

Bayesian Inference for Multivariate Spatial Models with INLA

by Francisco Palmí-Perales, Virgilio Gómez-Rubio, Roger S. Bivand, Michela Cameletti, and Håvard Rue

Abstract Bayesian methods and software for spatial data analysis are well-established now in the broader scientific community generally and in the spatial data analysis community specifically. Despite the wide application of spatial models, the analysis of multivariate spatial data using the integrated nested Laplace approximation through its R package (R-INLA) has not been widely described in the existing literature. Therefore, the main objective of this article is to demonstrate that R-INLA is a convenient toolbox to analyse different types of multivariate spatial datasets. This will be illustrated by analysing three datasets which are publicly available. Furthermore, the details and the R code of these analyses are provided to exemplify how to fit models to multivariate spatial datasets with R-INLA.

1 Introduction

In recent times, spatial data analysis has increased in popularity due to the widespread availability of different types of geo-referenced data. Spatial models will exploit the geographical information in the data so that spatial dependence is exploited, among other things, to build better predictive models. There is also a growing interest (and application) of multivariate models as often spatial data incorporates multiple variables. For instance, healthcare, econometrics, climate and other fields require multivariate models. One of the main advantages of multivariate models is their ability to find similar spatial patterns in the different response variables. This is the main motivation of this work.

Analysing spatial data poses a number of methodological and computational challenges. In this regard, Bayesian inference has been particularly successful as it has provided data analysts with both a modelling framework and estimation methods for spatial data analysis (see, for example, Banerjee et al., 2014). Bayesian inference for spatial models has often relied upon Markov Chain Monte Carlo (MCMC, Gilks et al., 1995) algorithms, which provide samples from the joint posterior distribution of the model parameters. However flexible, MCMC algorithms have proved slow when dealing with large datasets or highly parameterized models.

For this reason, the integrated nested Laplace approximation (INLA, Rue et al., 2009) has become an alternative approach to fit Bayesian hierarchical models. The advantage of INLA is its ability to approximate the marginal posterior distribution of the different hyperparameters and latent effects of a model that could be parametrised as a latent Gaussian model (Rue and Held, 2005) in less time than MCMC algorithms. This methodology has been implemented in the INLA R package. To avoid confusion between the INLA methodology and the INLA R package, we refer to the package as R-INLA (Rue et al., 2020).

The different spatial models which can be fit with R-INLA have been summarised by several authors (Lindgren et al., 2015; Blangiardo and Cameletti, 2015; Bakka et al., 2018). For a recent review, the reader is referred to Chapter 7 of Gómez-Rubio (2020). Krainski et al. (2019) provide an exhaustive tutorial about how to fit advanced spatial models in R-INLA using the stochastic partial differential equation approach (SPDE, Lindgren et al., 2011) to estimate continuous processes. Additionally, INLA can be combined with other algorithms such as Markov Chain Monte Carlo (MCMC, Gilks et al., 1995) techniques in order to fit models which can not be fit solely with INLA (Gómez-Rubio and Palmí-Perales, 2019).

Therefore, the main objective of this article is to describe how to analyse any multivariate spatial dataset using R-INLA in a Bayesian framework. Multivariate spatial models have been studied by several authors. For example, Van Lieshout and Baddeley (1999) describe dependence between multivariate point patterns by proposing novel summary statistics. Additionally, multivariate log-Gaussian Cox processes have been used to analyse multivariate point patterns (Diggle et al., 2013; Waagepetersen et al., 2016; Gómez-Rubio et al., 2015). Furthermore, several studies have been published for analysing multivariate lattice data. For example, in MacNab (2018) an insight into the generalization of univariate models to multivariate models is extensively discussed and in Martínez-Beneito et al. (2017) a framework to analyse multiple response variables is proposed in the context of disease mapping. A review of the multivariate spatial models in disease mapping is performed in Martínez-Beneito

and Botella-Rocamora (2019). These works rely on MCMC methods to develop their examples. In the context of spatial modeling, these methods can lead to a high computational burden.

The R programming (R Core Team, 2022) language offers a wide range of standalone packages for analysing spatial datasets. Several of them focus on a particular type of spatial data. For instance, point pattern analysis can be performed with `spatstat` (Baddeley et al., 2015) and `spatialkernel` (Gómez-Rubio et al., 2017). Geostatistical data can be modeled using `gstat` (Pebesma and Wesseling, 1998; Pebesma, 2004) or `spBayes` (Finley et al., 2007, 2015). Other R packages such as `CARBayes` (Lee, 2013) are designed to analyse lattice data.

The remainder of the manuscript is organised as follows. First, a short introduction to the INLA methodology and how the **R-INLA** package fits models to multivariate data is provided. Next, a brief description of the different multivariate models is detailed in the case of areal, continuous and point pattern data in three different sections. Each of these sections discusses the models to be fit, including the prior distribution choice and the structure of the data, and includes an example on how to analyse multivariate spatial data with **R-INLA**. Finally, a brief summary of the conclusions of this work appears in the last part of this manuscript.

2 Integrated nested Laplace approximation

Bayesian inference has been usually performed with MCMC. However, these methods have a high computational cost in a spatial modelling context, mainly because of the simulation-based approach they use to estimate the large number of latent effects present in spatial models. A good alternative is the integrated nested Laplace approximation (INLA, Rue et al., 2009) as it is able to accurately and efficiently approximate posterior distributions of high-dimensional models in a short computing time.

The INLA methodology focuses on the analysis of latent Gaussian models (Rue and Held, 2005). In essence, a latent Gaussian model is a Bayesian hierarchical model in which the mean parameter of the likelihood is linked to a linear predictor composed of different additive latent effects; the distribution of the vector of latent effects follows a multivariate Gaussian distribution (more details can be found in, for example, Chapter 2 of Rue and Held, 2005). Latent Gaussian models comprise a large group of statistical models such as many spatial and spatio-temporal models. The **R-INLA** package implements the INLA method in a flexible way, so that models with a number of latent effects can be fit. Additionally, INLA can be combined with other algorithms in order to fit other models (Gómez-Rubio and Palmí-Perales, 2019).

Gaussian Markov random fields (GMRF) with sparse precision matrices can be used to approximate Gaussian latent fields due to their conditional independence properties. The sparse precision matrices allow computationally efficient methods (Rue and Held, 2005). This is the reason of the low computational costs of INLA which is one of its main benefits. More details about how to implement Bayesian inference with INLA can be found in (Gómez-Rubio, 2020).

Model fitting with **R-INLA** is done via the `inla()` function, which works like the `glm()` and `gam()` functions. The model to fit is defined using a formula (that may include different types of random effects in the right-hand side). For example, to fit a Poisson regression with response variable y and covariate x , both in data frame d , the code is:

```
inla(y ~ x, family = "poisson", data = d)
```

The output includes the posterior marginal distribution of the model parameters, as well as summary statistics computed from these distributions. In its example, default priors have been used, but **R-INLA** can work with a good number of prior distributions.

In the previous example, y represents the vector of observations of a single variable. However, INLA can deal with multivariate responses so that they can be analyzed jointly. Fitting models with a multivariate responses requires the storage of variables in a particular format. Let us consider the simplest scenario in which a dataset with D variables have been measured in a study region divided into n areas. In order to analyse this dataset with **R-INLA**, a matrix with D columns and $D \times n$ rows has to be built. Specifically, this matrix will store the n values of the first variable in the first column, from the first to the n -th row, then the following values of this column will be NA. Then, the data of the

second variable would be placed in the second column between the $n + 1$ and the $2n$ rows. The rest of the values of the second column will be filled with NA's. Data for other variables will be added to the matrix accordingly.

As a toy example, we will take $n = 2$ and $D = 3$, so that the original dataset is structured as a 2×3 matrix given by

$$\begin{bmatrix} 1 & 4 & 3 \\ 2 & 6 & 5 \end{bmatrix}.$$

Following the above procedure, the matrix that should be passed to **R-INLA** would be the following:

$$\begin{bmatrix} 1 & \text{NA} & \text{NA} \\ 2 & \text{NA} & \text{NA} \\ \text{NA} & 4 & \text{NA} \\ \text{NA} & 6 & \text{NA} \\ \text{NA} & \text{NA} & 3 \\ \text{NA} & \text{NA} & 5 \end{bmatrix}.$$

This data format must be used in the case of the multivariate response but also for multivariate covariates. In particular, multivariate covariates will be included in the model so that values in a particular column only affect values in the same column of the response, i.e., the model will include a specific coefficient for each variable in the response. In the R code provided together with this paper (as a supplementary material), we have provided function `create_multivariate_data()` that will take a matrix or several vectors of data and create the appropriate matrix for R-INLA. When the covariate is expected to have the same effect for all the response variables (i.e., the coefficient will be the same), a single vector with the values of the covariates will be used. This is better explained in the examples that follow.

3 Multivariate Lattice Data

Areal (or lattice) data are obtained when the spatial data are observed on regions with defined boundaries. In this case, the domain is divided into (non-overlapping) areas in which the data are collected. It is usually considered that two areas are neighbours if they share a common boundary. This adjacency structure is often included to account for spatial autocorrelation (Banerjee et al., 2014).

When the values of several variables are recorded in each area the resulting data become a multivariate lattice dataset. The joint analysis of the spatial distribution of several variables allows to detect similar (spatial) patterns among some of these variables (Banerjee et al., 2014, Chapter 10) while estimating the spatial effects. We will illustrate the analysis of this type of data using a Poisson regression model, which is commonly employed in spatial epidemiology to analyse count data. Other similar models can be proposed for binary or continuous outcomes.

Given the d -th variable of interest (with $d = 1, \dots, D$) and area i (with $i = 1, \dots, n$), the response of interest $Y_{d,i}$ can be modeled using a Poisson distribution with mean $\mu_{d,i}$:

$$Y_{d,i} \sim \text{Po}(\mu_{d,i}).$$

The mean is usually modeled as a linear predictor made up of several terms and transformed via a convenient link function. The choice of these terms depends on the available data and the analyst's understanding of the data generating process. For instance, one option can be described as

$$\psi(\mu_{d,i}) = \alpha_d + u_i + v_{d,i}.$$

Here, $\psi(\cdot)$ is a link function (e.g., natural logarithm), α_d is a variable-specific intercept, u_i a shared (between all or a group of variables) spatial term, and $v_{d,i}$ a variable-specific random effect. Note that restrictions may be required on $v_{d,i}$ to make all effects identifiable (Rue and Held, 2005).

In the context of disease mapping, the usual variables of interest are the counts of mortality or incidence of different diseases over the study region. Now d represents the specific disease, therefore, following the above structure, the observed number of cases of disease d in area i , $Y_{d,i}$, can be modeled as

$$Y_{d,i} \sim \text{Po}(\mu_{d,i} = E_{d,i} \cdot \theta_{d,i})$$

$$\log(\theta_{d,i}) = \alpha_d + u_i + v_{d,i}$$

where $E_{d,i}$ and $\theta_{d,i}$ are the expected number of cases and the relative risk of disease d in area i , respectively. As before, α_d is a disease specific intercept (to account for differences in the total number of observed cases), u_i a shared spatial term (which does not depend on the disease) and $v_{d,i}$ a disease-specific spatial random effect.

In the case of multivariate lattice data, a prior distribution should be assigned to the dispersion parameter of each spatial effect. Some authors have discussed the most appropriate vague prior distributions in these cases. For instance, [Gelman \(2006\)](#) suggests to avoid inverse Gamma distributions on the precision and propose some alternatives. In this example, flat uniform prior distributions are assigned to the standard deviation parameters (see Section 5.3 in [Gómez-Rubio, 2020](#)).

Several authors (see, for example, [Martínez-Beneito, 2013](#), and the references therein) have proposed different approaches for modelling multiple diseases in space and time. [Gómez-Rubio et al. \(2019\)](#) propose a separable spatio-temporal model with weighted shared components that can be used to detect diseases with similar patterns. In [Palmí-Perales et al. \(2021\)](#), the authors have developed an R package ([INLAMSM](#)) which builds on top of [R-INLA](#) to provide some of the most common multivariate model for lattice data.

3.1 Example: spatial analysis of several diseases

[Gómez-Rubio and Palmí-Perales \(2019\)](#) study the variation of spatial risk of three types of cancer in peninsular Spain. In particular, they consider oral cavity, esophagus and stomach cancer, all at the province level. In order to assess similar spatial variation that may lead to the identification of shared risk factors, models with shared and disease specific spatial patterns can be proposed.

Data are available in a RData file available from GitHub at https://github.com/becarioprecario/INLAMCMC_spatial_examples (see [Gómez-Rubio and Palmí-Perales, 2019](#), for details). The following code creates the response variable (by stacking the three vectors of observed cases) as well as the expected counts. Note that due to the different structure of the variables involved, data are stored in a list object instead of a data.frame. Function `create_multivariate_data()` used below is available in the code provided as supplementary material.

```
# Load data
load("dismap_sim_data.RData")

# Set shorter names
names(OyE.sim)[1:6] <- c("Obs.Cb", "Esp.Cb", "Obs.Eso", "Esp.Eso", "Obs.Est", "Esp.Est")

# Create a dataset for INLA (n x 3)
n <- nrow(OyE.sim)

d <- list(OBS =
  create_multivariate_data(as.data.frame(OyE.sim)[, c("Obs.Cb", "Obs.Eso", "Obs.Est")])
)

# Expected cases
d$EXP <- c(OyE.sim$Esp.Cb, OyE.sim$Esp.Eso, OyE.sim$Esp.Est)
```

As an example, we will consider a model in which the log-relative risk of oral cavity cancer ($\theta_{o,i}$) is modeled as the sum of an intercept (α_o), and a shared spatial term (u_i). An intrinsic conditional autoregressive (ICAR, [Banerjee et al., 2014](#)) is assigned to this shared term following the model described for areal (or lattice) data in the previous section. Furthermore, log-relative risks of esophagus ($\theta_{e,i}$) and stomach cancer ($\theta_{s,i}$) are modeled using disease-specific intercepts (α_e and α_s , respectively) plus the shared spatial term and cancer-specific ICAR spatial terms ($v_{e,i}$ and $v_{s,i}$, respectively). Specifically, the relative risks are defined as follows:

$$\begin{aligned}\log(\theta_{o,i}) &= \alpha_o + u_i \\ \log(\theta_{e,i}) &= \alpha_e + u_i + v_{e,i} \\ \log(\theta_{s,i}) &= \alpha_s + u_i + v_{s,i} \quad i = 1, \dots, n\end{aligned}$$

The disease-specific spatial terms can be used to assess departures from the shared spatial term. The chosen prior distributions for the standard deviation of all the effects are improper flat uniform distributions.

The model formula is defined in the model below. The `rf` term represents the disease-specific intercepts (the value of the above α_s). Latent effects of type copy (see, for example, Section 6.5.1 in Gómez-Rubio, 2020) are used to define shared terms u_i in the model. Furthermore, spatial latent effects have the uniform prior for their standard deviation defined in object `prior.prec`.

```
# Formulas for the model
form <- OBS ~ -1 + rf +
  f(copy1, model = "besag", graph = W, hyper = list(prec = prior.prec)) +
  f(copy2, copy = "copy1", fixed = TRUE) +
  f(copy3, copy = "copy1", fixed = TRUE) +
  f(spatial2, model = "besag", graph = W, hyper = list(prec = prior.prec)) +
  f(spatial3, model = "besag", graph = W, hyper = list(prec = prior.prec))
```

Finally the model is fit with **R-INLA** using the code below. Note how the family argument takes a vector of three elements as this likelihood has three components (one for each disease).

```
res <- inla(formula = form, data = d, family = rep("poisson", 3), E = d$EXP)
```

Figure 1 shows the different spatial terms in the model. The shared term represents the spatial variation in the risk of oral cavity cancer and also serves as a baseline for the other types of cancer. The esophagus-specific spatial term is quite mild, which indicates that these two types of cancer have a very similar spatial pattern. The stomach-specific spatial term shows that some provinces in the center of the country have a higher mortality from stomach cancer as compared to oral cavity/esophagus cancer.

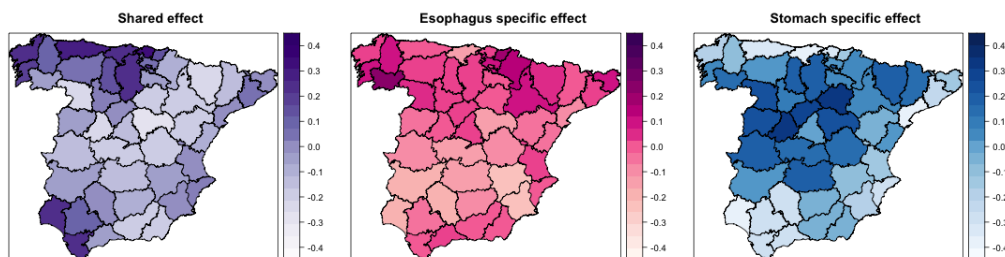


Figure 1: Posterior means of the shared spatial pattern (left), the esophagus-specific spatial pattern (middle) and stomach-specific spatial pattern (right).

4 Multivariate Continuous Data

Datasets for variables that vary continuously in space contain observations which are geographically referenced, i.e., both the value and where it is collected (e.g., the coordinates) appear in the dataset. Then, the spatial variation of the different variables is estimated using geostatistical models.

Similarly to lattice data, geostatistical multivariate models can be fit with R-INLA by sharing common terms. Therefore, the first variable can be modelled so that the mean includes a shared spatial term assumed to be a Gaussian process with a covariance defined using a Matérn function, and all the other variables can depend on this shared spatial term plus specific spatial effects. Hence, for example,

D variables of interest (Y_d for $d = 0, \dots, D - 1$), with a general likelihood function $P(\cdot)$, measured at n different locations can be written as:

$$Y_{i,d} \sim P(\mu_{i,d}).$$

Then, the mean of the baseline variable ($\mu_{i,0}$) will be modelled as a sum of an intercept (α_0) and a shared spatial effect ($u_{i,0}$). Furthermore, the mean of observation i and variable d ($\mu_{i,d}$) will be modeled through an intercept for each variable (α_d), the shared spatial effect ($u_{i,0}$) and a specific spatial effect ($u_{i,d}$) as follows:

$$\begin{aligned} \mu_{i,0} &= \alpha_0 + u_{i,0}; & i &= 1, \dots, n \\ \mu_{i,d} &= \alpha_d + u_{i,0} + u_{i,d} & i &= 1, \dots, n; d = 1, \dots, D - 1 \end{aligned}$$

Here, $u_{i,0}$ represents the shared (between variables) spatial term, while $u_{i,d}$ ($d \geq 1$) represents specific terms that can be used to assess departures from the shared spatial term. These random effect terms are assumed to be Gaussian processes with covariance defined using a Matérn covariance function, which for a generic spatial random effect $u(s)$ is defined as:

$$\text{Cov}(u_p, u_l) = \text{Cov}(u(s_p), u(s_l)) = \frac{\sigma^2}{\Gamma(\lambda)2^{\lambda-1}} (\kappa \|s_p - s_l\|)^\lambda K_\lambda(\kappa \|s_p - s_l\|),$$

where $\|s_p - s_l\|$ represents the Euclidean distance between points s_p and s_l , σ^2 is the marginal variance of the latent Gaussian process, κ is the scaling parameter, which is related to the range, K_λ represents the modified Bessel function of the second kind, and λ is the order term, which measures the smoothness of the process. Furthermore, this latent Gaussian field $u(s)$ can be approximated using the stochastic partial differential equation (SPDE) approach described in Lindgren et al. (2011). This approach relies on expressing the random effect $u(s)$ as the solution of an SPDE which is approximated through (an appropriate choice of) deterministic basis functions defined in a triangulation of the domain:

$$u(s) = \sum_{k=1}^m \phi_k(s) w_k.$$

Here, $\phi_k(s)$ are the basis functions (pairwise linear functions), m is the total number of nodes (triangle vertices) and w_k are zero-mean Gaussian distributed weights. For more details, the reader is referred to Lindgren et al. (2011).

In this example, all the measurements of the different variables are obtained from the same n locations. However, the measurements of each variable can be from different locations of the study region. Furthermore, INLA is not able to estimate any cross-covariance using the proposed model as dependence among the variables relies on the shared spatial term. A multivariate geostatistical analysis can be performed using the R package **gstat**. However, the models implemented in this R package are based on a classic and frequentist statistical approach.

Regarding the priors, the SPDE approximation requires setting a prior distribution to the nominal range, r , and the nominal standard deviation, σ . The nominal range is the distance at which the correlation is 0.1. Penalized complexity prior distributions (PC-priors, Simpson et al., 2017) can be chosen for both parameters. In a nutshell, PC-priors are based on the idea of penalising the complexity from a simple baseline model (that is, a model in which the parameter has been fixed to a particular reference value), i.e., the prior density is related to the distance from a baseline model. A remarkable benefit of the PC-priors are their high intuitiveness in their definition as they are set by stating values for the probabilities of the parameters, in particular, $P(r < r_0)$ and $P(\sigma > \sigma_0)$ for certain values of r_0 and σ_0 .

The prior on the range is set such that r is lower than half the maximum distance (d_m) between any two points in the study region, with high probability, i.e., $P(r < d_m/2) \approx 1$. As for the standard deviation, following the example of Sørbye et al. (2019) we set an upper limit U_α such that σ is less than this limit with high probability. Therefore, the probability that the standard deviation is greater than this upper limit ($P(\sigma > U_\alpha)$) is set to be almost 0.

4.1 Example: spatial distribution of heavy metals

The meuse dataset in the **gstat** package gives the locations and measurements of topsoil heavy metals collected in a flood plain by the Meuse river close to the village of Stein (located in Netherlands). After

loading the **gstat** package, the data can be loaded using the following code:

```
# Load the data
data(meuse)

# Create the spatial object
coordinates(meuse) <- ~ x + y
proj4string(meuse) <- CRS("+init=epsg:28992")
```

These measurements are highly correlated and we will explore in this example how to fit geostatistical models with the SPDE approach. Note that observations do not need to be in a regular grid. Instead of a grid, a mesh is defined to apply the SPDE approach. The boundary of the study region is stored in object `meuse.bdy` (see accompanying code). The definition of the mesh is done using the coordinates of the boundary of the area following the code below:

```
# Create the mesh
mesh <- inla.mesh.2d(boundary = meuse.bdy, loc = coordinates(meuse),
  max.edge = c(250, 500), offset = c(250, 500), n = c(32, 32))
```

The left plot in Figure 2 shows the mesh built with the above code for this example.

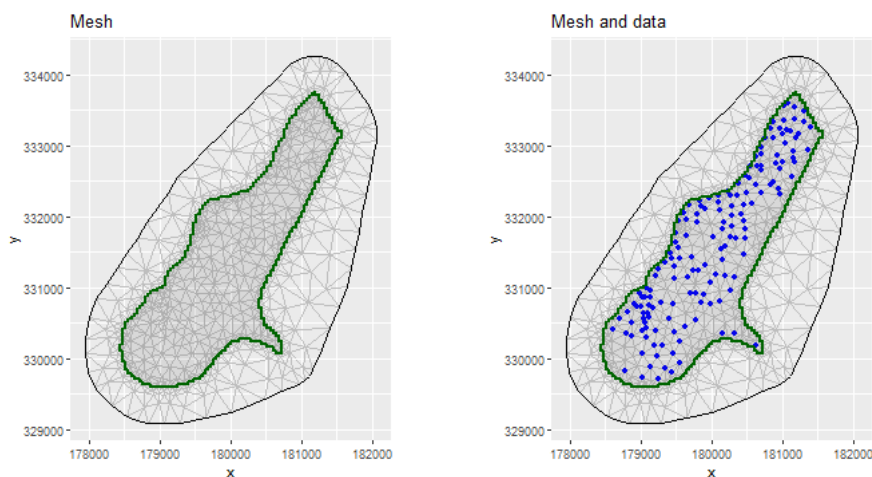


Figure 2: Mesh used in the estimation of the concentration of heavy metals around river Meuse, only with the boundary of the study region (left) and jointly with the survey locations (right).

In particular, the model will consider concentrations of lead and zinc. Concentrations have been measured at different locations which are displayed in the right plot in Figure 2. However, data on the concentrations of other metals is also available within the same dataset. Then, a model with all the metal concentrations could be handled following a similar structure. Values (in the original dataset) are considered in the log-scale. These log-transformed concentrations ($\log(y_l)$ and $\log(y_z)$, respectively) are assumed to be normally distributed. The mean of the log-concentration of lead is modeled using an intercept α_l plus a Gaussian process with a Matérn covariance, $u_{i,s}$, while the mean of the log-concentration of zinc is modeled using an intercept, α_z , the shared spatial effect, $u_{i,s}$, plus another spatial Gaussian process with Matérn covariance, $u_{i,z}$. This will allow us to assess differences in the spatial distribution of the concentration of both heavy metals. Furthermore, the Euclidian distance from the location of each measurement to the closest point of the river ($dist$) is included in both linear predictors as a covariate, each with a different slope parameter (β_l and β_z).

Hence, the model can be written as:

$$\begin{aligned}\log(y_l) &\sim N(\mu_{i,l}, \sigma_l) \\ \log(y_z) &\sim N(\mu_{i,z}, \sigma_z)\end{aligned}$$

where $\mu_{i,l}$ and $\mu_{i,z}$ represent the mean of the concentration of lead and zinc, respectively and which are modelled as follows:

$$\begin{aligned}\mu_{i,l} &= \alpha_l + \beta_l \text{dist}_i + u_{i,s}; & i &= 1, \dots, n \\ \mu_{i,z} &= \alpha_z + \beta_z \text{dist}_i + u_{i,s} + u_{i,z} & i &= 1, \dots, n\end{aligned}$$

In this case, the prior choices for the nominal range (r) and standard deviation (σ) for both the shared and zinc-specific spatial random effects are:

$$\begin{aligned}P(r < 2394.16) &= 0.95, \\ P(\sigma > 1000) &= 0.05.\end{aligned}$$

In plain words, this represents the belief that substantial correlations (> 0.1) between points more than half the maximum distance (2394.16 meters) apart are unlikely, and so are standard deviations greater than 1,000. These prior distributions are specified using the `inla.spde2.pcmatern()` function:

```
spde <- inla.spde2.pcmatern(mesh = mesh,
  prior.range = c(2394.16, 0.95), prior.sigma = c(1000, 0.05))
```

The SPDE approximation estimates the spatial random effects at the vertices of the mesh, so that estimates at any other point are based on the estimates at the vertices of the triangle that contains the point. The position of this point inside the triangle is identified using barycentric coordinates (see [Krainski et al., 2019](#), for details). The projector matrix contains all these coordinates for all the points in the dataset and it is obtained with function `inla.spde.make.A()`:

```
A.m <- inla.spde.make.A(mesh = mesh, loc = coordinates(meuse))
```

In order to fit models using the SPDE approach, data must be *stacked* using the appropriate format. Helper function `inla.stack()` can be used to build a stack object which contains the data (properly structured), the effects considered in the model, and the projector matrix. Specifically, a stack object will be built to model each variable. Then, all these objects will be put together into a single stack object so that a joint model can be fit.

In all the stacks the name of the response variable will be the same. The values will be given in a matrix with as many columns as the number of variables and as many rows as the number of observations. In the first stack, the values of the first response variable will be in the first column and the rest of columns will be filled with NA's. The other stacks and corresponding response variables will be defined analogously.

Regarding fixed and other latent effects included in the linear predictor, these can take the same name across stacks when the effect is shared for different response variables. When the effect of the same covariate or latent effect needs to be different across response variables, then different names must be used. See the R code below for more details.

The data are structured using the `inla.stack()` function specifying three elements: the data, the projector matrix, the different effects of the linear predictor and a tag to identify each part of the dataset in the final stack. This tag is used when retrieving particular results from the output.

```
# Create the stack object for lead
stk.lead <- inla.stack(
  data = list(log.y = cbind(log(meuse$lead), NA)),
  A = list(A.m, 1),
  effects = list(spatial.field.lead = 1:spde$n.spde,
    data.frame(Intercept.lead = 1, dist.lead = meuse$dist)),
  tag = "Lead")

# Create the stack object for zinc
stk.zinc <- inla.stack(
  data = list(log.y = cbind(NA, log(meuse$zinc))),
  A = list(A.m, A.m, 1),
  effects = list(
    spatial.field.zinc = 1:spde$n.spde, base.copy.zinc = 1:nv,
    data.frame(Intercept.zinc = 1, dist.zinc = meuse$dist)),
  tag = "Zinc")
```


A projector matrix and a stack for the prediction grid have also been created as follows:

```
# Create the projector matrix for the prediction
A.pr <- inla.spde.make.A(mesh = mesh, loc = coordinates(meuse.grid))

# Prepare the data for the prediction
y.pred <- matrix(NA, nrow = nrow(meuse.grid), ncol = 2)

# Build predicting stack for lead
stk.lead.pr <- inla.stack(
  data = list(log.y = y.pred),
  A = list(A.pr, 1),
  effects = list(spatial.field.lead = 1:spde$n.spde,
    data.frame(Intercept.lead = 1, dist.lead = meuse.grid$dist)),
  tag = "Lead.pred")

# Build predicting stack for zinc
stk.zinc.pr <- inla.stack(
  data = list(log.y = y.pred),
  A = list(A.pr, A.pr, 1),
  effects = list(
    spatial.field.zinc = 1:spde$n.spde, base.copy.zinc = 1:nv,
    data.frame(Intercept.zinc = 1, dist.zinc = meuse.grid$dist)),
  tag = "Zinc.pred")
```

Next, a stack object is built for the shared and zinc-specific effects in order to study the spatial patterns of these effects:

```
# Stack for the shared effect
stk.shared <- inla.stack(
  data = list(log.y = y.pred),
  A = list(A.pr),
  effects = list(spatial.field.lead = 1:spde$n.spde),
  tag = "Shared")

# Stack for the specific sp effect zinc
stk.zinc.spec <- inla.stack(
  data = list(log.y = y.pred),
  A = list(A.pr),
  effects = list(spatial.field.zinc = 1:spde$n.spde),
  tag = "Zinc.spec")
```

All the stack objects are put together in a single joint stack object using the `inla.stack()` function:

```
# Put all the stacks together
join.stack <- inla.stack(
  stk.lead, stk.zinc,
  stk.zinc.pr, stk.lead.pr,
  stk.shared, stk.zinc.spec)
```

The model formula is defined below. The latent effect of type copy is used to define the shared term of the model.

```
# Formulas for the model
form <- log.y ~ -1 + Intercept.lead + Intercept.zinc + dist.lead + dist.zinc +
  f(spatial.field.lead, model = spde) +
  f(spatial.field.zinc, model = spde) +
  f(base.copy.zinc, copy = "spatial.field.lead", fixed = TRUE)
```

Finally, the model is fit with **R-INLA**. Note how the family argument takes a vector of two "gaussian" elements (one for each heavy metal concentration). Furthermore, note that the data are obtained from the joint stack with the `inla.stack.data()` function and that the projector matrix `A` is also obtained using the `inla.stack.A()` function.

```
meuse.res <- inla(formula = form, verbose = FALSE,
  data = inla.stack.data(join.stack, spde = spde),
  family = rep("gaussian", 2),
  control.family = list(zero.prec, zero.prec),
  control.predictor = list(A = inla.stack.A(join.stack), compute = TRUE),
  control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE, mlik = TRUE, po = TRUE))
```

Given that the structure of the data used for model fitting is a stack now, particular parameter estimates can be accessed through function `inla.stack.index()`. This function can provide the indices to access the entries of specific data or latent effects in the summaries provided by **R-INLA**. For example, the following code can be used to obtain the posterior means of the fitted values obtained with the effects in the 'Lead.pred' stack:

```
idx.lead <- inla.stack.index(join.stack, 'Lead.pred')$data
meuse.grid$lead.pr <- meuse.res$summary.fitted.values[idx.lead, 'mean']
```

Figure 3 shows point estimates (posterior means) of the log-concentration of lead and zinc; the posterior mean of the shared and the zinc-specific effect are also shown. Note the similar spatial pattern across both metals.

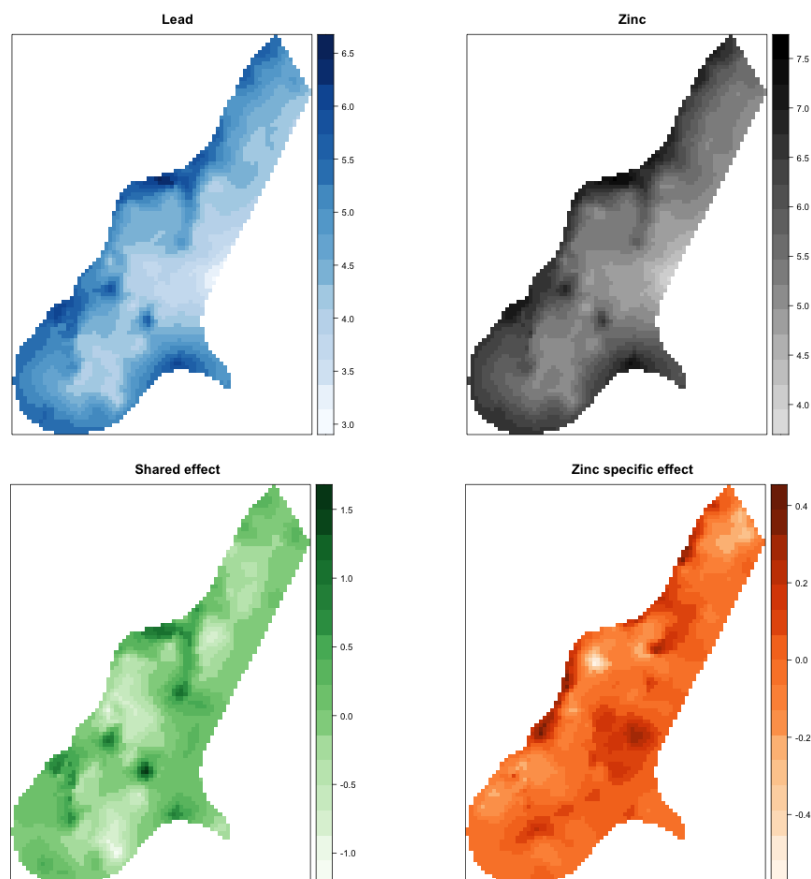


Figure 3: Estimates of the posterior means of the log-concentration of lead (top-left) and zinc (top-right). Estimates of the posterior mean of the shared spatial effect (bottom-left) and the zinc-specific spatial effect (bottom-right).

5 Multivariate point patterns

A point pattern is defined as a group of points (geographically located) which are a single realization of a stochastic process called point process. A multivariate point pattern can be defined as a group of several point patterns where each point pattern has a different origin, i.e., each point pattern is caused

by different processes. These are also referred to a specific case of *marked* point pattern (Baddeley et al., 2015) where each point pattern is labelled with a categorical mark.

In a completely random point process, points appear independently of each other and uniformly over the study region. This is also known as a homogeneous Poisson process with (constant) intensity λ , which measures the average number of points per unit area. It is also possible to consider a spatially varying intensity, $\lambda(s)$ (with s being a point of the study region), so that the process becomes an *inhomogeneous* Poisson process. The model can be extended to more complex point patterns (see, for example, Baddeley et al., 2015).

Several methods have been used to model the intensity function $\lambda(s)$. A complete spatial randomness scenario will not be considered here, that is, the intensity function of each point pattern will not be considered as constant over the study area. Specifically, intensities will be considered as continuous processes over the entire study region and they will be modeled using log-Gaussian Cox processes (Møller et al., 1998; Diggle et al., 2013). Log-Gaussian Cox processes can be fit by including spatial terms using the SPDE approach implemented in INLA (Simpson et al., 2016). The analysis of the intensity as a continuous function over the study region is similar to the case of multivariate geostatistics.

Given K point patterns in a region \mathcal{D} , an example of how to structure a multivariate point patterns model is:

$$\begin{aligned}\log(\lambda_0(s)) &= \alpha_0 + u_0(s); & s \in \mathcal{D} \\ \log(\lambda_j(s)) &= \alpha_j + u_0(s) + u_j(s) & s \in \mathcal{D}; j = 1, \dots, K - 1\end{aligned}$$

where $\lambda_0(s)$ is the intensity of the baseline point pattern and $\lambda_j(s)$ the intensity function of the j th point pattern. Moreover, α_0, α_j terms represent the intercepts and $u_0(s), u_j(s)$ are the spatial effects. Specifically, $u_0(s)$ is the shared (by the different point patterns) spatial term and $u_j(s)$ are the specific spatial terms which model the differences between each point pattern and the baseline.

In spatial epidemiology, the goal is often to discern whether a distribution of cases follows the spatial distribution of a set of controls, or whether it depends on exposure to pollution sources or other risk factors such as pollution (Palmí-Perales et al., 2021). This is an application of the model described here, in which the log-intensity of the controls can be modeled using a shared spatial term and the log-intensity of the cases can include this shared spatial term plus a disease-specific spatial term. Furthermore, the linear predictor can include other terms to account for risk factors.

Priors are set similarly as in the example for continuous spatial data as the spatial random effects $u_j(s)$, $j = 0, \dots, K - 1$ are actually modeled as SPDE latent effects. Other parameters of effects included in the model (e.g., fixed effects) will be assigned priors accordingly.

5.1 Example: forest fires in Castilla-La Mancha (Spain)

The `spatstat` package contains the `clmfires` dataset. This dataset records the occurrence of forest fires in the region of Castilla-La Mancha (Spain) from 1998 to 2007. These forest fires are classified by four different causes: lightning, accidental, intentional, and other fires (Figure 4). After loading the packages, the data are accessed using the `data()` function:

```
#Load and display the data
data("clmfires")
```

Specifically, in this example, the intensities of the different types of forest fires are estimated by considering lightning fires as the baseline pattern. Furthermore, INLA is used to assess the similarities or differences among their spatial patterns. First of all, the mesh is built using the coordinates of the boundary of the dataset (`bdy.SP`) using the code below.

```
mesh <- inla.mesh.2d(
  boundary = list(bdy.SP, NULL), cutoff = 2, max.edge = c(20, 50),
  min.angle = 27, offset = c(1, 50), n=c(16,16))
```

Following the model structure for multivariate point patterns detailed above, the log-intensity of the lightning fires $\lambda_l(s)$ will be modeled using an intercept, α_l , and a spatial Gaussian effect with Matérn covariance, $u_l(s)$ as follows

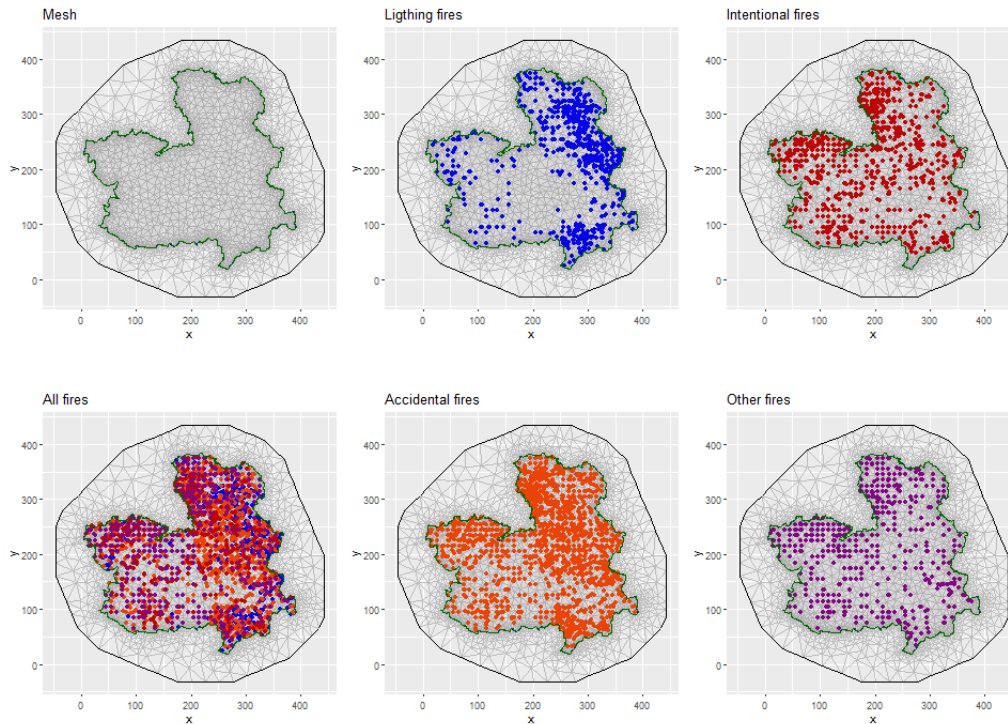


Figure 4: The mesh used is shown alone (top-left) and with all the fire types (bottom-left). Furthermore, the mesh is also displayed for each fire type separately: lightning fires (top-middle), accidental fires (bottom-middle), intentional fires (top-right), and other fires (bottom-right).

$$\log \lambda_l(s) = \alpha_l + u_l(s); s \in D$$

Spatial effect $u_l(s)$ will also be shared in the linear predictor of the other types of fire. Similarly, the log-intensity of the accidental, intentional, and other fires (which includes unknown cause) will be modeled using specific intercepts, plus the shared spatial effect, plus a specific spatial effect as follows:

$$\log \lambda_a(s) = \alpha_a + u_l(s) + u_a(s); s \in D$$

$$\log \lambda_i(s) = \alpha_i + u_l(s) + u_i(s); s \in D$$

$$\log \lambda_o(s) = \alpha_o + u_l(s) + u_o(s) s \in D$$

Following the prior specification section, the chosen PC-priors in this example are:

$$P(r < 200) = 0.95$$

$$P(\sigma > 100) = 0.05$$

where a nominal range higher than half the maximum distance (e.g., 200 kilometres) of the domain is unlikely. Similarly for the nominal standard deviation, its prior assumes that it is really unlikely that it is higher than 100 in this context. These prior distributions have been specified using function `inla.spde2.pcmatern()`:

```
spde <- inla.spde2.pcmatern(mesh = mesh,
  prior.range = c(200, 0.95), prior.sigma = c(10, 0.05))
```

Here, argument `prior.range` sets the prior for the range and the argument `prior.sigma` sets the prior for the standard deviation of the spatial effect.

As this model includes SPDE latent effects, data must be put together using stack objects. In the case of point patterns, the data included in the stack function is different from the geostatistical example as point pattern data has to be detailed following a specific structure (see below). Model fitting now relies on the methods described in [Simpson et al. \(2016\)](#). In this case, two elements will be

included in a list for storing the data of each point pattern.

The first element of the list will be a matrix with $N_v + N_i$ rows and K columns where N_v is the number of vertices of the SPDE mesh, N_i is the total number of points of the i -th point pattern and K is the number of different point patterns. The matrix of the stack of the i -th point pattern will be filled with NA's except for the i -th column. This i -th column will contain firstly N_v zeros corresponding to the points of the mesh. After these zeros, there will be N_i ones corresponding to the N_i points of the i -th point pattern.

The second element of the list will be a vector containing an offset. Specifically, the length of this vector is also $N_v + N_i$ where the first N_v elements will contain the "weights" of the mesh points. The reader is referred to [Simpson et al. \(2016\)](#) for more details about this approximation. The rest of the values will be zeros.

These two elements have to be created for each stack of each point pattern. Then, as before, the stacks are combined in a single stack object. As a toy example, consider a mesh with $N_v = 3$, two point patterns ($K = 2$) with three and four points, respectively ($N_1 = 3$ and $N_2 = 4$). The weights associated to the three mesh points will be 2.3, 4.3 and 6.2. Then, the data passed to the first and second stack are:

$$\begin{bmatrix} 0 & \text{NA} \\ 0 & \text{NA} \\ 0 & \text{NA} \\ 1 & \text{NA} \\ 1 & \text{NA} \\ 1 & \text{NA} \end{bmatrix}, \begin{bmatrix} 2.3 \\ 4.3 \\ 6.2 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \begin{bmatrix} \text{NA} & 0 \\ \text{NA} & 0 \\ \text{NA} & 0 \\ \text{NA} & 1 \\ \text{NA} & 1 \\ \text{NA} & 1 \\ \text{NA} & 1 \end{bmatrix}, \begin{bmatrix} 2.3 \\ 4.3 \\ 6.2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

As stated above, when analysing point patterns it is necessary to assign some weights to the points of the mesh. This is done by creating a Voronoi tessellation using these points, so that the area of the associated polygon becomes the associated weight. The following code illustrates how to obtain the Voronoi tessellation and the associated weights:

```
library(deldir)
dd <- deldir(mesh$loc[, 1], mesh$loc[, 2])
# Create a list of tiles in a tessellation
mytiles <- tile.list(dd)

# Boundary as a polygon
pl.study <- as(bdy, "gpc.poly")
# Area of the study area
area.poly(pl.study)

# Compute weights as the area of the polygon given as an
# intersection between Voronoi tiles and domain polygon
w <- unlist(lapply(mytiles,
  function(p) area.poly(
    intersect(as(cbind(p$x, p$y), "gpc.poly"), pl.study)
  )
))
)
```

These computed weights are introduced as the expected weights on the mesh points. Furthermore, the data should be structured as follows:

```
# Data for the stack function: lightning fires
e.lig <- c(w, rep(0, n.lig))
y.lig <- matrix(NA, nrow = nv + n.lig, ncol = n.pp)
y.lig[, 1] <- rep(0:1, c(nv, n.lig))

# Data for the stack function: accidental fires
e.acc <- c(w, rep(0, n.acc))
y.acc <- matrix(NA, nrow = nv + n.acc, ncol = n.pp)
y.acc[, 2] <- rep(0:1, c(nv, n.acc))

# Data for the stack function: intentional fires
```

```
e.int <- c(w, rep(0, n.int))
y.int <- matrix(NA, nrow = nv + n.int, ncol = n.pp)
y.int[, 3] <- rep(0:1, c(nv, n.int))

# Data for the stack function: other fires
e.oth <- c(w, rep(0, n.oth))
y.oth <- matrix(NA, nrow = nv + n.oth, ncol = n.pp)
y.oth[, 4] <- rep(0:1, c(nv, n.oth))
```

Another element of the SPDE approach is the projector matrix that represents each point using barycentric coordinates from the mesh points (as explained above). When analysing multivariate point patterns, this matrix has two parts: one for the mesh points (named `imat`), which is a diagonal matrix as every mesh point matches itself when using barycentric coordinates, and the other for the points of the point pattern (named `lmat`). Then, the projector matrix is the combination of these two matrices:

```
#imat: define imat
imat <- Diagonal(nv, rep(1, nv))

#lmat: define lmat
lmat.lig <- inla.spde.make.A(mesh, pts.lig)
lmat.acc <- inla.spde.make.A(mesh, pts.acc)
lmat.int <- inla.spde.make.A(mesh, pts.int)
lmat.oth <- inla.spde.make.A(mesh, pts.oth)

#Projector matrix: Put together imat and lmat
A.lig <- rbind(imat, lmat.lig)
A.acc <- rbind(imat, lmat.acc)
A.int <- rbind(imat, lmat.int)
A.oth <- rbind(imat, lmat.oth)
```

Once all the elements of the stack object are set, the `inla.stack()` function is used to build the different stack objects. For instance, the stack objects of two (out of four) types of forest fires are shown here below:

```
# Create the stack for the lightning fires
stk.lig <- inla.stack(
  data = list(y = y.lig, e = e.lig),
  A = list(A.lig, 1),
  effects = list(spatial.field.lig = s.index.lig,
    data.frame(Intercept.lig = rep(1, n.lig))
  ),
  tag = "Lighting")

# Create the stack for the accidental fires
stk.acc <- inla.stack(
  data = list(y = y.acc, e = e.acc),
  A = list(A.acc, A.acc, 1),
  effects = list(
    base.copy.acc = 1:nv,
    spatial.field.acc = s.index.acc,
    data.frame(Intercept.acc = rep(1, n.acc))
  ),
  tag = "Accidental")
```

As in the example for continuous spatial data, the stack objects for the predictions have to be built following the same structure (their name will end with ".pr" in the R code). Finally, all the stack objects are joined into a single joint stack object as follows:

```
# All stacks together
join.stack <- inla.stack(
  stk.lig, stk.acc, stk.int, stk.oth,
  stk.lig.pr, stk.acc.pr, stk.int.pr, stk.oth.pr,
  stk.shared, stk.acc.spec, stk.int.spec, stk.oth.spec)
```

Note that in this example separate stack objects have been created for estimating the intensity of each of the different types of forest types, the shared spatial pattern (`stack.stk.shared`) and the three different specific spatial effects (`stacks.stk.acc.spec`, `stk.int.spec` and `stk.oth.spec`).

The model formula is defined here below, where the latent effects of type copy are used to define the shared terms of the model:

```
form <- y ~ -1 + Intercept.lig + Intercept.acc + Intercept.int + Intercept.oth +
  f(spatial.field.lig, model = spde) +
  f(spatial.field.acc, model = spde) +
  f(base.copy.acc, copy = "spatial.field.lig", fixed = TRUE) +
  f(spatial.field.int, model = spde) +
  f(base.copy.int, copy = "spatial.field.lig", fixed = TRUE) +
  f(spatial.field.oth, model = spde) +
  f(base.copy.oth, copy = "spatial.field.lig", fixed = TRUE)
```

The model is fit with **R-INLA** using the code below. Note how the family argument takes a vector of four "poisson" elements (one for each fire type).

```
pp.res <- inla(formula = form, verbose = FALSE,
  data = inla.stack.data(join.stack, spde = spde),
  family = rep("poisson", 4),
  control.predictor = list(A = inla.stack.A(join.stack), compute = TRUE, link = 1),
  control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE, mlik = TRUE, po = TRUE)
)
```

Figure 5 shows the posterior mean of the intensity of each type of the forest fires. A different spatial pattern can be seen for each fire type. Lightning fires mostly appear in the east part of the region, while the other three types of forest fires are more likely to appear in the west and the central parts of the region. Additionally, the posterior means of the shared and specific spatial effects can also be seen in Figure 6.

6 Discussion

In this paper, we have shown how to fit multivariate spatial models using the **R-INLA** package. In particular, the details of how to analyse each spatial data type (lattice, continuous, and point patterns) have been given. Furthermore, we have illustrated the application of these models using three datasets: simulated data on mortality by three types of cancer (available from https://github.com/becarioprecario/INLAMCMC_spatial_examples), the `clmfires` dataset in the **spatstat** package and the `meuse` dataset in the **gstat** package. Furthermore, more complex spatial and spatio-temporal models to multivariate data can be fitted with **R-INLA** (see, for example, [Palmí-Perales et al., 2021](#)).

The main goal of this work has been to illustrate how to perform multivariate spatial Bayesian inference using **R-INLA**. In particular, we have paid attention to the different steps required to create the necessary data structures for model fitting. The advantage of **R-INLA** compared with alternatives is its computational efficiency; particularly, MCMC-based methods can struggle in the computationally demanding high-dimensional setting of spatial data sets. Hence, it has been shown that **R-INLA** is a useful and a worthwhile toolbox for fitting multivariate spatial models. Additionally, the necessary R scripts to reproduce the examples are available at <https://github.com/FranciscoPalmiPerales/Mult-Sp-INLA>.

7 Acknowledgements

This work has been supported by grants SBPLY/17/180501/000491 and SBPLY/21/180501/000241, funded by Consejería de Educación, Cultura y Deportes (JCCM, Spain) and Fondo Europeo de Desarrollo Regional, grant MTM2016-77501-P and PID2019-106341GB-I00, funded by Ministerio de Economía y Competitividad (Spain), grants PID2019-106341GB-I00 and PID2022-136455NB-I00, funded by Ministerio de Ciencia e Innovación (Spain), and grant CIAICO/2022/165, funded by Dirección General de Ciencia e Investigación (Generalitat Valenciana). F. Palmí-Perales was supported by a doctoral scholarship awarded by the University of Castilla-La Mancha (Spain) and by grant PID2021-128228NB-I00 funded by Ministerio de Ciencia e innovación.

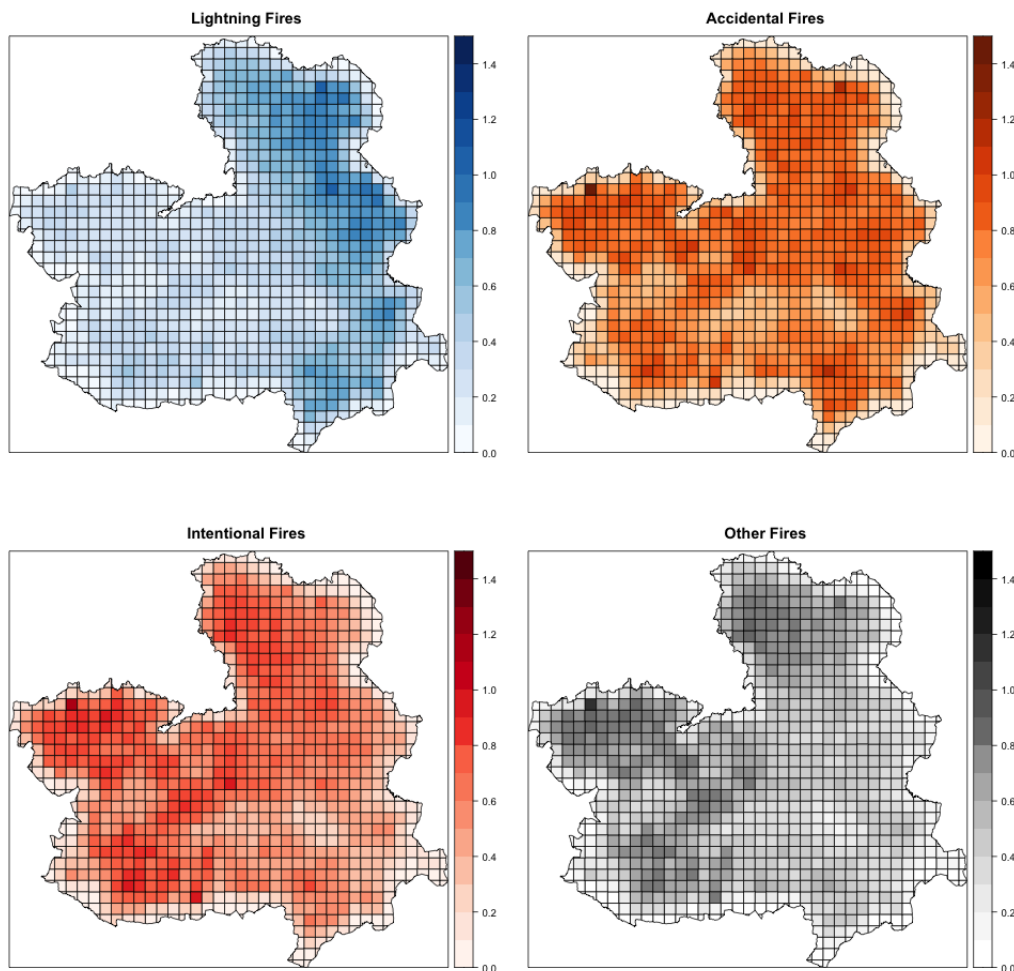


Figure 5: Posterior mean of the intensity of the lightning (top-left), accidental (top-right), intentional (bottom-left) and other (bottom-right) fires.

References

- A. Baddeley, E. Rubak, and R. Turner. *Spatial Point Patterns: Methodology and Applications with R*. Chapman & Hall/CRC Press, London, 2015. URL <http://www.crcpress.com/Spatial-Point-Patterns-Methodology-and-Applications-with-R/Baddeley-Rubak-Turner/9781482210200/>. [p173, 182]
- H. Bakka, H. Rue, G.-A. Fuglstad, A. Riebler, D. Bolin, E. Krainski, D. Simpson, and F. Lindgren. Spatial modelling with R-INLA: A review. *WIREs Comput Stat*, 10(6):1–24, 2018. URL <https://doi.org/10.1002/wics.1443>. [p172]
- S. Banerjee, B. P. Carlin, and A. E. Gelfand. *Hierarchical modeling and analysis for spatial data*. CRC press, 2014. [p172, 174, 175]
- M. Blangiardo and M. Cameletti. *Spatial and spatio-temporal Bayesian models with R-INLA*. John Wiley & Sons, 2015. [p172]
- P. J. Diggle, P. Moraga, B. Rowlingson, and B. M. Taylor. Spatial and spatio-temporal log-Gaussian Cox processes: Extending the geostatistical Paradigm. *Statistical Science*, 28(4):542 – 563, 2013. URL <https://doi.org/10.1214/13-STS441>. [p172, 182]
- A. O. Finley, S. Banerjee, and B. P. Carlin. spBayes: An R package for univariate and multivariate hierarchical point-referenced spatial models. *Journal of Statistical Software*, 19(4):1, 2007. URL <https://www.jstatsoft.org/index.php/jss/article/view/v019i04>. [p173]

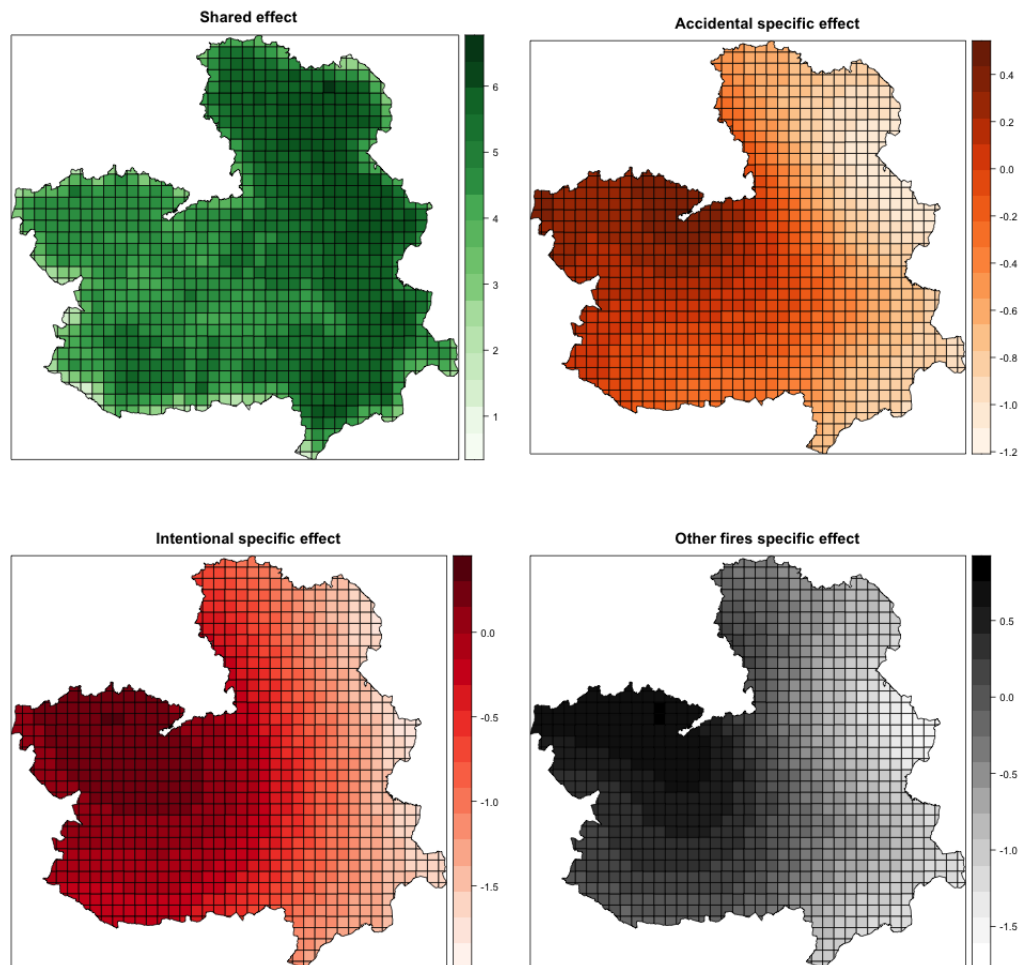


Figure 6: Posterior mean of the shared spatial effect (top-left), the accidental-specific (top-right), the intentional-specific (bottom-left) and the other-specific (bottom-right) spatial effects.

- A. O. Finley, S. Banerjee, and A. E. Gelfand. *spBayes* for large univariate and multivariate point-referenced spatio-temporal data models. *Journal of Statistical Software*, 63(13):1–28, 2015. URL <http://www.jstatsoft.org/v63/i13/>. [p173]
- A. Gelman. Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). *Bayesian Analysis*, 1(3):515 – 534, 2006. URL <https://doi.org/10.1214/06-BA117A>. [p175]
- W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in practice*. Chapman & Hall, 1995. [p172]
- V. Gómez-Rubio. *Bayesian inference with INLA*. CRC Press, 2020. [p172, 173, 175, 176]
- V. Gómez-Rubio and F. Palmí-Perales. Multivariate posterior inference for spatial models with the integrated nested Laplace approximation. *Journal of the Royal Statistical Society, Series C*, 68(1): 199–215, 2019. URL <https://doi.org/10.1111/rssc.12292>. [p172, 173, 175]
- V. Gómez-Rubio, M. Cameletti, and F. Finazzi. Analysis of massive marked point patterns with stochastic partial differential equations. *Spatial Statistics*, 14:179–196, 2015. URL <https://doi.org/10.1016/j.spasta.2015.06.003>. [p172]
- V. Gómez-Rubio, P. Zheng, P. Diggle, D. C. Sterratt, R. D. Peng, D. Murdoch, and B. Rowlingson. *spatialkernel: Non-Parametric Estimation of Spatial Segregation in a Multivariate Point Process*, 2017. URL <https://CRAN.R-project.org/package=spatialkernel>. R package version 0.4-23. [p173]

- V. Gómez-Rubio, F. Palmí-Perales, G. López-Abente, R. Ramis-Prieto, and P. Fernández-Navarro. Bayesian joint spatio-temporal analysis of multiple diseases. *SORT*, 1:51–74, 2019. URL <https://doi.org/10.2436/20.8080.02.79>. [p175]
- E. T. Krainski, V. Gómez-Rubio, H. Bakka, A. Lenzi, D. Castro-Camilo, D. Simpson, F. Lindgren, and H. Rue. *Advanced Spatial Modeling with Stochastic Partial Differential Equations Using R and INLA*. Chapman & Hall/CRC, Boca Raton, FL, 2019. [p172, 179]
- D. Lee. CARBayes: An R package for Bayesian spatial modeling with conditional autoregressive priors. *Journal of Statistical Software*, 55(13):1–24, 2013. URL <https://www.jstatsoft.org/v55/i13/>. [p173]
- F. Lindgren, H. Rue, and J. Lindström. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011. URL <https://doi.org/10.1111/j.1467-9868.2011.00777.x>. [p172, 177]
- F. Lindgren, H. Rue, et al. Bayesian spatial modelling with R-INLA. *Journal of Statistical Software*, 63(19):1–25, 2015. URL <https://doi.org/10.18637/jss.v063.i19>. [p172]
- Y. C. MacNab. Some recent work on multivariate Gaussian Markov random fields. *Test*, 27(3):497–541, 2018. URL <https://doi.org/10.1007/s11749-018-0605-3>. [p172]
- M. Martínez-Beneito. A general modelling framework for multivariate disease mapping. *Biometrika*, 100(3):539–553, 2013. URL <https://doi.org/10.1093/biomet/ast023>. [p175]
- M. A. Martínez-Beneito and P. Botella-Rocamora. *Disease Mapping: From Foundations to Multidimensional Modeling*. CRC Press, 2019. [p172]
- M. A. Martínez-Beneito, P. Botella-Rocamora, and S. Banerjee. Towards a multidimensional approach to Bayesian disease mapping. *Bayesian analysis*, 12(1):239, 2017. URL <https://doi.org/10.1214/16-BA995>. [p172]
- J. Møller, A. R. Syversveen, and R. P. Waagepetersen. Log Gaussian Cox Processes. *Scandinavian Journal of Statistics*, 25(3):451–482, 1998. URL <https://doi.org/10.1111/1467-9469.00115>. [p182]
- F. Palmí-Perales, V. Gómez-Rubio, and M. A. Martínez-Beneito. Bayesian Multivariate Spatial Models for Lattice Data with INLA. *Journal of Statistical Software*, 98(2):1–29, 2021. URL <https://www.jstatsoft.org/index.php/jss/article/view/v098i02>. [p175, 186]
- F. Palmí-Perales, V. Gómez-Rubio, G. López-Abente, R. Ramis, J. M. Sanz-Anquela, and P. Fernández-Navarro. Approximate Bayesian inference for multivariate point pattern analysis in disease mapping. *Biometrical Journal*, 63(3):632–649, 2021. URL <https://doi.org/10.1002/bimj.201900396>. [p182]
- E. J. Pebesma. Multivariable geostatistics in S: the gstat package. *Computers & Geosciences*, 30(7):683–691, 2004. [p173]
- E. J. Pebesma and C. G. Wesseling. Gstat: A Program for Geostatistical Modelling, Prediction and Simulation. *Computers & Geosciences*, 24(1):17–31, 1998. [p173]
- R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2022. URL <https://www.R-project.org/>. [p173]
- H. Rue and L. Held. *Gaussian Markov Random Fields. Theory and Applications*. Chapman & Hall, New York, 2005. [p172, 173, 174]
- H. Rue, S. Martino, and N. Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the royal statistical society: Series b (statistical methodology)*, 71(2):319–392, 2009. URL <https://doi.org/10.1111/j.1467-9868.2008.00700.x>. [p172, 173]
- H. Rue, F. Lindgren, D. Simpson, S. Martino, E. Teixeira Krainski, H. Bakka, A. Riebler, and G.-A. Fuglstad. *INLA: Full Bayesian Analysis of Latent Gaussian Models using Integrated Nested Laplace Approximations*, 2020. R package version 20.03.17. [p172]
- D. Simpson, J. Illian, F. Lindgren, S. H. Sørbye, and H. Rue. Going off grid: Computationally efficient inference for log-Gaussian Cox processes. *Biometrika*, 103(1):49–70, 2016. URL <https://doi.org/10.1093/biomet/asv064>. [p182, 183, 184]

- D. Simpson, H. Rue, A. Riebler, T. G. Martins, and S. H. Sørbye. Penalising model component complexity: A principled, practical approach to constructing priors. *Statistical Science*, 32(1):1–28, 2017. ISSN 08834237, 21688745. URL <http://www.jstor.org/stable/26408114>. [p177]
- S. H. Sørbye, J. B. Illian, D. P. Simpson, D. Burslem, and H. Rue. Careful prior specification avoids incautious inference for log-Gaussian Cox point processes. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 68(3):543–564, 2019. URL <https://doi.org/10.1111/rssc.12321>. [p177]
- M. N. M. Van Lieshout and A. J. Baddeley. Indices of dependence between types in multivariate point patterns. *Scandinavian Journal of Statistics*, 26(4):511–532, 1999. URL <https://doi.org/10.1111/1467-9469.00165>. [p172]
- R. Waagepetersen, Y. Guan, A. Jalilian, and J. Mateu. Analysis of multispecies point patterns by using multivariate log-Gaussian Cox processes. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 65(1):77–96, 2016. ISSN 00359254, 14679876. URL <http://www.jstor.org/stable/24773017>. [p172]

Francisco Palmí-Perales

Department of Statistics and Operational Research, Faculty of Mathematics, Universitat de València

C/Dr. Moliner, 50, 46100

Burjassot, Spain

0000-0002-0751-7315

Francisco.Palmi@uv.es

Virgilio Gómez-Rubio

Department of Mathematics, E.T.S. Ingeniería Industrial-Albacete, Universidad de Castilla-La Mancha

Av. de España, s/n, 02071

Albacete, Spain

0000-0002-4791-3072

Virgilio.Gomez@uclm.es

Roger S. Bivand

Department of Economics, Norwegian School of Economics

Helleveien 30, N-5045

Bergen, Norway

0000-0003-2392-6140

Roger.Bivand@nhh.no

Michela Cameletti

Department of Economics, Università degli studi di Bergamo

Via dei Caniana 2. IT-24127

Bergamo, Italy

0000-0002-6502-7779

michela.cameletti@unibg.it

Håvard Rue

King Abdullah University of Science and Technology

Thuwal, Saudi Arabia

0000-0002-0222-1881

haavard.rue@kaust.edu.sa