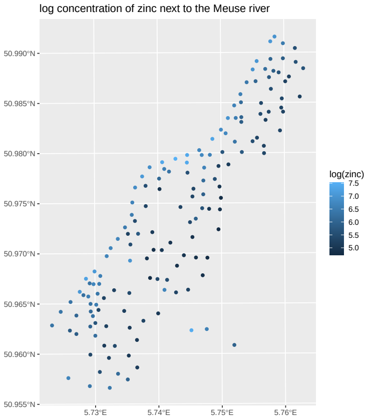


Observed response(s):

- Measurement over each grid cell (e.g. number of individuals in cell; rainfall in province)

Continuous Space: Geostatistics

- phenomenon that is continuous in space
- examples: nutrient levels in soil, salinity in the sea
- measurements at a given set of locations that are determined by surveyor

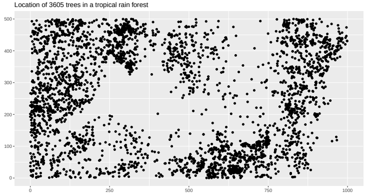


Observed response(s):

- measurement(s) taken at given locations

Continuous Space: Point Process

- locations of objects (individuals) in space (typically 2D)
- examples: locations of trees in a forest, groups of animals



Observed response(s):

- x, y coordinates of points (individuals/groups)
- maybe also properties of individuals/groups ("marks")

Gaussian Random Field models

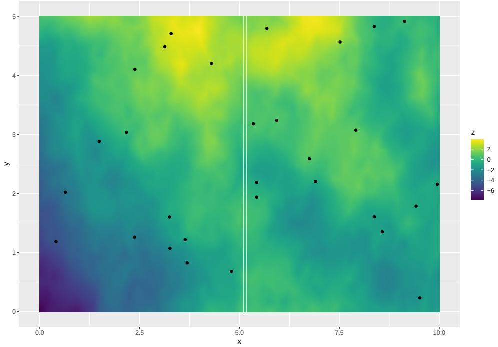
Gaussian Random fields

Definition:A random function $u(x) : R^d \to R$ is a Gaussian random field if for any finite collection of locations, $(x_1, \ldots, x_n)$, $x_i \in R^d$, the joint distribution of $\mathbf{u} = (u(x_1), \ldots, u(x_n))$ is $\mathbf{u} \sim N(0, \Sigma)$, and

$$E(u(x)) = 0,$$
$$Cov(u(x), u(x')) = R(x, x'), \quad \Sigma_{ij} = R(x_i, x_j)$$

for some expectation function $\mu(\cdot)$ and positive definite covariance function $R(\cdot, \cdot)$. Σ is the covariance matrix for the specific location collection.

Example



Gaussian Random field

GRF are a very popular model

- Flexible and easy to use
- Can be part of the latent Gaussian field in a LGM

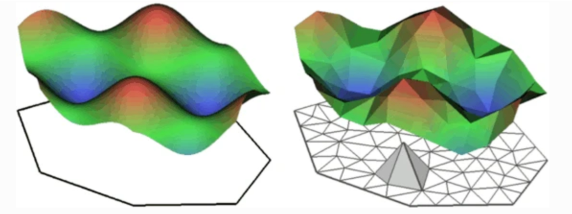
However:

- **computationally inefficient** (the precision matrix is dense)
- not flexible enough (complicated boundary, barrier,...)

The SPDE approach

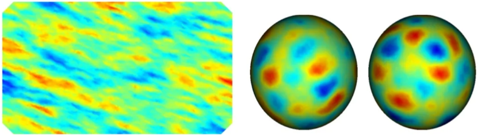
The SPDE approach

- Matern fields can be seen as solution to a PDE
- Using finite element methods such solution can be represented using a GRMF



Advantages of the SPDE approach

- Computationally fast
- Allows for flexible modeling
  - non-stationary models (anisotropy)
  - models on a sphere
  - non separable models

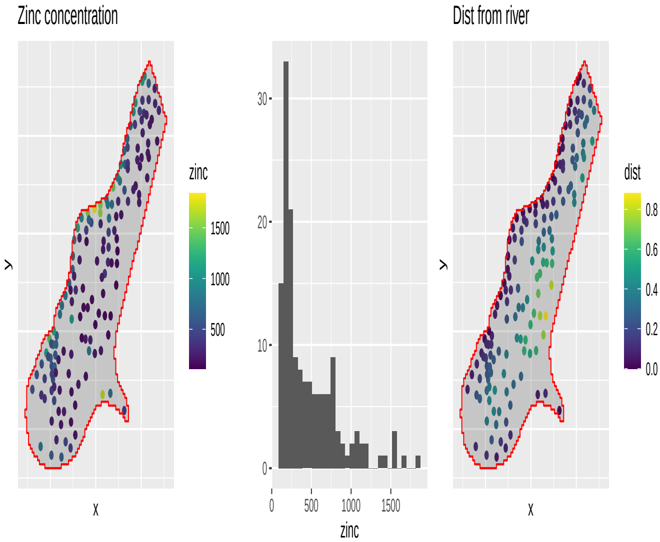


All these models (and my more) can be fitted with `R-INLA` and `inlabru`

Fitting spatial models

Example: Meuse Data

Measures of zinc concentration.



The model

$$\log(Y(s)) \sim \mathcal{N}(\eta(s), \sigma_y^2)$$
$$\eta(s) = \alpha + u(s)$$

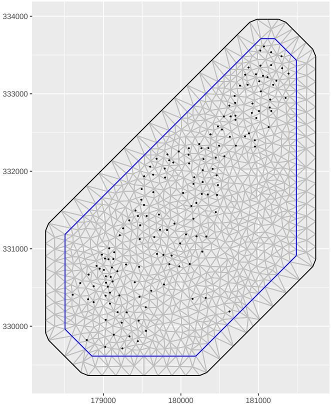
where

- $Y(s)$ is the measure of zinc in location s
- α a common intercept
- $u(s)$ the Matern Gaussian field

Define the SPDE representation: The mesh

1. Define the mesh

```
mesh <- inla.mesh.2d(loc.domain = cbind(meuse$x, meuse$y),
                     max.edge = c(150, 500),
                     offset = c(100, 250) )
```



Define the SPDE representation: The mesh

- All random field models need to be discretised for practical calculations.
- The SPDE models were developed to provide a consistent model definition across a range of discretisations.
- We use finite element methods with local, piecewise linear basis functions defined on a triangulation of a region of space containing the domain of interest.
- Deviation from stationarity is generated near the boundary of the region.
- The choice of region and choice of triangulation affects the numerical accuracy.

Define the SPDE representation: The mesh

- All random field models need to be discretised for practical calculations.
- The SPDE models were developed to provide a consistent model definition across a range of discretisations.
- We use finite element methods with local, piecewise linear basis functions defined on a triangulation of a region of space containing the domain of interest.
- Deviation from stationarity is generated near the boundary of the region.
- The choice of region and choice of triangulation affects the numerical accuracy.

Two separate issues:

- Continuous space with bounded domain: Boundary effect
- Discretised model: Numerical accuracy

Sometimes the boundary effect may be desireable.

Define the SPDE representation: The mesh

- Too fine meshes → heavy computation
- Too coarse mesh → not accurate enought

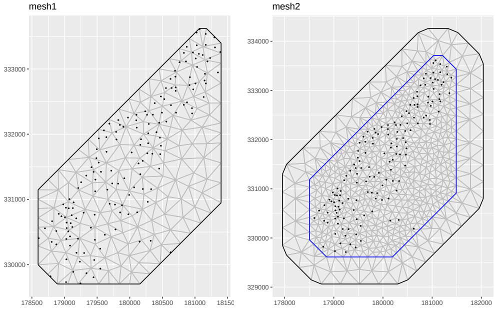
Some guidelines

- Create triangulation meshes with `inla.mesh.2d()`
- Move undesired boundary effects away from the domain of interest by extending to a smooth external boundary (`inla.nonconvex.hull(loc, convex)`, `convex` $\geq$ correlation range)
- Use a coarser resolution in the extension to reduce computational cost (`max.edge=c(inner, outer)`)
- Use a fine resolution (subject to available computational resources) for the domain of interest (inner correlation range) and filter out small input point clusters (`0 < cutoff < inner`)
- Coastlines and similar can be added to the domain specification in `inla.mesh.2d()`

Define the SPDE representation: The mesh

```
mesh1 = inla.mesh.2d(loc.domain = cbind(meuse$x, meuse$y),
                     max.edge = 350,
                     offset = 10)

mesh2 = inla.mesh.2d(loc.domain = cbind(meuse$x, meuse$y),
                     max.edge = c(150, 500),
                     cutoff = 100,
                     offset = c(100, 550) )
```



Define the SPDE representation: The SPDE model

```
meuse.spde <- inla.spde2.pcmatern(mesh = mesh,
                                  prior.sigma = c(1, 0.1),
                                  prior.range = c(1000, 0.5))
```

PC-priors for the range ρ and the standard deviarion σ

- Define the prior for the range `prior.range = (range0,Prange)` $\text{Prob}(\rho < \rho_0) = p_\rho$
- Define the prior for the range `prior.sigma = (sigma0,Psigma)` $\text{Prob}(\sigma < \sigma_0) = p_\sigma$

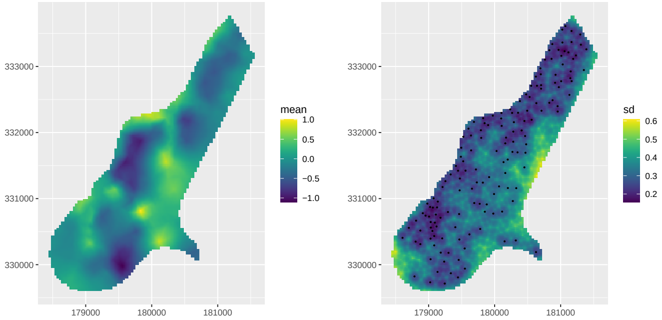
Run the model inlabru

```
# create a spatial object
coordinates(meuse) = c("x","y")
#  covariate values
dist_SPDE = SpatialPixelsDataFrame(data$dist_raster[,c(1,2)],
                                   data = data.frame(dist = data$dist_raster[,3]))
# model components
cmp =  ~ Intercept(1) + dist(dist_SPDE, model = "linear") +
  spde(coordinates, model = meuse.spde)
# define likelihood
lik = like(formula = Y ~ Intercept + dist +   spde,
           family = "gaussian",
           data  = meuse)
#fit the model
fit <- bru(cmp, lik)
# define prediction area
pix <- pixels(mesh, nx = 200, ny = 200, mask = boundary)
# generate predictions
pred = predict(fit, pix, ~ data.frame(
                                       spde = spde,
                                       logscale = Intercept + dist + spde,
                                       naturalscale = exp(Intercept + dist + spde)))
```

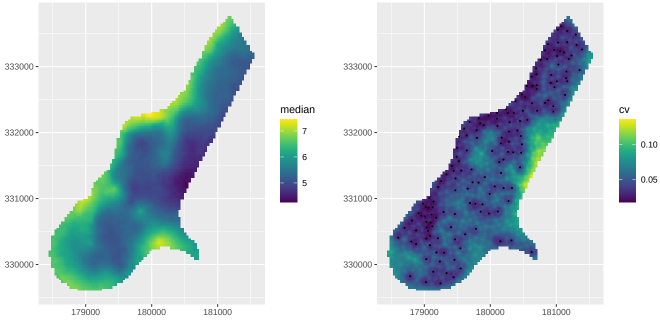
Notes!

- The data are a spatial object!
- The covariates are stored in a `SpatialPixelsDataFrame` and need to cover all the mesh nodes

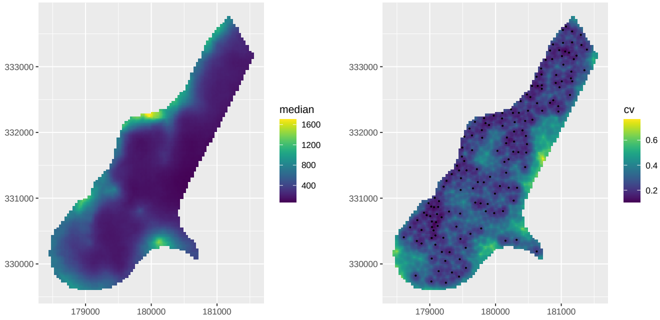
Predictions: The SPDE field



Predictions: The log concentrations



Predictions: The concentrations



Same in plain INLA (1)

```
A.meuse <- inla.spde.make.A(mesh = mesh, loc = coordinates(meuse))
s.index <- inla.spde.make.index(name = "spatial.field",
  n.spde = meuse.spde$n.spde)

#Create data structure
meuse.stack <- inla.stack(data  = list(zinc = meuse$zinc),
  A = list(A.meuse, 1),
  effects = list(c(s.index, list(Intercept = 1)),
    list(dist = meuse$dist)),
  tag = "meuse.data")

data(meuse.grid)
coordinates(meuse.grid) = ~x+y
gridded(meuse.grid) = TRUE

#Create data structure for prediction
A.pred <- inla.spde.make.A(mesh = mesh, loc = coordinates(meuse.grid))
meuse.stack.pred <- inla.stack(data = list(zinc = NA),
  A = list(A.pred, 1),
  effects = list(c(s.index, list (Intercept = 1)),
    list(dist = meuse.grid$dist)),
  tag = "meuse.pred")

#Join stack
join.stack <- inla.stack(meuse.stack, meuse.stack.pred)
```

Same in plain INLA (2)

```
#Fit model
form <- log(zinc) ~ -1 + Intercept + dist + f(spatial.field, model = spde)

m1 <- inla(form, data = inla.stack.data(join.stack, spde = meuse.spde),
  family = "gaussian",
  control.predictor = list(A = inla.stack.A(join.stack), compute = TRUE))
```

Note: We still have not compute predictions...and this is not too easy in plain `INLA`!!

When is `inlabru` easier to use

- spatial modeling.
- point processes.
- multiple likelihoods
- when interested in spatial predictions

When is `inlabru` easier to use

- spatial modeling.
- point processes.
- multiple likelihoods
- when interested in spatial predictions
- `inlabru` is also useful if one has non-linearities in the predictor η
    - born for ecological models (for example transect sampling) but used also in other fields

Space-time Modeling