Lecture 7

Introduction to spatial modelling – geo-referenced data

Janine Illian
University of Glasgow
janine.illian@glasgow.ac.uk

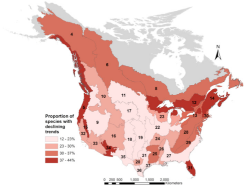
November 13, 2025

Outline

- to come

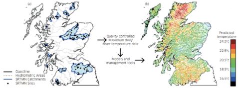
We can distinguish three types of spatial data structures

Areal data



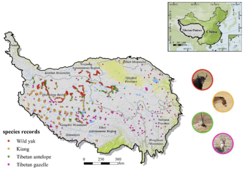
Map of bird conservation regions (BCRs) showing the proportion of bird species within each region showing a declining trend

Geostatistical data



Scotland river temperature monitoring network

Point-referenced data



Occurrence records of four ungulate species in the Tibet,

Types of spatial data

Recall:

Discrete space:

- Data on a spatial grid (areal data)

Continuous space:

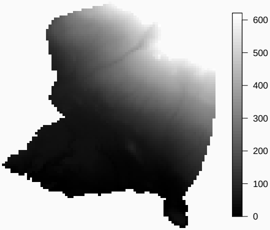
- **Geostatistical (geo-referenced) data**
- Spatial point pattern data

Model components are used to reflect spatial dependence structures in discrete and continuous space.

spatial data – continuous case

recall: geostatistical data

- phenomenon that is continuous in space
- examples: nutrient levels in soil, salinity in the sea, altitude
- measurements in only a finite number of locations



aim: estimate the continuous field

- continuous random variable, a **random field**: function in space with values in continuous space
- **Gaussian random field**
- characterised by mean and covariance (function)

geostatistical data

- we cannot observe everywhere in space

geostatistical data

- we cannot observe everywhere in space

$\rightsquigarrow$ we'd like to predict into unobserved locations

, $\omega(s)$, where s is a location in the area of interest

- we cannot represent every locations in the computer

geostatistical data

- we cannot observe everywhere in space

$\rightsquigarrow$ we'd like to predict into unobserved locations

$\rightsquigarrow$ we need some "look-up device" – this will be the **Gaussian random field (GRF)**, $\omega(s)$, where s is a location in the area of interest

- we cannot represent every locations in the computer

$\rightsquigarrow$ we need to approximate

geostatistical data

- we cannot observe everywhere in space

$\rightsquigarrow$ we'd like to predict into unobserved locations

$\rightsquigarrow$ we need some "look-up device" – this will be the **Gaussian random field (GRF)**, $\omega(s)$, where s is a location in the area of interest

- we cannot represent every locations in the computer

$\rightsquigarrow$ we need to approximate how do we best approximate the $w(s)$, GRF)?

$\rightsquigarrow$ flexibly and efficiently using the **SPDE** approach

geostatistical data

- we cannot observe everywhere in space

$\rightsquigarrow$ we'd like to predict into unobserved locations

$\rightsquigarrow$ we need some "look-up device" – this will be the **Gaussian random field (GRF)**, $\omega(s)$, where s is a location in the area of interest

- we cannot represent every locations in the computer

$\rightsquigarrow$ we need to approximate how do we best approximate the $w(s)$, GRF)?

$\rightsquigarrow$ flexibly and efficiently using the **SPDE** approach

Gaussian random fields

we have a process that is occurring everywhere in space

Gaussian random fields

we have a process that is occurring everywhere in space
$\rightsquigarrow$ natural to try to model it using some sort of function (of space),
a random field $\omega(s)$ $\rightsquigarrow$ a random field is a random function...

a random field is a random variable that generates smooth surfaces

Gaussian random fields

we have a process that is occurring everywhere in space
$\rightsquigarrow$ natural to try to model it using some sort of function (of space),
a random field $\omega(s)$ $\rightsquigarrow$ a random field is a random function...

a random field is a random variable that generates smooth surfaces

- This is hard...
- We typically make our lives easier by making everything **Gaussian**, i.e. following the normal distribution

Gaussian random fields

we have a process that is occurring everywhere in space
$\rightsquigarrow$ natural to try to model it using some sort of function (of space), a random field $\omega(s)$ $\rightsquigarrow$ a random field is a random function...

a random field is a random variable that generates smooth surfaces

- This is hard...
- We typically make our lives easier by making everything **Gaussian**, i.e. following the normal distribution
- What makes a function Gaussian...?

Gaussian random fields

If we are trying to model $\omega(s)$ what sort of things do we need?

- We don't ever observe a function *everywhere*

Gaussian random fields

If we are trying to model $\omega(s)$ what sort of things do we need?

- We don't ever observe a function *everywhere*
- If ω is a vector of observations of $\omega(s)$ at different locations, we want this to be normally distributed:

$$\omega = (\omega(s_1), \ldots, \omega(s_p))^T \sim N(0, \Sigma_{\omega(s_1), \ldots, \omega(s_p)})$$

Gaussian random fields

If we are trying to model $\omega(s)$ what sort of things do we need?

- We don't ever observe a function *everywhere*
- If ω is a vector of observations of $\omega(s)$ at different locations, we want this to be normally distributed:

$$\omega = (\omega(s_1), \ldots, \omega(s_p))^T \sim N(0, \Sigma_{\omega(s_1),\ldots,\omega(s_p)})$$

- This is actually quite tricky: the covariance matrix Σ has to behave well (or be "positive definite")

Gaussian random fields

If we are trying to model $\omega(s)$ what sort of things do we need?

- We don't ever observe a function *everywhere*
- If ω is a vector of observations of $\omega(s)$ at different locations, we want this to be normally distributed:

$$\omega = (\omega(s_1), \ldots, \omega(s_p))^T \sim N(0, \Sigma_{\omega(s_1),\ldots,\omega(s_p)})$$

- This is actually quite tricky: the covariance matrix Σ has to behave well (or be "positive definite")
- use a **covariance function** that depends on the distance between two points and that

Gaussian random fields

If we are trying to model $\omega(s)$ what sort of things do we need?

- We don't ever observe a function *everywhere*
- If ω is a vector of observations of $\omega(s)$ at different locations, we want this to be normally distributed:

$$\omega = (\omega(s_1), \ldots, \omega(s_p))^T \sim N(0, \Sigma_{\omega(s_1),\ldots,\omega(s_p)})$$

- This is actually quite tricky: the covariance matrix Σ has to behave well (or be "positive definite")
- use a **covariance function** that depends on the distance between two points and that
  - has no negative variances
  - is symmetric
  - is decreasing, with maximum at distance $= 0$
- Not every function will ensure that Σ is positive definite!

Gaussian random fields

If we are trying to model $\omega(s)$ what sort of things do we need?

- We don't ever observe a function *everywhere*
- If ω is a vector of observations of $\omega(s)$ at different locations, we want this to be normally distributed:

$$\omega = (\omega(s_1), \ldots, \omega(s_p))^T \sim N(0, \Sigma_{\omega(s_1),\ldots,\omega(s_p)})$$

- This is actually quite tricky: the covariance matrix Σ has to behave well (or be "positive definite")
- use a **covariance function** that depends on the distance between two points and that
  - has no negative variances
  - is symmetric
  - is decreasing, with maximum at distance $= 0$
- Not every function will ensure that Σ is positive definite!

the SPDE approach explained...

a different approach to GRF models:

the SPDE approach explained...

a different approach to GRF models:

- forget about the covariance function!

the SPDE approach explained...

a different approach to GRF models:

- forget about the covariance function!
- express spatial structure through an SPDE (stochastic partial differential equation):

the SPDE approach explained...

a different approach to GRF models:

- forget about the covariance function!
- express spatial structure through an SPDE (stochastic partial differential equation):
    - a simple differential equation describes change in time

the SPDE approach explained...

a different approach to GRF models:

- forget about the covariance function!
- express spatial structure through an SPDE (stochastic partial differential equation):
    - a simple differential equation describes change in time
    - a partial differential equation describes change in space (in more than 1 direction!)

the SPDE approach explained...

a different approach to GRF models:

- forget about the covariance function!
- express spatial structure through an SPDE (stochastic partial differential equation):
  - a simple differential equation describes change in time
  - a partial differential equation describes change in space (in more than 1 direction!)
  - a stochastic partial differential equation makes this random, i.e. not deterministic

the SPDE approach explained...

a different approach to GRF models:

- forget about the covariance function!
- express spatial structure through an SPDE (stochastic partial differential equation):
    - a simple differential equation describes change in time
    - a partial differential equation describes change in space (in more than 1 direction!)
    - a stochastic partial differential equation makes this random, i.e. not deterministic
- approximate space in a clever way...

the SPDE approach explained...

a different approach to GRF models:

- forget about the covariance function!
- express spatial structure through an SPDE (stochastic partial differential equation):
  - a simple differential equation describes change in time
  - a partial differential equation describes change in space (in more than 1 direction!)
  - a stochastic partial differential equation makes this random, i.e. not deterministic
- approximate space in a clever way...

the mesh!

the SPDE approach – more flexible models

- simple models use a simple gridding approach to approximate the continuous spatial field
- this is easy to implement
- however: this can be
  - **computationally inefficient** and
  - not flexible enough (complicated boundaries or domains)

$\Rightarrow$ use continuously specified finite dimensional Gaussian random fields

$\Rightarrow$ spatial field as solution to a stochastic partial differential equation ("SPDE approach")

continuous specification

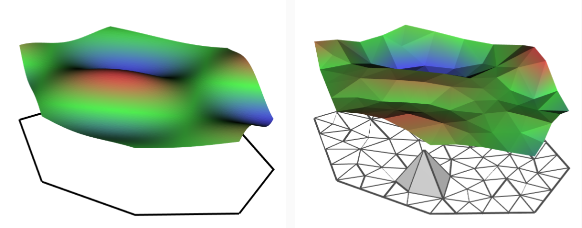
- use a **continuous** specification of the random field model:

continuous specification

- use a **continuous** specification of the random field model: a finite-dimensional basis function expansion

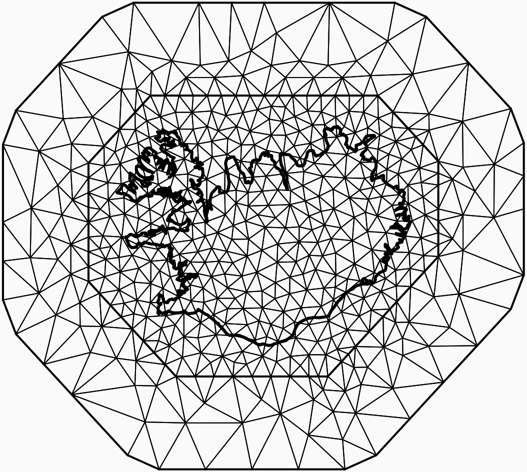
continuous specification

- use a **continuous** specification of the random field model: a finite-dimensional basis function expansion in the "mesh" - a triangulation of space



  - e.g. for point processes: no "binning" of the points
- allows computation using the **exact** positions of the points...

A mesh...



benefits of the SPDE approach...

After all those technicalities... here's the important bit:

benefits of the SPDE approach...

After all those technicalities... here's the important bit: The SPDE approach yields

- more flexible modelling; it is easier to
  - build more general models
  - work with changing observation areas over time
  - work with “funny” observation areas
- we don't need to worry about covariance functions...

Part of the magic: SPDE models are still GMRF!

$\Rightarrow$ we can still use INLA to fit these and it is still fast!

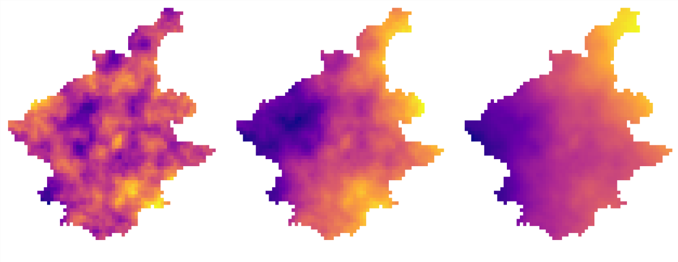
SPDE models

We call spatial Markov models defined on a mesh *SPDE models*.

SPDE models have 3 parts

- a mesh
- a range parameter ρ
- a variance parameter σ^2 (or precision parameter τ)

Different realisations of the same SPDE model with varying range parameter κ.



And now...?

What do we need the SPDE models for?

And now...?

What do we need the SPDE models for?
recall: "Model components are used to reflect spatial dependence structures in discrete and continuous space."

And now...?

What do we need the SPDE models for?

recall: "Model components are used to reflect spatial dependence structures in discrete and continuous space."

spatial structures in a model are explained by

- covariate(s) – relationship between covariate and response might be of interest

And now...?

What do we need the SPDE models for?

recall: "Model components are used to reflect spatial dependence structures in discrete and continuous space."

spatial structures in a model are explained by

- covariate(s) – relationship between covariate and response might be of interest
- an SPDE model – to account for additional dependence for valid inference

And now...?

What do we need the SPDE models for?
recall: "Model components are used to reflect spatial dependence structures in discrete and continuous space."
spatial structures in a model are explained by

- covariate(s) – relationship between covariate and response might be of interest
- an SPDE model – to account for additional dependence for valid inference

Here:

$$y(s)|\eta(s) \sim \text{Binom}(1, p(s))$$

$$\eta(s) = \text{logit}(p(s)) = \beta_0 + \omega(s) + \beta_1 \text{ depth}(s)$$

prior choice for GRFs

priors determine the smoothness of the random field

- if the field is too smooth, spurious significance
- if the field is too wiggly, overfitting

$\Rightarrow$ choice has to be done very carefully

prior choice for GRFs

priors determine the smoothness of the random field

- if the field is too smooth, spurious significance
- if the field is too wiggly, overfitting

$\Rightarrow$ choice has to be done very carefully

approaches (in the spatial stats literature):

- choose an arbitrary prior that your friend used and liked...

prior choice for GRFs

priors determine the smoothness of the random field

- if the field is too smooth, spurious significance
- if the field is too wiggly, overfitting

$\Rightarrow$ choice has to be done very carefully

approaches (in the spatial stats literature):

- choose an arbitrary prior that your friend used and liked...
- play around – choose some arbitrary prior value and change it and check what happens until you are happy

prior choice for GRFs

priors determine the smoothness of the random field

- if the field is too smooth, spurious significance
- if the field is too wiggly, overfitting

$\Rightarrow$ choice has to be done very carefully

approaches (in the spatial stats literature):

- choose an arbitrary prior that your friend used and liked...
- play around – choose some arbitrary prior value and change it and check what happens until you are happy
- believe in some default prior and trust it blindly

all arbitrary...

as a software developer...

options:

1) provide some standard prior distributions

- e.g. a log-gamma distribution; user chooses parameters

as a software developer...

options:

1) provide some standard prior distributions

- e.g. a log-gamma distribution; user chooses parameters

$\Rightarrow$ BUT: those parameters do not have a direct meaning...

as a software developer...

options:

1) provide some standard prior distributions

- e.g. a log-gamma distribution; user chooses parameters

$\Rightarrow$ BUT: those parameters do not have a direct meaning...

2) to make things informed (non-arbitrary) and still user friendly – **automate**:

- choose a criterion (e.g. "degree of wigglyness") and penalise violation of that criterion in an automated way

as a software developer...

options:

1) provide some standard prior distributions

- e.g. a log-gamma distribution; user chooses parameters

$\Rightarrow$ BUT: those parameters do not have a direct meaning...

2) to make things informed (non-arbitrary) and still user friendly – **automate**:

- choose a criterion (e.g. "degree of wigglyness") and penalise violation of that criterion in an automated way

$\Rightarrow$ BUT: we don't know what the ideal smoothness/wigglyness is...

as a software developer...

here: re-think priors:

we want to automate, but:

as a software developer...

here: re-think priors:

we want to automate, but:

penalise something different: deviation from a base model

aim: make prior choice transparent and problem-driven

penalised complexity priors

Occam's razor: Prefer simpler model until there is enough support for a more complex model.

penalised complexity priors

Occam's razor: Prefer simpler model until there is enough support for a more complex model.

- model component is flexible version of a **base model**
- penalise deviation from base model (using Kullback-Leibler divergence, measuring information loss)
- "shrink" towards the base model – the data really have to "convince" us that the more complicated model is necessary

penalised complexity priors

Occam's razor: Prefer simpler model until there is enough support for a more complex model.

- model component is flexible version of a **base model**
- penalise deviation from base model (using Kullback-Leibler divergence, measuring information loss)
- "shrink" towards the base model – the data really have to "convince" us that the more complicated model is necessary
- the maths says that having a prior on this parameterisation will avoid overfitting
- we are penalising unnecessary model complexity
- also: interpretable priors and **flexible modelling**

make prior choice transparent

penalise deviation from a base model in our context...

make prior choice transparent

penalise deviation from a base model in our context...

The base model:

- model with intercept and covariates, no (or rather a very flat) Gaussian random field

$\Rightarrow$ the approach is conservative about including a random field – penalises overfitting

make prior choice transparent

penalise deviation from a base model in our context...

The base model:

- model with intercept and covariates, no (or rather a very flat) Gaussian random field

$\Rightarrow$ the approach is conservative about including a random field – penalises overfitting

$\Rightarrow$ we set a prior on the spatial scale that we consider overfitting
$\Rightarrow$ we have another prior that reflects how confident we are about this scale

pc prior for range

prior value that reflects the spatial scale that we consider overfitting

prior value that reflects how confident we are about this scale

How do we do this?

pc prior for range

prior value that reflects the spatial scale that we consider overfitting

prior value that reflects how confident we are about this scale

How do we do this?

Use `inla.spde2.pcmatern` with prior parameters:

```
spde_model1 =  inla.spde2.pcmatern(mesh,
                                   prior.sigma = c(.1, 0.5),
                                   prior.range = c(30, 0.5))
```

pc prior for range

prior value that reflects the spatial scale that we consider overfitting
prior value that reflects how confident we are about this scale

How do we do this?

Use `inla.spde2.pcmatern` with prior parameters:

```
spde_model1 =  inla.spde2.pcmatern(mesh,
                                   prior.sigma = c(.1, 0.5),
                                   prior.range = c(30, 0.5))
```

pc prior for range

```r
spde_model1 =  inla.spde2.pcmatern(mesh,
                                   prior.sigma = c(.1, 0.5),
                                   prior.range = c(30, 0.5))
```

Let's look at:

`prior.range=c(range0, Prange)`:

pc prior for range

```
spde_model1 =  inla.spde2.pcmatern(mesh,
                                    prior.sigma = c(.1, 0.5),
                                    prior.range = c(30, 0.5))
```

Let's look at:

`prior.range=c(range0, Prange)`:

The probability that the range will be $<$ `range0` is `Prange`.

pc prior for range

```
spde_model1 =  inla.spde2.pcmatern(mesh,
                                   prior.sigma = c(.1, 0.5),
                                   prior.range = c(30, 0.5))
```

Let's look at:

`prior.range=c(range0, Prange)`:

The probability that the range will be $<$ `range0` is `Prange`.

- if we make `Prange` small, we are saying here is that we want to avoid the range to be smaller than a certain value (range0) with a very small probability

pc prior for range

```
spde_model1 =  inla.spde2.pcmatern(mesh,
                                   prior.sigma = c(.1, 0.5),
                                   prior.range = c(30, 0.5))
```

Let's look at:

`prior.range=c(range0, Prange)`:

The probability that the range will be $<$ `range0` is `Prange`.

- if we make `Prange` small, we are saying here is that we want to avoid the range to be smaller than a certain value (range0) with a very small probability
- i.e. we are pretty sure that a range of this size or smaller is overfitting