INLA for Spatial Statistics

9. Grouped models

Daniel Simpson

Department of Mathematical Sciences
University of Bath

Outline

Models with groups

Yesterday we talked about replicated random effects, where we observed i.i.d. draws from the random effect distribution.

- Point patterns observed at different plots
- Annual rainfall observed during different years

But is this enough?

No it isn't!

In a lot of applications, the assumptions that the repeated random effects are *independent* is very restrictive.

- Monthly / daily rainfall data
- The results of nearby plots could be correlated

INLA provides the concept of a "group" that allows more complicated dependence structures

Group dependence

Grouped random effects work as follows

- There is a *within group* correlation structure
  - Any INLA latent model (`iid`, `ar1`, `bym`, `spde` etc)
- There is also a *between group* correlation model
  - Not every model: "exchangeable" "ar1" "ar" "rw1" "rw2" "besag"

If $x_{g,i}$ is the ith element in group g, then

$$\begin{aligned}\mathrm{Cov}(x_{g_1,i_1}, x_{g_2,i_2}) = &(\text{cov between groups } g_1 \text{ and } g_2)\\ &\times (\text{cov between elements } i_1 \text{ and } i_2)\end{aligned}$$

The Kronecker structure

Grouped models are a special case of "Kronecker models"

- These models have covariance matrices of the form $\boldsymbol{\Sigma}_{\text{between group}} \otimes \boldsymbol{\Sigma}_{\text{within group}}$
- We are working to implement the general structure (so you can group any models in INLA together)
- We're going to look through some examples...

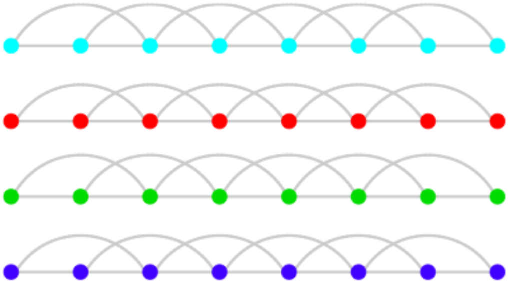
Outline

Correlated random effects

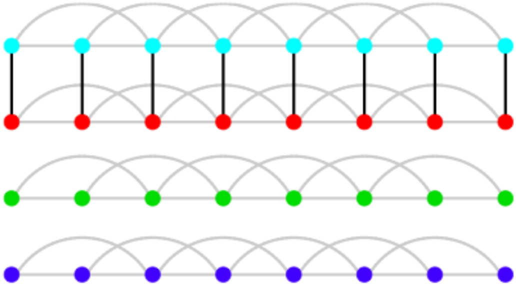
The simplest group model in INLA is the `exchangeable` model

- "Uniform correlation matrix"
- $\text{Corr}(\text{group } i, \text{group } j) = \rho$, $-1 < \rho < 1$
- This basically says that all of the groups are correlated in the same way
- This is all you need for *two* correlated effects
- Allows for some dependence in other cases.

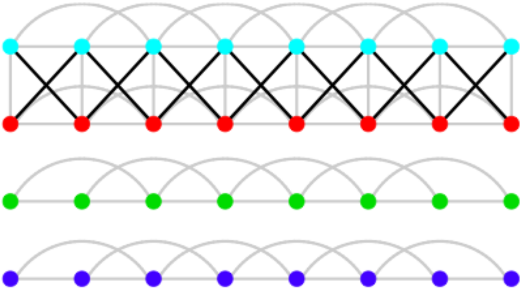
Graph for correlated RW2



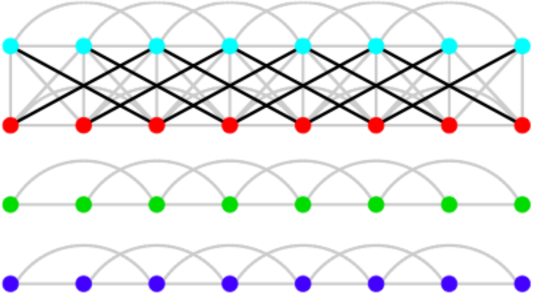
Graph for correlated RW2



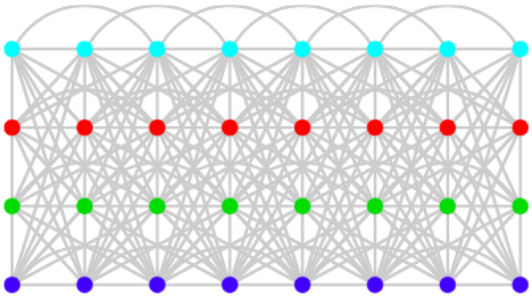
Graph for correlated RW2



Graph for correlated RW2



Graph for correlated RW2

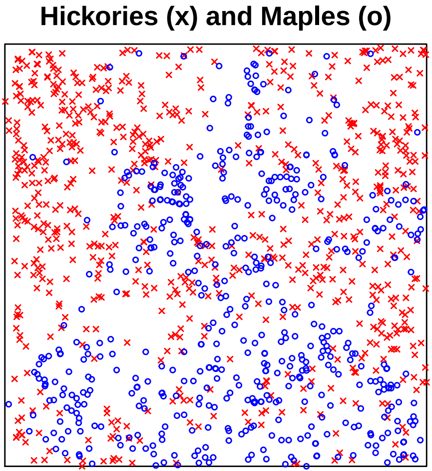


Multispecies point patterns

Think about trees

- Many species appear together
- We don't really think that these patterns are independent
- We can fit bivariate patterns and take a look at the correlation

Maple and Hickory



The Linear Model of Co-regionalisation (LMC)

The easiest way of modelling this is the LMC, which says

- Fit a common random effect for the two species
- For one species, add an independent random effect to "mop up" the extra structure

$$\eta_{\text{maple}} = (\text{common effect})$$
$$\eta_{\text{hickory}} = \beta(\text{common effect}) + (\text{extra hickory effect})$$

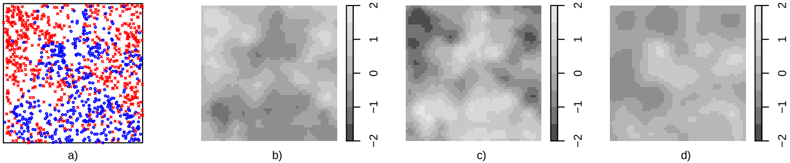
LMC in INLA

```
#Make indices
common_maple = c(1:n,rep(NA,n))
common_hickory = c(rep(NA,n), 1:n)
extra_hickory = c(rep(NA,n),1:n)

# Make formula

formula = y ~ ... + f(common_maple,model="rw2d")
      + f(common_hickory,copy="common_maple",
              hyper = list(beta=list(fixed=FALSE)))
      + f(extra_hickory, model="rw2")
```

Results



(b): posterior mean for hickories, (c) post. mean for maples, (d) excess effect

The grouped version

The other option is to model the random effect for each species separately and let them be correlated.

- Advantage: A single parameter (ρ) that tells you about correlation
- Disadvantage: You don't get the pretty picture

```
#indices
effect = c(1:n,1:n)
group = rep(c(1,2), each=n)

#formula
formula = y~ ... + f(effect,model="rw2d",group=group,
              control.group = list(model="exchangeble"))
```

Results with SPDE model

	range hickory	range maple	correlation	DIC
est	64	67	-0.69	-
group	70 (48, 98)	-	-0.63 (-0.77, -0.46)	5568.5
LMC	70 (42, 109)	110 (72, 178)	-0.79 (-0.95, -0.53)	5566.3

- Fitted using SPDE models (not `rw2d`)
- This allows for estimation of the correlation range for each parameter
- We see strong negative correlation
- In this case, the LMC fits better
- The better fit is attributed to the components having different correlation ranges for different species

Outline

Spatiotemporal models

- Data frequently has a temporal component
- Easy fixes:
  - Treat them as independent (`replicate`)
  - Add a temporal random effect

$$\eta = \ldots + f(\text{space}) + f(\text{time})$$

Spatiotemporal models

- Data frequently has a temporal component
- Easy fixes:
  - Treat them as independent (`replicate`)
  - Add a temporal random effect

$$\eta = \ldots + f(\text{space}) + f(\text{time})$$

- Harder fix: Try to make space time models

There are two types of space-time models:

- Separable models:
  - Correlation between two points in space-time $=$ Corr in space $\times$ Corr in time
  - This is easy to do and works well
  - Doesn't capture "spreading fronts"

We're going to fit a separable model

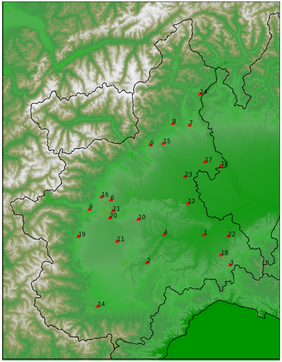
There are two types of space-time models:

- Separable models:
  - Correlation between two points in space-time = Corr in space $\times$ Corr in time
  - This is easy to do and works well
  - Doesn't capture "spreading fronts"
- Non-separable models:
  - Anything that isn't separable!
  - Much more flexible
  - But harder to fit...
  - Not in INLA (yet...)

We're going to fit a separable model

PM-10 concentration in Piemonte, Italy

Everything that I'm talking about today is described in Cameletti *et al.* (2011) on `r-inla.org`. (It's a really good paper!)



PM10 concentration:

- 24 monitoring stations
- Daily data from 10/05 to 03/06

Covariates

- Daily mean wind speed (WS, m/s)
- Daily maximum mixing height (HMIX, m)
- Daily precipitation (P, mm)
- Daily mean temperature (TEMP, K°)
- Daily emissions (EMI, g/s)
- Altitude (A, m) Coordinates (UTMX and UTMY, km).

The latent field (state equation)

We use an AR(1) structure

$$\boldsymbol{\xi}_t = a\boldsymbol{\xi}_{t-1} + \boldsymbol{\omega}_t,$$

where $a \in (0, 1)$ is a constant and

$$\boldsymbol{\omega}_t \overset{\text{i.i.d.}}{\sim} N(\mathbf{0}, \boldsymbol{Q}^{-1}),$$

is taken from a spatial SPDE model.

The measurement equation

We take the measurement equation to be

$$\boldsymbol{y}_t = \boldsymbol{X}_t \boldsymbol{\beta} + \boldsymbol{A}\boldsymbol{\xi}_t + \boldsymbol{\epsilon}_t,$$

where $\boldsymbol{X}_t$ is a matrix of covariates, $\boldsymbol{\beta}$ are the weights, $\boldsymbol{A}$ picks out the appropriate values of $\boldsymbol{\xi}_t$ and

$$\boldsymbol{\epsilon}_t \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2 \boldsymbol{I}).$$

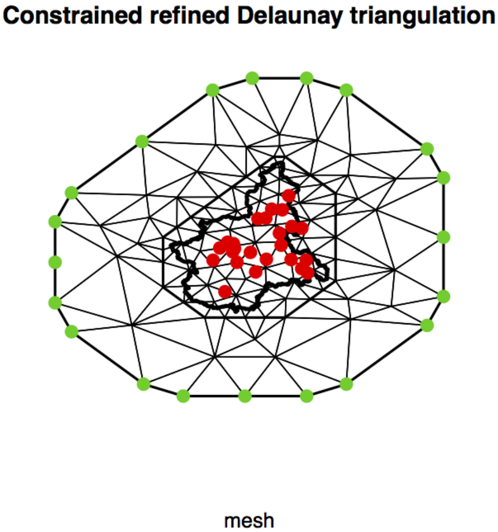
Step 1: Make the mesh

```
mesh =
     inla.mesh.2d(points =NULL,
                           points.domain=borders,
                           offset=c(10, 140),
                           max.edge=c(40,1000),
                           min.angle=21,
                           cutoff=0,
                           plot.delay=NULL
                           )

boundary = inla.mesh.boundary(mesh)[[1]]

nmesh = mesh$n
#select (the rows of) the position of the stations
mesh.idx = 1:nmesh
```

A mesh



Step 2: Make the latent model

In order to construct a kronecker product model in INLA, we use the (experimental) group feature

```
spde = inla.create.spde(mesh,model="matern")

formula = y ~  WS + HMIX +...
                    + intercept + f(field, model=spde,
                       group =time,
                       control.group=list(model="ar1")+
                     )
```

- This tells INLA that the observations are grouped in a certain way.
- `control.group` contains the grouping model (only `ar1` and `exchangable`) as well as their prior specifications.
- NB: intercept!

Step 3: Make an A matrix

There are two ways to construct the A matrix: A for loop or an inbuilt function.

```
LocationMatrix = inla.spde.make.A(mesh = mesh,
        loc =dataLoc, group=time, n.group=nT)
```

This locates the data points in each `group=time` level and stacks the corresponding local A matrices in an appropriate way.

Step 4: Organising the data

We have a problem: we have the covariates at the data points, but the latent field only defined their through the A matrix.

*We need to make sure that **A** only applies to the random effect.*

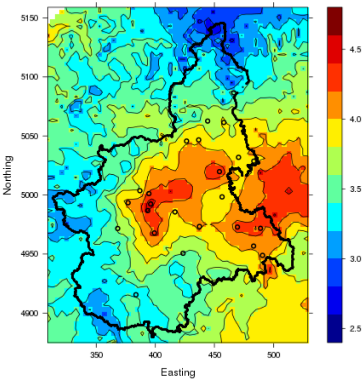
Solution: Padding by `NA`s.

Step 5: Organising the data with `inla.stack`

We can now put everything together.

```
stack = inla.stack( data = dat,
        A = list(1, LocationMatrix),
        effects = list( list(WS = cov$WS,...),
            c(inla.spde.make.index("mesh.idx",n.field=nmesh,
                   n.group=T),
                       list(intercept=rep(1,mesh$n*nT)))
        )
     )
result = inla(formula, family = "gaussian",
        data=inla.stack.data(stack).
        control.predictor = list(A=inla.stack.A(stack)),
        verbose=TRUE)
```

Posterior mean PM10 concentration for 30/01/2006 (log scale)



Outline

But did we answer the question?

- The question was not *fit a space-time surface*
- The limit value fixed by the European directive 2008/50/EC for PM_{10} is $50 \mu g/m^3$. The daily mean concentration cannot exceed this value more than 35 days in a year.
- The question was "Does the PM-10 concentration exceed the EU-mandated maximum levels?"
- So can we get the answer to this question?

Multiple comparisons

- The easiest thing is to compute, for each point, the probability of exceeding the threshold
- We can do that with `inla.pmarginal`
- But this is bad...
- We want areas where *everything* exceeded the level... multiple comparisons
- These sets are called *excursion* sets

Excursions and INLA

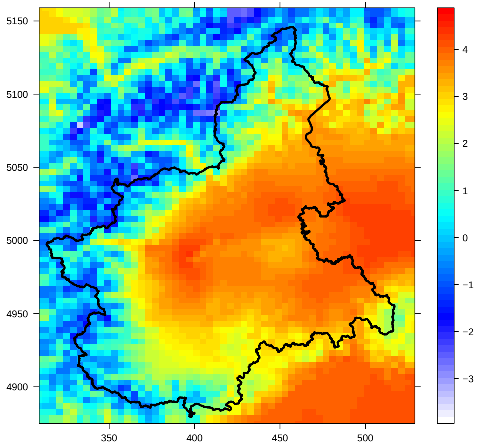
David Bolin (Chalmers) wrote an R package called `excursions` that works with INLA to solve this problem.

- It's pretty easy to use

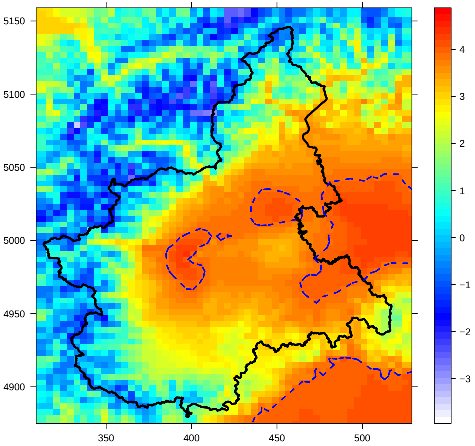
```
excursions.inla(result.inla, ind=indices, alpha=0.99,
        u=0, method='QC', type='>' )
```

- `result.inla` is the output from INLA
- You need to run INLA with the option `control.compute=list(config=TRUE)`
- `ind=indices` tells it which indices of the model you're interested in
- `u` and `alpha` are the level and the confidence
- `type=">"` says you want the set of things above level `u`
- `method='QC'` tells the function how to deal with the non-Gaussianity

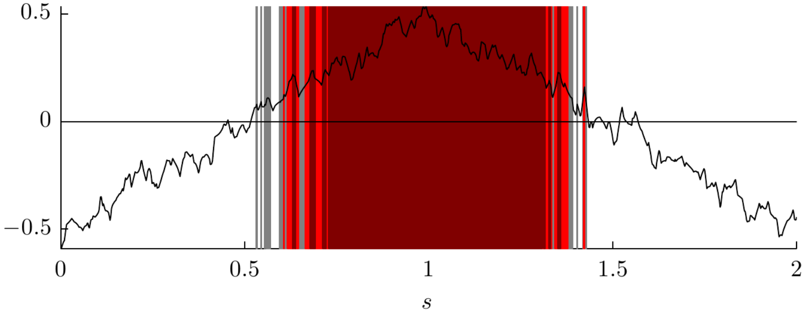
PM_{10} in Piemonte: Where is $PM_{10} > 50$?



PM_{10} in Piemonte: Where is $PM_{10} > 50$? Uncertainty?



Example 1: Gaussian process with exponential covariance

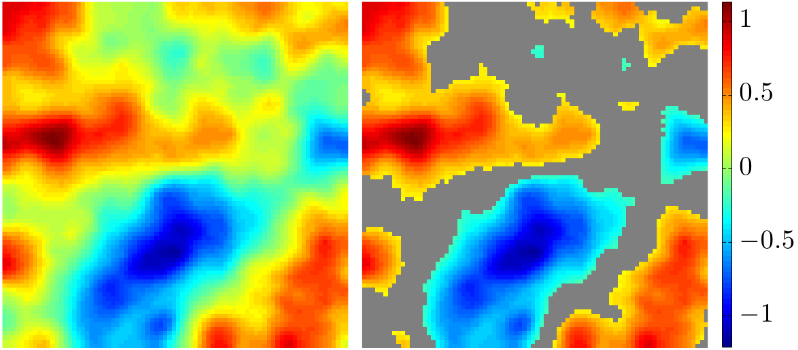


- Gaussian process with exponential covariance function.
- The 95% excursion set is shown in red.
- The grey area contains $\{s : \Pr(x(s) > 0) > 0.95\}$.
- The dark red set is the Bonferroni lower bound.
- The black curve is the kriging estimate of $x(s)$.

Contours and excursions

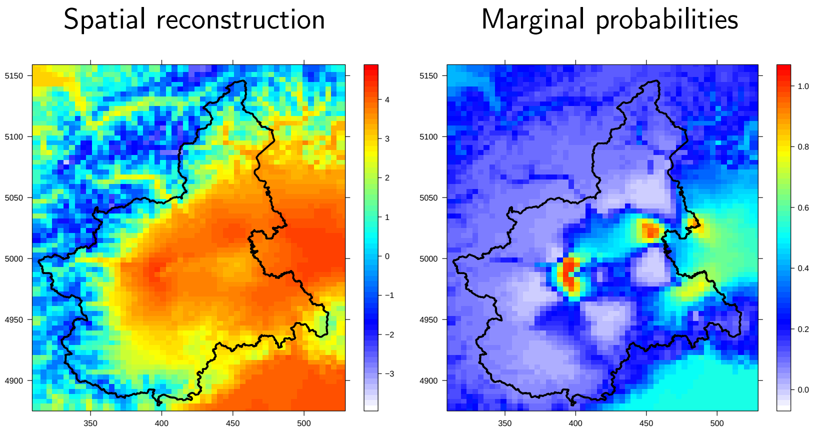
- A contour curve of a reconstructed field can (almost) be found from the pointwise marginal distributions.
- But they are uncertain...
- The *uncertainty* depends on the full joint distribution.
- A credible contour region is a region where the field transitions from being clearly below, to being clearly above.
- This is the same problem as the excursion problem

Example 2: Gaussian Matérn field

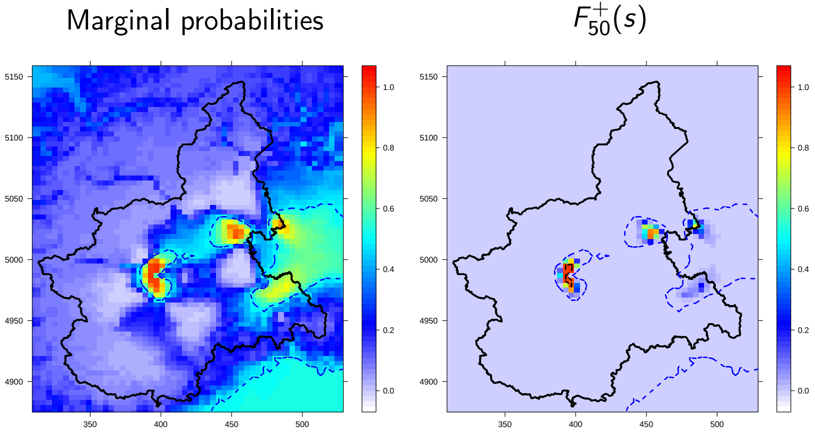


- Gaussian Matérn field measured under Gaussian noise.
- Left panel shows the kriging estimate,
- The grey block on the right is the 95% *contour* for the zero level
- i.e. The field is, with high probability, equal to zero somewhere in that region.

PM-10: January 30, 2006



PM-10: January 30, 2006



PM-10: January 30, 2006

Contour function $F^c_{50}(s)$ Signed avoidance $\pm F_{50}(s)$