01-SPDEtoy-INLA-SPDE

Frances Lin

3/28/2022

Again, explanations are a bit confusing but we will follow the example for the purpose of learning the **R** package `R-INLA`.

Load packages

```
library(INLA)
```

```
## Loading required package: Matrix

## Loading required package: foreach

## Loading required package: parallel

## Loading required package: sp

## This is INLA_22.03.16 built 2022-03-16 13:24:07 UTC.
##  - See www.r-inla.org/contact-us for how to get help.
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x purrr::accumulate() masks foreach::accumulate()
## x tidyr::expand()     masks Matrix::expand()
## x dplyr::filter()     masks stats::filter()
## x dplyr::lag()        masks stats::lag()
## x tidyr::pack()       masks Matrix::pack()
## x tidyr::unpack()     masks Matrix::unpack()
## x purrr::when()       masks foreach::when()
```

```
library(pander)
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

Simulate or load data

It'll be helpful to walk through the simulation section.

```
# Simulate data
# https://becarioprecario.bitbucket.io/spde-gitbook/ch-intro.html#sec:simulatoy
n <- 200 #5
set.seed(123)
pts <- cbind(
  s1 = sample(1:n / n - 0.5 / n)^2,
  s2 = sample(1:n / n - 0.5 / n)^2)
```

```
# Get a (lower triangular) matrix of distances
dmat <- as.matrix(dist(pts))
#dmat
```

```
# Set parameters
beta0 <- 10       # for mean
sigma2e <- 0.3    # for the nugget
sigma2u <- 5      # for the Matérn covariance
kappa <- 7        # for the Matérn covariance
nu <- 1           # for the Matérn covariance
```

```
# Create a function cMatern to compute the Matérn covariance of two points at distance h
# https://becarioprecario.bitbucket.io/spde-gitbook/ch-intro.html#sec:matern
cMatern <- function(h, nu, kappa) {
  ifelse(h > 0, besselK(h * kappa, nu) * (h * kappa)^nu /
    (gamma(nu) * 2^(nu - 1)), 1)
}
```

```
# Get a Matérn covariance of the spatial process
mcor <- cMatern(dmat, nu, kappa)
mcov <- sigma2e * diag(nrow(mcor)) + sigma2u * mcor
```

```
# Get the samples
R <- chol(mcov)
set.seed(234)
y1 <- beta0 + drop(crossprod(R, rnorm(n)))
```

The values differ a bit but it's fine. Let's stick to the data `SPDEtoy` for consistency purposes.

```
# Put them in a df
SPDEtoy_sim <- tibble(
  s1 = pts[, 1],
  s2 = pts[, 2],
  y = y1
)
```

Load data

```
class(SPDEtoy)
```

```
## [1] "data.frame"
```

```
SPDEtoy %>% head %>% pander
```

s1	s2	y
0.08266	0.05641	11.52
0.6123	0.9168	5.278
0.162	0.357	6.903
0.7526	0.2576	13.18
0.851	0.1541	14.6
0.001806	0.7353	9.78

The SPDE (Stochastic Partial Differential Equations) model

The SPDE model can be defined

$$y|\beta_0, u, \sigma_e^2 \sim N(\beta_0 + Au, \sigma_e^2)$$
$$u \sim GF(0, \Sigma),$$

where β_0 is the intercept, A is the projector matrix that links the spatial Gaussian random field to the locations of the observed data, and u is a spatial Gaussian random field.

I am unable to follow the rest of the sections.

Reference

2.1.3 The Matérn covariance.

2.3 A toy example.